EPTCS 380

Proceedings of the

Fifth International Conference on Applied Category Theory

Glasgow, United Kingdom, 18-22 July 2022

Edited by: Jade Master and Martha Lewis

7th August 2023
ISSN: 2075-2180
Open Publishing Association

Table of Contents

Table of Contents i

Preface iii

Canonical Gradings of Monads 1
Flavien Breuvart, Dylan McDermott and Tarmo Uustalu

Categorification of Negative Information using Enrichment 22
Andrea Censi, Emilio Frazzoli, Jonathan Lorand and Gioele Zardini

A Category of Surface-Embedded Graphs 41
Malin Altenmüller and Ross Duncan

Coend Optics for Quantum Combs 63
James Hefford and Cole Comfort

Compositional Modeling with Stock and Flow Diagrams 77
John Baez, Xiaoyan Li, Sophie Libkind, Nathaniel D. Osgood and Evan Patterson

Cornering Optics 97
Guillaume Boisseau, Chad Nester and Mario Román

Data Structures for Topologically Sound Higher-Dimensional Diagram Rewriting 111
Amar Hadzihasanovic and Diana Kessler

Dependent Optics 128
Pietro Vertechi

Diegetic Representation of Feedback in Open Games 145
Matteo Capucci

Differential 2-rigs 159
Fosco Loregian and Todd Trimble

Dynamic Operads, Dynamic Categories: From Deep Learning to Prediction Markets 183
Brandon T. Shapiro and David I. Spivak

Extending Resource Monotones using Kan Extensions 203
Robin Cockett, Isabelle Jianing Geng, Carlo Maria Scandolo and Priyaa Varshinee Srinivasan

Fibrational Linguistics (FibLang): Language Acquisition 224
Fabrizio Genovese, Fosco Loregian and Caterina Puca

Lax Liftings and Lax Distributive Laws 237
Ezra Schoen

Magnitude and Topological Entropy of Digraphs 254
Steve Huntsman

Monoidal Width: Capturing Rank Width 268
Elena Di Lavore and Paweł Sobociński

On the Pre- and Promonoidal Structure of Spacetime 284
James Hefford and Aleks Kissinger

Open Dynamical Systems as Coalgebras for Polynomial Functors, with Application to Predictive Processing 307
Toby St. Clere Smithe

Polynomial Functors and Shannon Entropy 331
David I. Spivak

Promonads and String Diagrams for Effectful Categories 344
Mario Román

String Diagrams for Layered Explanations 362
Leo Lobski and Fabio Zanasi

Unification of Modal Logic via Topological Categories 383
Lingyuan Ye

Universal Properties of Lens Proxy Pullbacks 400
Matthew Di Meglio

Value Iteration is Optic Composition 417
Jules Hedges and Riu Rodríguez Sakamoto

Preface

The Fifth International Conference on Applied Category Theory took place at the University of Strathclyde on 18–22 July 2022, following the previous meetings at Leiden (2018), Oxford (2019), MIT (2020, fully online), and Cambridge (2021). It was preceded by the Adjoint School 2022 (11–15 July), a collaborative research event in which junior researchers worked on cutting-edge topics under the mentorship of experts. The conference comprised 59 contributed talks, a poster session, an industry showcase session, and a session where junior researchers who had attended the Adjoint School presented the results of their research at the school. Information regarding the conference may be found at https://msp.cis.strath.ac.uk/act2022.

ACT 2022 was a hybrid event, with physical attendees present in Glasgow and other participants taking part online. All talks were streamed to YouTube and with synchronous discussion on Zulip. Links to recordings of the talks as well as notes taken during the conference may be found at https://statebox.tv/act2022.

Submission to ACT2022 had three tracks: extended abstracts, software demonstrations, and proceedings. The extended abstract and software demonstration submissions had a page limit of 2 pages, and the proceedings track had a page limit of 14 pages. Only papers submitted to the proceedings track were considered for publication in this volume. ACT2022 was the first year with a software demonstration track. In total, there were 97 submissions, of which 59 were accepted for presentation and 24 for publication in this volume. Publication of accepted submissions in the proceedings was determined by personal choice of the authors and not based on quality. Each submission received a review from three different members of the programming committee, and papers were selected based on discussion and consensus by these reviewers.

The contributions to ACT2022 ranged from pure to applied and included contributions in a wide range of disciplines in science and engineering. ACT2022 included talks in linguistics, functional programming, classical mechanics, quantum physics, probability theory, electrical engineering, epidemiology, thermodynamics, engineering, and logic. The quality of submissions to ACT2022 was very high, containing both cutting-edge category theory and a high degree of relevance to the chosen application. Many of the submissions had software demonstrating their work or represented work done in collaboration with industry or a scientific organization. The industry session included 10 invited talks by practitioners using category theory in private enterprise. ACT2022 was sponsored by Huawei, Protocol Labs, Cambridge Quantum, Conexus, Topos, and SICSA (Scottish Informatics and Computer Science Alliance).

ACT2022 was the first meeting since the inaugural which was not severely affected by COVID-19. The conference saw the community return in full force with more attendees, submissions, and general initiatives than previous editions of the conference. In particular, we were inspired by how many new people from a very wide range of backgrounds have joined in research with the applied category theory community. We hope that ACT, the conference and the community, continue to grow in future years. We are excited to see the new developments at ACT2023.

Jade Master and Martha Lewis
Chairs of the ACT 2022 Programme Committee

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. iii–iii, doi:10.4204/EPTCS.380.0

Canonical Gradings of Monads

Flavien Breuvart
LIPN, Université Sorbonne Paris Nord, France
flavien.breuvart@lipn.univ-paris13.fr

Dylan McDermott
Reykjavik University, Iceland
dylanm@ru.is

Tarmo Uustalu
Reykjavik University, Iceland
Tallinn University of Technology, Estonia
tarmo@ru.is

We define a notion of grading of a monoid T in a monoidal category $\mathscr{C}$, relative to a class of morphisms $\mathscr{M}$ (which provide a notion of $\mathscr{M}$-subobject). We show that, under reasonable conditions (including that $\mathscr{M}$ forms a factorization system), there is a canonical grading of T. Our application is to graded monads and models of computational effects. We demonstrate our results by characterizing the canonical gradings of a number of monads, for which $\mathscr{C}$ is endofunctors with composition. We also show that we can obtain canonical grades for algebraic operations.

1 Introduction

This paper is motivated by quantitative modelling of computational effects from mathematical programming semantics. It is standard in this domain to model notions of computational effect, such as nondeterminism or manipulation of external state, by (strong) monads [11]. In many applications, however, it is useful to be able to work with quantified effects, e.g., how many outcomes a computation may have, or to what degree it may read or overwrite the state. This is relevant, for example, for program optimizations or analyses to assure that a program can run within allocated resources. Quantification of effectfulness is an old idea and goes back to type-and-effect systems [8]. Mathematically, notions of quantified effect can be modelled by graded (strong) monads [13, 10, 4].

It is natural to ask if there are systematic ways for refining a non-quantitative model of some effect into a quantitative version, i.e., for producing a graded monad from a monad. In this paper, we answer this question in the affirmative. We show how a monad on a category can be graded with any class of subfunctors (intuitively, predicates on computations) satisfying reasonable conditions, including that it forms a factorization system on some monoidal subcategory of the endofunctor category. Moreover, this grading is canonical, namely universal in a certain 2-categorical sense. We also show that algebraic operations of the given monad give rise to *flexibly graded* algebraic operations [5] of the canonically graded monad. Instead of working concretely with monads on a category, we work abstractly with monoids in a (skew) monoidal category equipped with a factorization system.

The structure of the paper is this. In Section 2, we introduce the idea of grading by subobjects for general objects and instantiate this for grading of functors. We then proceed to gradings of monoids and monads in Section 3. In Section 4, we explore the specific interesting case of grading monads canonically by subsets of their sets of shapes. In Section 5, we explain the emergence of canonical flexibly graded algebraic operations for canonical gradings of monads. One longer proof is in Appendix A.

We introduce the necessary concepts regarding the classical topics of monads, monoidal categories and factorization systems. For additional background on the more specific concepts of graded monad and skew monoidal category, which we also introduce, we refer to [4, 2] and [14, 7] as entry points.

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 1–21, doi:10.4204/EPTCS.380.1

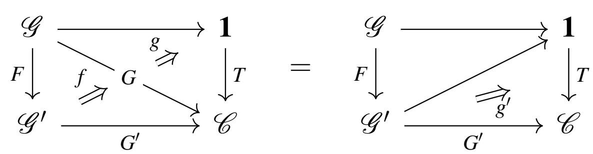
2 Grading objects and functors

As a first step towards gradings of monoids, we introduce the notion of a grading of an object of a category $\mathscr{C}$ with respect to a class of morphisms $\mathscr{M}$ in $\mathscr{C}$. We show that every object T has a canonical such grading. The case we care most about is when $\mathscr{C}$ is a category of endofunctors, so that, in the next section, where we extend these results to gradings of monoids, the monoids are exactly monads.

Definition 2.1. Let $\mathscr{G}$ be a category, whose objects e we call *grades*. A *$\mathscr{G}$-graded object* of a category $\mathscr{C}$ is a functor $G : \mathscr{G} \to \mathscr{C}$.

Let $\mathscr{M}$ be a class of morphisms of a category $\mathscr{C}$, and T be an object of $\mathscr{C}$. There is a category $\mathscr{M}/T$, which has as objects *$\mathscr{M}$-subobjects* of T, i.e., pairs (S,s) of an object S and an $\mathscr{M}$-morphism $s : S \rightarrowtail T$. Morphisms $f : (S,s) \to (S',s')$ are $\mathscr{C}$-morphisms $f : S \to S'$ such that $s = s' \circ f$. We then have a $\mathscr{M}/T$-graded object $T_{\mathscr{M}}$ of $\mathscr{C}$, defined by $T_{\mathscr{M}}(S,s) = S$. This graded object forms an $\mathscr{M}$-grading in the sense of the following definition, and is in fact the canonical $\mathscr{M}$-grading (see Theorem 2.3 below).

Definition 2.2. Let $\mathscr{M}$ be a class of morphisms of a category $\mathscr{C}$. An *$\mathscr{M}$-grading* $(\mathscr{G},G,g)$ of an object T of $\mathscr{C}$ consists of a category $\mathscr{G}$, a functor $G : \mathscr{G} \to \mathscr{C}$ (= a $\mathscr{G}$-graded object of $\mathscr{C}$), and a natural transformation typed $g_d : Gd \rightarrowtail T$ whose components are all in $\mathscr{M}$. A *morphism* $(F,f) : (\mathscr{G},G,g) \to (\mathscr{G}',G',g')$ between such gradings is a functor $F : \mathscr{G} \to \mathscr{G}'$ equipped with a natural isomorphism $f : G' \cdot F \cong G$, such that $g_d \circ f_d = g'_{Fd}$.



A *2-cell* $\beta : (F,f) \Rightarrow (F',f')$ between such morphisms is a natural transformation $\beta : F \Rightarrow F'$ such that $f' \circ (G' \cdot \beta) = f$. These form a 2-category $\mathbf{Grade}_{\mathscr{M}}T$.

We organize gradings into a 2-category so that we can prove a universal property (the following theorem) that characterizes the canonical grading. The characterization is up to equivalence; there is no reason to distinguish between isomorphic grades, therefore we can work with gradings that are equivalent to the canonical one. Working up to equivalence has the added benefit that, since $\mathscr{M}/T$ is often equivalent to a small category, we often have a canonical grading with a small set of grades.

Theorem 2.3. *Let $\mathscr{M}$ be a class of a morphisms of a category $\mathscr{C}$, and let T be an object of $\mathscr{C}$. The data $(\mathscr{M}/T, T_{\mathscr{M}}, \mathrm{snd})$, where $T_{\mathscr{M}}(S,s) = S$ and $\mathrm{snd}(S,s) = s$, make a grading of T. This grading is canonical in the sense that it is the pseudoterminal object of $\mathbf{Grade}_{\mathscr{M}}T$. Explicitly, for every other $\mathscr{M}$-grading $(\mathscr{G},G,g)$ of T:*

- *there is a morphism $(F,f) : (\mathscr{G},G,g) \to (\mathscr{M}/T, T_{\mathscr{M}}, \mathrm{snd})$ of $\mathscr{M}$-gradings;*
- *this morphism is essentially unique in the sense that there is a natural assignment of an isomorphism $(F',f') \cong (F,f)$ to every $(F',f') : (\mathscr{G},G,g) \to (\mathscr{M}/T, T_{\mathscr{M}}, \mathrm{snd})$.*

Proof. For existence, define $(F,f) : (\mathscr{G},G,g) \to (\mathscr{M}/T, T_{\mathscr{M}}, \mathrm{snd})$ by

$$Fd = (Gd, g_d) \quad Fh = Gh \qquad f_d = \mathrm{id}_{Gd}$$

For uniqueness, given (F',f'), we have 2-cells $\beta_{(F',f')} : (F',f') \Rightarrow (F,f)$ and $\beta^{-1}_{(F',f')} : (F,f) \Rightarrow (F',f')$ given by $\beta_{(F',f'),d} = f'_d$ and $\beta^{-1}_{(F',f'),d} = f'^{-1}_d$. These are clearly natural in (F',f') and inverse to each other, so we can use β as the required natural isomorphism $(F',f') \cong (F,f)$. □

Remark 2.4. In this paper, we discuss the problem of constructing canonical gradings, but one can also consider the dual problem of constructing canonical *degradings*. For graded monads, this problem is discussed in [1, 9]. In the setting of this section, the initial degrading of a functor $G : \mathscr{G} \to \mathscr{C}$ would be the colimit $\operatorname{colim} G$, together with the morphisms $\mathrm{in}_e : Ge \to \operatorname{colim} G$ (when the colimit exists). The data $(\mathscr{G}, G, \mathrm{in})$ is then an $\mathscr{M}$-grading of $\operatorname{colim} G$ whenever in_e is in $\mathscr{M}$ for all e (which is the case for our examples). This grading will typically not be the canonical grading of $\operatorname{colim} G$ however. (For graded monads the situation is more complex: one does not take an ordinary colimit, but instead a colimit in a 2-category of monoidal categories, as discussed in [1].)

2.1 Canonical gradings of endofunctors on Set

We give several examples for the case where $\mathscr{C} = [\mathbf{Set}, \mathbf{Set}]$ and $\mathscr{M}$ is the class of natural transformations whose components are injective functions. In this case, every $\mathscr{M}$-subobject of an endofunctor $T : \mathbf{Set} \to \mathbf{Set}$ is isomorphic in $\mathscr{M}/T$ to a unique $\mathscr{M}$-subobject $s : S \rightarrowtail T$ in which each injection s_X is an inclusion. In other words, s is a choice of a subset $SX \subseteq TX$ for every set X, closed under the action of T in the sense that $x \in SX$ implies $T f x \in SY$ for every $f : X \to Y$. Below we characterize $\mathscr{M}/T$ up to equivalence for various endofunctors T, using the fact that we need only consider the case where s is a family of inclusions.

Example 2.5. The category $\mathscr{M}/\mathrm{Id}$ is equivalently the poset $\{\bot \leq \top\}$, with $\bot$ corresponding to the $\mathscr{M}$-subobject S given by $SX = \emptyset$ for all X, and $\top$ corresponding to $SX = X$ for all X.

Example 2.6. Consider the endofunctor $M \times T-$, where M is a set and T is an endofunctor on **Set**. The category $\mathscr{M}/(M \times T-)$, is equivalent to the category $(\mathscr{M}/T)^M$, in which objects are M-indexed families $(\Sigma_z \in \mathscr{M}/T)_{z \in M}$ of $\mathscr{M}$-subobjects of T. This is the case because, from every $S \rightarrowtail M \times T-$ in which each component is an inclusion, we can construct such a family $\mathbf{\Sigma}[S]$, and this construction forms a bijection with inverse $\mathbf{S}[-]$.

$$\mathbf{\Sigma}[S]_z X = \{x \in TX \mid (z,x) \in SX\} \qquad \mathbf{S}[\Sigma] X = \{(z,x) \in M \times TX \mid x \in \Sigma_z X\}$$

In the special case $T = \mathrm{Id}$, we have $\mathscr{M}/(M \times (-)) \simeq (\mathscr{M}/\mathrm{Id})^M \simeq \{\bot \leq \top\}^M \simeq (\mathscr{P}M, \subseteq)$, so the $\mathscr{M}$-subobjects of $M \times (-)$ are equivalently the subsets of M, ordered by inclusion.

Example 2.7. Consider the endofunctor $V \Rightarrow (-)$ (the underlying functor of the reader monad on **Set**), where V is a fixed set, and let $\mathscr{M}$ be the class of componentwise injective natural transformations. The $\mathscr{M}$-subobjects of $V \Rightarrow (-)$, and hence the objects of the canonical $\mathscr{M}$-grading of $V \Rightarrow (-)$, are equivalently upwards-closed sets of equivalence relations on V.

To explain this in more detail, let Equiv_V be the set of equivalence relations R on V, considered as subsets $R \subseteq V \times V$. A function $f : V \to X$ *respects* $R \in \mathrm{Equiv}_V$ when $v R v'$ implies $f v = f v'$ for all $v, v' \in V$, equivalently, when f factors through the quotient $[-]_R : V \to V/R$. A set $\Sigma \subseteq \mathrm{Equiv}_V$ of equivalence relations is *upwards-closed* when $R \in \Sigma$ implies $R' \in \Sigma$ for all $R, R' \in \mathrm{Equiv}_V$ with $R \subseteq R'$. Every such Σ induces a subfunctor $\mathbf{S}[\Sigma] \rightarrowtail V \Rightarrow (-)$, defined by

$$\mathbf{S}[\Sigma] X = \{f : V \to X \mid f \text{ respects some } R \in \Sigma\}$$

To go in the other direction, consider a subfunctor $S \rightarrowtail V \Rightarrow (-)$ in which every component of the $\mathscr{M}$-morphism is an inclusion. We obtain an upwards-closed $\mathbf{\Sigma}[S] \subseteq \mathrm{Equiv}_V$:

$$\mathbf{\Sigma}[S] = \{R \in \mathrm{Equiv}_V \mid [-]_R \in S(V/R)\}$$

This is upwards-closed because if $R \subseteq R'$ then $[-]_{R'} : V \to V/R'$ factors through $[-]_R : V \to V/R$, and since S forms a functor, the family S is closed under postcomposition. These two constructions are in bijection, with $\mathbf{S}[\mathbf{\Sigma}[S]] = S$ and $\mathbf{\Sigma}[\mathbf{S}[\Sigma]] = \Sigma$. It follows that $\mathscr{M}/(V \Rightarrow (-))$ is equivalent to the poset of upwards-closed sets $\Sigma \subseteq \mathrm{Equiv}_V$, ordered by inclusion, and hence that this poset forms the canonical $\mathscr{M}$-grading of $V \Rightarrow (-)$.

Example 2.8. Consider the endofunctor $V \Rightarrow V \times (-)$ (the underlying functor of the state monad), where V is a set. Since $V \Rightarrow V \times (-) \cong (V \Rightarrow V) \times (V \Rightarrow (-))$, we can combine Examples 2.6 and 2.7 to characterize the $\mathscr{M}$-subobjects of $V \Rightarrow V \times (-)$. Every such subobject equivalently consists of an upwards-closed set $\Sigma_p \subseteq \mathrm{Equiv}_V$ for each function $p : V \to V$. These can also be seen as subsets $\Sigma \subseteq (V \Rightarrow V) \times \mathrm{Equiv}_V$ such that $\{R \mid (p,R) \in \Sigma\}$ is upwards-closed for each $p : V \to V$. Given such a Σ, the corresponding $\mathscr{M}$-subobject $\mathbf{S}[\Sigma] \rightarrowtail (V \Rightarrow V \times (-))$ is

$$\mathbf{S}[\Sigma]X = \{f : V \to V \times X \mid \exists (p,R) \in \Sigma.\ \pi_1 \circ f = p \wedge \pi_2 \circ f \text{ respects } R\}$$

3 Grading monoids and monads

We proceed to grading monoids. To define the notion of grading of a monoid, we need an appropriate multiplication operation on the grades. The obvious idea to to ask for the grades to form a monoidal category instead of just a category, and much of the previous work on graded monads (such as [10]) does exactly this. However, in some of examples we do not get a monoidal category of grades, but only a *skew* monoidal category [14] of grades.

Definition 3.1. A *(left-)skew monoidal* category is a category $\mathscr{C}$ with a distinguished object I, a functor $\otimes : \mathscr{C} \times \mathscr{C} \to \mathscr{C}$ and three natural transformations λ, ρ, α typed

$$\lambda_X : \mathsf{I} \otimes X \to X \qquad \rho_X : X \to X \otimes \mathsf{I} \qquad \alpha_{X,Y,Z} : (X \otimes Y) \otimes Z \to X \otimes (Y \otimes Z)$$

satisfying the equations

$$\text{(m1)}\quad \begin{array}{ccc} & \mathsf{I}\otimes\mathsf{I} & \\ {\scriptstyle\rho_{\mathsf{I}}}\nearrow & & \searrow{\scriptstyle\lambda_{\mathsf{I}}} \\ \mathsf{I} & = & \mathsf{I} \end{array} \qquad \text{(m2)}\quad \begin{array}{ccc} (X\otimes\mathsf{I})\otimes Y & \xrightarrow{\alpha_{X,\mathsf{I},Y}} & X\otimes(\mathsf{I}\otimes Y) \\ {\scriptstyle\rho_X\otimes Y}\uparrow & & \downarrow{\scriptstyle X\otimes\lambda_Y} \\ X\otimes Y & = & X\otimes Y \end{array}$$

$$\text{(m3)}\quad \begin{array}{ccc} (\mathsf{I}\otimes X)\otimes Y & \xrightarrow{\alpha_{\mathsf{I},X,Y}} & \mathsf{I}\otimes(X\otimes Y) \\ {\scriptstyle\lambda_X\otimes Y}\searrow & & \swarrow{\scriptstyle\lambda_{X\otimes Y}} \\ & X\otimes Y & \end{array} \qquad \text{(m4)}\quad \begin{array}{ccc} (X\otimes Y)\otimes\mathsf{I} & \xrightarrow{\alpha_{X,Y,\mathsf{I}}} & X\otimes(Y\otimes\mathsf{I}) \\ {\scriptstyle\rho_{X\otimes Y}}\nwarrow & & \nearrow{\scriptstyle X\otimes\rho_Y} \\ & X\otimes Y & \end{array}$$

$$\text{(m5)}\quad \begin{array}{ccccc} (X\otimes(Y\otimes Z))\otimes W & & \xrightarrow{\alpha_{X,Y\otimes Z,W}} & & X\otimes((Y\otimes Z)\otimes W) \\ {\scriptstyle\alpha_{X,Y,Z}\otimes W}\uparrow & & & & \downarrow{\scriptstyle X\otimes\alpha_{Y,Z,W}} \\ ((X\otimes Y)\otimes Z)\otimes W & \xrightarrow{\alpha_{X\otimes Y,Z,W}} & (X\otimes Y)\otimes(Z\otimes W) & \xrightarrow{\alpha_{X,Y,Z\otimes W}} & X\otimes(Y\otimes(Z\otimes W)) \end{array}$$

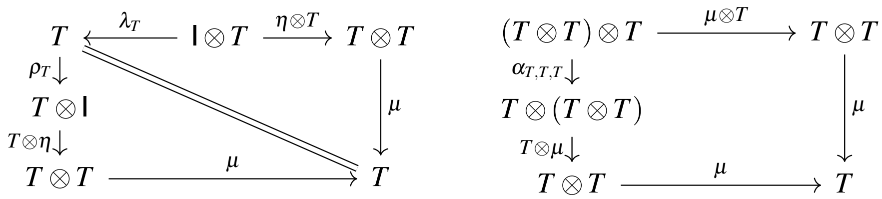
$(\mathscr{C}, \mathsf{I}, \otimes)$ is partially normal if one or several of λ, ρ or α is a natural isomorphism. In particular, it is *left-normal* if λ is an isomorphism. A monoidal category is a fully normal skew monoidal category.

A *right-skew monoidal category* is given by $(\mathscr{C}, \mathsf{I}, \otimes, \lambda, \rho, \alpha)$ such that the data $(\mathscr{C}, \mathsf{I}, \otimes^{\mathrm{rev}}, \rho, \lambda, \alpha)$, where $X \otimes^{\mathrm{rev}} Y = Y \otimes X$, form a left-skew monoidal category.

Definition 3.2. A *monoid* in a skew monoidal category $(\mathscr{C}, \mathsf{I}, \otimes)$ is an object T of $\mathscr{C}$ equipped with morphisms

$$\eta : \mathsf{I} \to T \qquad \mu : T \otimes T \to T$$

satisfying the equations



The concept of lax monoidal functor between skew monoidal categories is defined as for monoidal categories; the same applies to the concept of monoidal transformations between lax monoidal functors.

Definition 3.3. Given a skew monoidal category $\mathbb{G} = (\mathscr{G}, I, \odot)$, a $\mathbb{G}$-*graded monoid* in a skew monoidal category $\mathbb{C} = (\mathscr{C}, \mathsf{I}, \otimes)$ is the same as a lax monoidal functor $\mathsf{G} : \mathbb{G} \to \mathbb{C}$. Explicitly, it is a functor $G : \mathscr{G} \to \mathscr{C}$ with a morphism $\eta : \mathsf{I} \to GI$ and a natural transformation typed $\mu_{d,d'} : Gd \otimes Gd' \to G(d \odot d')$ subject to equations similar to those of a monoid.

Definition 3.4. Let $\mathsf{T} = (T, \eta, \mu)$ be a monoid in a skew monoidal category $\mathbb{C} = (\mathscr{C}, \mathsf{I}, \otimes)$, and let $\mathscr{M}$ be a class of morphisms of $\mathscr{C}$. An $\mathscr{M}$-*grading* $(\mathbb{G}, \mathsf{G}, g)$ of the monoid T consists of a skew monoidal category $\mathbb{G}$, a lax monoidal functor $\mathsf{G} : \mathbb{G} \to \mathbb{C}$ (= a $\mathbb{G}$-graded monoid in $\mathbb{C}$), and a monoidal transformation typed $g_d : Gd \rightarrowtail T$, whose components are all in $\mathscr{M}$. A *morphism* $(\mathsf{F}, f) : (\mathbb{G}, \mathsf{G}, g) \to (\mathbb{G}', \mathsf{G}', g')$ between such gradings is a lax monoidal functor $\mathsf{F} : \mathbb{G} \to \mathbb{G}'$ equipped with a monoidal isomorphism $f : \mathsf{G}' \cdot \mathsf{F} \cong \mathsf{G}$, such that $g_d \circ f_d = g'_{Fd}$. A *2-cell* $\beta : (\mathsf{F}, f) \Rightarrow (\mathsf{F}', f')$ is a monoidal transformation $\beta : \mathsf{F} \Rightarrow \mathsf{F}'$ such that $f' \circ (\mathsf{G}' \cdot \beta) = f$. We write $\mathbf{Grade}_{\mathscr{M}}\mathsf{T}$ for this 2-category.

Example 3.5. The situation we are mainly interested in is when $\mathscr{C} = [\mathscr{D}, \mathscr{D}]$ is the category of endofunctors on some $\mathscr{D}$, with the identity for I and functor composition for $\otimes$. In this case, monoids in $\mathbb{C}$ are exactly monads on $\mathscr{D}$, and a lax monoidal functor $\mathsf{G} : \mathbb{G} \to \mathbb{C}$ (a $\mathbb{G}$-graded monoid in $\mathbb{C}$) is a $\mathbb{G}$-graded monad on $\mathscr{D}$, in the sense of [13, 10, 4]. Explicitly, the unit and multiplication of G have the form

$$\eta_X : X \to GIX \qquad \mu_{e,e',X} : Ge(Ge'X) \to G(e \odot e')X$$

For a concrete example, let V be a set (of states), and let T be the state monad over V:

$$TX = V \Rightarrow V \times X \qquad \eta_X\, x\, v = (v, x) \qquad \mu_X\, f\, v = g\, v' \;\text{ where } (v', g) = f\, v$$

We give a $\mathscr{M}$-grading $(\mathbb{G}, \mathsf{G}, g)$ of T, where $\mathscr{M}$ is componentwise injective natural transformations. Let $\mathscr{G}$ be the poset of subsets of $\{\mathsf{get}, \mathsf{put}\}$ ordered by inclusion, which forms a strict monoidal category with $\emptyset$ for the unit I and $e \cup e'$ for the tensor $e \odot e'$. We then define G by

$$\begin{aligned} Ge = \{ f : V \to V \times X \\ &\mid \mathsf{get} \notin e \Rightarrow (\pi_1 \circ f \text{ is a constant function or } \mathrm{id}_V \;\wedge\; \pi_2 \circ f \text{ is a constant function}) \\ &\;\wedge\; \mathsf{put} \notin e \Rightarrow \pi_1 \circ f \text{ is } \mathrm{id}_V \} \end{aligned}$$

with unit and multiplication defined as for T. This forms an $\mathscr{M}$-grading with the inclusions for g.

This grading is suitable for interpreting a Gifford-style effect system [8] for global state. A function $f : V \to V \times X$ is a computation $f \in TX$, sending an initial state to a pair of a final state and a result. A grade e gives the set of operations that a computation may use when it is executed, so GeX is the subset of TX on the computation that only use the operations in e. For example, $G\{\mathsf{get}\}X$ contains computations that may use the initial state, but do not change the state (with put).

We turn now to the *canonical* grading of a monoid T in a skew monoidal category $\mathbb{C}$. Since the category $\mathscr{M}/T$ forms the canonical grading of the *object* T, we show that (under the conditions explained below), we can make $\mathscr{M}/T$ into a skew monoidal category, using the monoid structure of T.

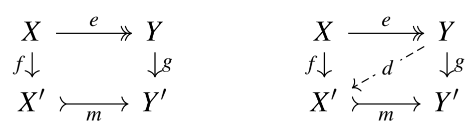
First consider the slice category $\mathscr{C}/T$ (where we do not restrict to $\mathscr{M}$-morphisms). This already forms a skew monoidal category, with

$$\mathsf{I} \xrightarrow{\eta} T \qquad S \otimes S' \xrightarrow{s \otimes s'} T \otimes T \xrightarrow{\mu} T$$

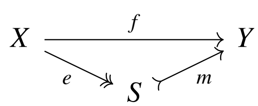
for the unit and tensor of (S,s) and (S',s') (see for example Kelly [6]). In general this skew monoidal structure will not restrict to $\mathscr{M}/T$, because the morphism μ is not in $\mathscr{M}$ for many of our examples. However, we can make $\mathscr{M}/T$ into a skew monoidal category by adapting the skew monoidal structure on $\mathscr{C}/T$. The idea is to just factorize the morphisms we use in the tensor and unit of $\mathscr{C}/T$ to obtain morphisms in $\mathscr{M}$. Hence we ask that $\mathscr{M}$ forms an orthogonal factorization system in the usual sense.

Definition 3.6. An (orthogonal) *factorization system* on a category $\mathscr{C}$ is a pair $(\mathscr{E},\mathscr{M})$ of classes of morphisms of $\mathscr{C}$, such that

- both $\mathscr{E}$ and $\mathscr{M}$ contain all isomorphisms, and are closed under composition;
- $\mathscr{E}$-morphisms are *orthogonal* to $\mathscr{M}$-morphisms: for every commuting square in $\mathscr{C}$ as on the left below, with $e \in \mathscr{E}$ and $m \in \mathscr{M}$, there is a unique d making the diagram on the right below commute.



- every morphism $f : X \to Y$ in $\mathscr{C}$ has an $(\mathscr{E},\mathscr{M})$-factorization: there exist an object S, an $\mathscr{E}$-morphism $e : X \twoheadrightarrow S$, and an $\mathscr{M}$-morphism $m : S \rightarrowtail Y$ such that the diagram below commutes.



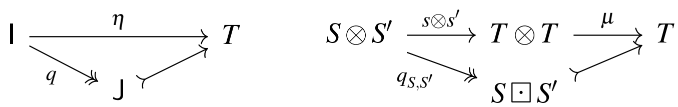
Example 3.7. Writing **Mor** for the class of all morphisms and **Iso** for the class of isomorphisms, every category has $(\mathbf{Mor},\mathbf{Iso})$ and $(\mathbf{Iso},\mathbf{Mor})$ as factorization systems. On **Set**, the classes of surjective and of injective functions form a factorization system $(\mathbf{Surj},\mathbf{Inj})$. To factorize a function $f : X \to Y$ in this case, we let $S = \{fx \mid x \in X\}$ be the image of f, and define $X \xrightarrow{e} S \xrightarrow{m} Y$ by $ex = fx$ and $my = y$. On the category **Poset** of partially ordered sets and monotone functions, we have a factorization system $(\mathbf{Surj},\mathbf{Full})$, where **Surj** is the class of surjective monotone functions and **Full** is the class of full functions, i.e. monotone functions $m : S \to Y$ such that $mx \le my$ implies $x \le y$. Factorizations are given as in **Set**, with the order on S inherited from the order on Y.

We are primarily interested in canonically grading monads, which are monoids in the endofunctor category $\mathscr{C} = [\mathscr{D},\mathscr{D}]$, with functor composition as the tensor. We therefore want a factorization system on $[\mathscr{D},\mathscr{D}]$. We give the most standard option for this as the following example, but there are others we are interested in (see Lemma 4.2 below). In models of computational effects we in fact usually want a *strong* monad. If $\mathscr{D}$ is monoidal, then a *strong endofunctor* on $\mathscr{D}$ is a functor $F : \mathscr{D} \to \mathscr{D}$ equipped with a *strength*, i.e. a natural transformation $\mathrm{str}_{\Gamma,X} : \Gamma \otimes_{\mathscr{D}} FX \to F(\Gamma \otimes_{\mathscr{D}} X)$ satisfying two laws for compatibility with the left unitor and associator of $\mathscr{D}$. These form a monoidal category $[\mathscr{D},\mathscr{D}]_s$, in which morphisms are strength-preserving natural transformations and the tensor is composition. Strong monads are monoids in $[\mathscr{D},\mathscr{D}]_s$. Below we consider non-strong monads for simplicity, but we can also apply our results to strong monads using a factorization system on $[\mathscr{D},\mathscr{D}]_s$.

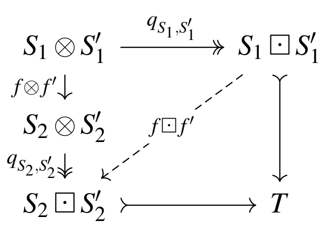
Example 3.8. If $(\mathscr{E},\mathscr{M})$ is a factorization system on a category $\mathscr{D}$, then the endofunctor category $[\mathscr{D},\mathscr{D}]$ has a factorization system (componentwise-$\mathscr{E}$, componentwise-$\mathscr{M}$). Factorizations $F \twoheadrightarrow S \rightarrowtail G$ of natural transformations are componentwise. If $\mathscr{D}$ is monoidal and $\mathscr{E}$ is closed under $\Gamma \otimes_{\mathscr{D}} (-)$ for all Γ then (componentwise-$\mathscr{E}$, componentwise-$\mathscr{M}$) is a factorization system on $[\mathscr{D},\mathscr{D}]_s$. Morphisms are again factorized componentwise; for the construction of the strength for S see [3, Section 2.2].

Forming a factorization system $(\mathscr{E},\mathscr{M})$ is merely a property of a class $\mathscr{M}$ of morphisms, because $\mathscr{E}$ is necessarily the class of all morphisms e that are orthogonal to all $\mathscr{M}$-morphisms. Factorizations of morphisms are unique up to unique isomorphism.

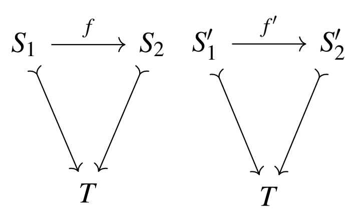
If $\mathscr{M}$ forms a factorization system $(\mathscr{E},\mathscr{M})$, then for a given monoid T we construct a unit J and a tensor $\boxdot$ for the category $\mathscr{M}/T$ by factorizing morphisms as follows:



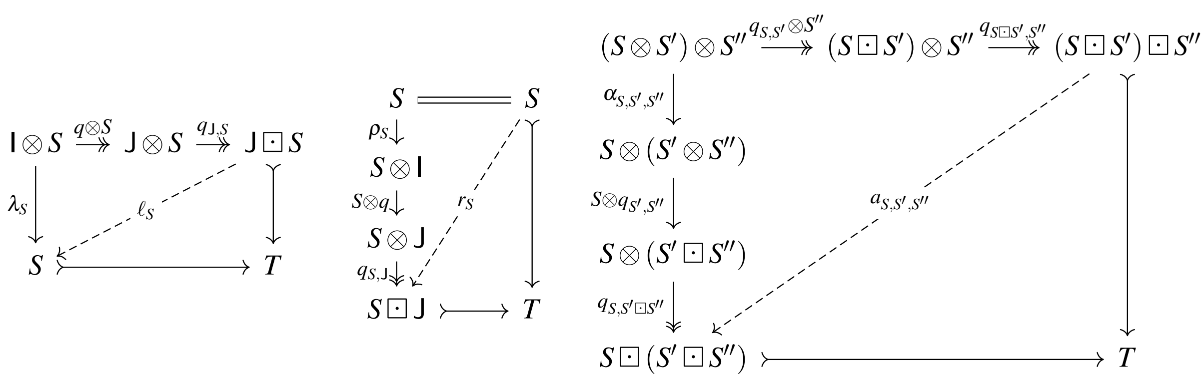
To construct the required structural morphisms, we make the additional assumption that $\mathscr{E}$ is closed under $(-) \otimes S$ for every $S \rightarrowtail T$. Under this assumption, the following squares, which all commute, have a unique diagonal, and these diagonals are the required structural morphisms.



where



are morphisms in $\mathscr{M}/T$



If λ is a natural isomorphism, then so is ℓ. This does not apply to ρ and α unless $\mathscr{E}$ is also closed under $S \otimes (-)$ for all $S \rightarrowtail T$.

Theorem 3.9. *Let T be a monoid in a skew monoidal category $\mathbb{C} = (\mathscr{C}, \mathsf{I}, \otimes)$, and let $(\mathscr{E},\mathscr{M})$ be a factorization system on $\mathscr{C}$. If $\mathscr{E}$ is closed under $(-) \otimes S$ for every $\mathscr{M}$-subobject $S \rightarrowtail T$, then $\mathscr{M}/\mathsf{T} = (\mathscr{M}/T, \mathsf{J}, \boxdot)$ is a skew monoidal category. If $\mathbb{C}$ is left-normal, then so is $\mathscr{M}/\mathsf{T}$. If $\mathscr{E}$ is also closed under $S \otimes (-)$ for every $S \rightarrowtail T$ and $\mathbb{C}$ is monoidal, then $\mathscr{M}/\mathsf{T}$ is monoidal.*

Proof. See Appendix A. □

Remark 3.10. Closure of $\mathscr{E}$ under $T \otimes (-)$ does not in general imply closure of $\mathscr{E}$ under $S \otimes (-)$ for $S \rightarrowtail T$. Consider the factorization system $(\mathscr{E}, \mathscr{M}) = (\text{componentwise surjective}, \text{componentwise full})$ on $[\mathbf{Poset}, \mathbf{Poset}]$, with composition as $\otimes$. For an endofunctor $F : \mathbf{Poset} \to \mathbf{Poset}$, closure of $\mathscr{E}$ under $F \otimes (-)$ amounts to closure of **Surj** under F. The class **Surj** is closed under $V \Rightarrow (-)$ exactly when V is discrete. Hence, while this property holds for $\{0,1\} \Rightarrow (-)$, it does not hold for $\{0 \leq 1\} \Rightarrow (-) \rightarrowtail \{0,1\} \Rightarrow (-)$.

There are cases in which $\mathscr{E}$ is closed under $S \otimes (-)$ for all $S \rightarrowtail T$ even if $\mathscr{E}$ is not closed under $F \otimes (-)$ for general F: for example every $\mathscr{M}$-subobject of the endofunctor $M \times (-)$ on **Poset** has the form $S \times (-)$ for some $S \rightarrowtail M$, and functors of the form $S \times (-)$ send surjections to surjections.

Our task is now to show that the skew monoidal category $\mathscr{M}/\mathsf{T} = (\mathscr{M}/T, \mathsf{J}, \boxdot)$ forms the canonical grading $(\mathscr{M}/\mathsf{T}, \mathsf{T}_{\mathscr{M}}, \mathrm{snd})$ of the monoid T when $\mathscr{E}$ is closed under $S \otimes (-)$ for each $S \rightarrowtail T$. The lax monoidal functor $\mathsf{T}_{\mathscr{M}} : \mathscr{M}/\mathsf{T} \to \mathbb{C}$ is given on objects by

$$T_{\mathscr{M}}(S, s : S \rightarrowtail T) = S$$

and has as unit and multiplication the $\mathscr{E}$-morphisms from the construction of J and $\boxdot$:

$$\mathsf{I} \xrightarrow{q} \!\!\!\!\rightarrow T_{\mathscr{M}}\mathsf{J} \qquad\qquad T_{\mathscr{M}}(S,s) \otimes T_{\mathscr{M}}(S',s') \xrightarrow{q_{S,S'}} \!\!\!\!\rightarrow T_{\mathscr{M}}((S,s) \boxdot (S',s'))$$

That this is lax monoidal is immediate from the definition of the structural morphisms of $\mathscr{M}/\mathsf{T}$. Finally, the monoidal transformation $\mathrm{snd} : \mathsf{T}_{\mathscr{M}} \Rightarrow \mathsf{T}$ is given by $\mathrm{snd}_{(S,s)} = s$. Monoidality of snd is immediate from the definitions of J and $\boxdot$. Hence $(\mathscr{M}/\mathsf{T}, \mathsf{T}_{\mathscr{M}}, \mathrm{snd})$ is a grading of T. Canonicity is the following theorem.

Theorem 3.11. *Let T be a monoid in a skew monoidal category $\mathbb{C} = (\mathscr{C}, \mathsf{I}, \otimes)$, and let $(\mathscr{E}, \mathscr{M})$ be a factorization system on $\mathscr{C}$ such that $\mathscr{E}$ is closed under $(-) \otimes S$ for each $\mathscr{M}$-subobject S of T. The grading $(\mathscr{M}/\mathsf{T}, \mathsf{T}_{\mathscr{M}}, \mathrm{snd})$ is canonical in the sense that it is the pseudoterminal object of* $\mathbf{Grade}_{\mathscr{M}}\mathsf{T}$. *Explicitly, for every $\mathscr{M}$-grading $(\mathbb{G}, \mathsf{G}, g)$ of the monoid T:*

- *there is a morphism $(\mathsf{F}, f) : (\mathbb{G}, \mathsf{G}, g) \to (\mathscr{M}/\mathsf{T}, \mathsf{T}_{\mathscr{M}}, \mathrm{snd})$, of $\mathscr{M}$-gradings of T;*
- *this morphism is essentially unique in the sense that there is a natural assignment of an isomorphism $(\mathsf{F}', f') \cong (F, f)$ to every $(\mathsf{F}', f') : (\mathbb{G}, \mathsf{G}, g) \to (\mathscr{M}/\mathsf{T}, \mathsf{T}_{\mathscr{M}}, \mathrm{snd})$.*

Proof. We have done most of the proof already as Theorem 2.3; we fill in the remaining parts. Recall from there that we define

$$Fd = (Gd, g_d) \quad Fh = Gh \qquad f_d = \mathrm{id}_{Gd}$$

We make F into a lax monoidal functor by using the unique diagonals of the following squares as the unit and multiplication. The squares commute because G is lax monoidal.

$$\begin{array}{ccc} \mathsf{I} & \xrightarrow{q} & \mathsf{J} \\ {\scriptstyle \eta}\downarrow & \swarrow_{\eta} & \downarrow \\ GI & \underset{g_I}{\rightarrowtail} & T \end{array} \qquad\qquad \begin{array}{ccc} Gd \otimes Gd' & \xrightarrow{q_{Gd,Gd'}} & Gd \boxdot Gd' \\ {\scriptstyle \mu_{d,d'}}\downarrow & \swarrow_{\mu_{d,d'}} & \downarrow \\ G(d \odot d') & \underset{g_{d \odot d'}}{\rightarrowtail} & T \end{array}$$

This definition immediately implies that f is monoidal, and hence that (F, f) is a morphism of $\mathscr{M}$-gradings of T. Finally, given (F', f'), we show that $\beta_d = f'_d$ defines an isomorphism $\beta : (\mathsf{F}', f') \cong (\mathsf{F}, f)$. For this it remains to show that β is monoidal (it follows automatically that the inverse β^{-1} is monoidal). For compatibility with the multiplications, this amounts to showing that the square on the left below commutes. For this it is enough to show both paths in that square provide the unique diagonal of the square on the right, and this follows from the fact that f' is monoidal.

$$
\begin{array}{ccc}
F'd \boxdot F'd' & \xrightarrow{f'_d \boxdot f'_{d'}} & Gd \boxdot Gd' \\
\downarrow{\scriptstyle \mu_{d,d'}} & & \downarrow{\scriptstyle \mu_{d,d'}} \\
F'(d \odot d') & \xrightarrow[f'_{d \odot d'}]{} & G(d \odot d')
\end{array}
\qquad
\begin{array}{ccc}
F'd \otimes F'd' & \overset{q_{F'd,F'd'}}{\twoheadrightarrow} & F'd \boxdot F'd' \\
\downarrow{\scriptstyle f'_d \otimes f'_{d'}} & \swarrow & \\
Gd \otimes Gd' & & \downarrow \\
\downarrow{\scriptstyle \mu_{d,d'}} & & \\
G(d \odot d') & \xrightarrow[g_{d \boxdot d'}]{} & T
\end{array}
$$

Compatibility with the units is similar. □

Example 3.12. Let $(M, \varepsilon, \cdot)$ be a monoid in the cartesian monoidal category **Set**, and let T be the corresponding writer monad, which has endofunctor $TX = M \times X$, unit $\eta_X x = (\varepsilon, x)$, and multiplication $\mu_X(z, (z', x)) = (z \cdot z', x)$. Then T is a monoid in the monoidal category of endofunctors on **Set**, with functor composition. Consider the factorization system $(\mathscr{E}, \mathscr{M})$ on $[\mathbf{Set}, \mathbf{Set}]$ in which $\mathscr{E}$ (respectively $\mathscr{M}$) is componentwise surjective (resp. injective) natural transformations. The class $\mathscr{E}$ is closed under functor composition on both sides, so $\mathscr{M}/\mathsf{T}$ is monoidal and provides the canonical grading of T. We show in Example 2.6 that $\mathscr{M}$-subobjects of T are equivalently subsets $\Sigma \subseteq M$. Under this equivalence, the monoidal structure on $\mathscr{M}/\mathsf{T}$ is given by $\mathsf{J} = \{\varepsilon\}$ and $\Sigma \boxdot \Sigma' = \{z \cdot z' \mid z \in \Sigma, z' \in \Sigma'\}$. The graded monad $\mathsf{T}_{\mathscr{M}}$ is given by

$$T_{\mathscr{M}}\Sigma = \Sigma \times X \qquad \eta_X x = (\varepsilon, x) \qquad \mu_{\Sigma,\Sigma',X}(z, (z', x)) = (z \cdot z', x)$$

Example 3.13. We show in Example 2.8 that, when $\mathscr{M}$ is componentwise injective natural transformations and $T = V \Rightarrow V \times (-)$, the objects of $\mathscr{M}/T$ are equivalently subsets $\Sigma \subseteq (V \Rightarrow V) \times \mathrm{Equiv}_V$ satisfying a closure condition. When T is the state monad, these form a monoidal category $\mathscr{M}/\mathsf{T}$, and the graded monad $\mathsf{T}_{\mathscr{M}}$ has underlying functor

$$T_{\mathscr{M}}\Sigma X = \{f : V \to V \times X \mid \exists (p, R) \in \Sigma.\, p = \pi_1 \circ f \wedge (\pi_2 \circ f) \text{ respects } R\}$$

Example 3.5 provides another grading of T, in which the grades are subsets of $\{\mathsf{get}, \mathsf{put}\}$. By Theorem 3.11, we obtain a morphism (F, f) of gradings. Under the characterization of grades as subsets Σ, the underlying functor F sends $e \subseteq \{\mathsf{get}, \mathsf{put}\}$ to $Fe \subseteq (V \Rightarrow V) \times \mathrm{Equiv}_V$ as follows:

$$F\emptyset = \{(\mathrm{id}_V, V \times V)\} \qquad F\{\mathsf{get}, \mathsf{put}\} = (V \Rightarrow V) \times \mathrm{Equiv}_V$$
$$F\{\mathsf{get}\} = \{(\mathrm{id}_V, R) \mid R \in \mathrm{Equiv}_V\} \qquad F\{\mathsf{put}\} = \{(p, V \times V) \mid p \text{ is a constant function or } \mathrm{id}_V\}$$

4 Canonical grading by sets of shapes

When assigning grades to computations $t \in TX$, where T is a monad on **Set**, we are often interested only in the *shape* of the computation. A *shape* is an element of $T1$, where 1 is the one-element set; and the shape of the computation $t \in TX$ is $T!t \in T1$, where ! is the unique function $X \to 1$. A grade in this case is a subset $e \subseteq T1$ of the set of shapes, and a computation has grade e when its shape $T!t$ is in e.

More generally, if T is a monad on a category $\mathscr{D}$ with a terminal object 1, then the object of shapes is $T1$. Given a class $\mathscr{M}$ of morphisms of $\mathscr{D}$, we can consider grading by $\mathscr{M}$-subobjects of $T1$. We show in this section that these grades can be considered canonical, using a suitable class $\mathscr{M}'$ of morphisms of $[\mathscr{D}, \mathscr{D}]$.

Definition 4.1. Let $\mathscr{A}$ be a category. A natural transformation $f : F \Rightarrow G : \mathscr{A} \to \mathscr{D}$ is *cartesian* if all of its naturality squares are pullbacks. If $\mathscr{A}$ has pullbacks, then a functor $F : \mathscr{A} \to \mathscr{D}$ is *cartesian* when it preserves pullbacks.

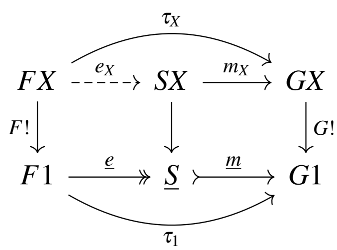
Lemma 4.2. *Let $(\mathscr{E}, \mathscr{M})$ be a factorization system on a category $\mathscr{D}$ with pullbacks, and let $\mathscr{A}$ be a category with a terminal object. Then we have a factorization system $(\mathscr{E}', \mathscr{M}')$ on $[\mathscr{A}, \mathscr{D}]$ as follows:*

$$\mathscr{E}' = \text{natural transformations } e \text{ such that } e_1 \in \mathscr{E}$$
$$\mathscr{M}' = \text{cartesian natural transformations } m \text{ such that } m_1 \in \mathscr{M}$$

For every functor $G : \mathscr{A} \to \mathscr{D}$, there is an equivalence of categories $\mathscr{M}'/G \simeq \mathscr{M}/G1$.

Before giving the proof, we note that Kelly [6] considers $(\mathscr{E}', \mathscr{M}')$ in the case $(\mathscr{E}, \mathscr{M}) = (\mathbf{Iso}, \mathbf{Mor})$.

Proof. That $\mathscr{E}'$ and $\mathscr{M}'$ are closed under isomorphisms and composition is straightforward. To factorize a natural transformation $\tau : F \Rightarrow G$, we first factorize τ_1 using $(\mathscr{E}, \mathscr{M})$, as on the bottom of the following diagram.



We then factorize any component τ_X as on the top, by taking as (SX, m_X) the pullback of $(\underline{S}, \underline{m})$ along $G!$, and taking as e_X the unique map from FX to this pullback. The objects SX form a functor using unique maps into pullbacks, and the morphisms e_X and m_X are natural transformations. When $X = 1$ we have $e_X \in \mathscr{E}$ and $m_X \in \mathscr{M}$ because the vertical morphisms in the diagram are all isomorphisms. Hence we have the required factorization of τ into $e \in \mathscr{E}'$ and $m \in \mathscr{M}'$. For orthogonality, unique diagonal fill-ins are constructed as unique maps to pullbacks.

The required equivalence of categories exists because every $\mathscr{M}'$-subobject $m : S \rightarrowtail G$ is determined up to isomorphism by the component m_1. The latter is the corresponding object of $\mathscr{M}/G1$. □

Lemma 4.2 provides a construction of a factorization system $(\mathscr{E}', \mathscr{M}')$ in particular on endofunctor categories $[\mathscr{D}, \mathscr{D}]$ when $\mathscr{D}$ has pullbacks and a terminal object. In this case the $\mathscr{M}'$-subobjects of T are $\mathscr{M}$-subobjects of $T1$. However, $\mathscr{E}'$ is often closed under neither $(-) \cdot S$ nor $S \cdot (-)$ for $S \rightarrowtail T$, and $\mathscr{M}'/T$ is neither left-skew nor right-skew monoidal for a monad T. In the following example, left-skew monoidality fails, but we do get right-skew monoidality.

Example 4.3. Consider the factorization system $(\mathscr{E}, \mathscr{M}) = (\mathbf{Surj}, \mathbf{Inj})$ on $\mathscr{A} = \mathscr{D} = \mathbf{Set}$. Surjections in $\mathbf{Set}$ are preserved by any functor S, therefore the class $\mathscr{E}'$ of the factorization system $(\mathscr{E}', \mathscr{M}')$ on $[\mathbf{Set}, \mathbf{Set}]$ is closed under $S \cdot (-)$ for any S (as $(S \cdot e)_1 = Se_1$). Given a set monad T, the category $\mathscr{M}'/T$ obtains a right-skew monoidal structure by the "reversal" of Theorem 3.9.

Let T be the state monad for a set of states V:

$$T = V \Rightarrow V \times (-) \qquad \eta_X\, x\, v = (v, x) \qquad \mu_X\, f\, v = g\, v' \;\text{ where } (v', g) = f\, v$$

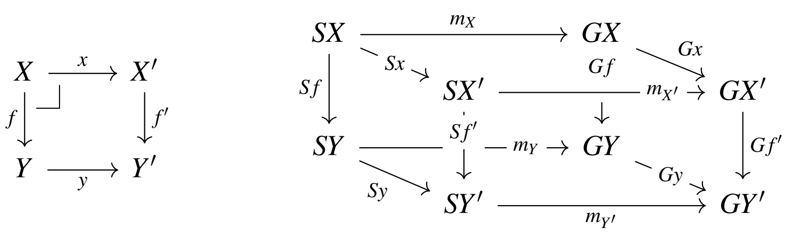
Since $T1 \cong V \Rightarrow V$, we have $\mathscr{M}'/T \simeq \mathbf{Inj}/(V \Rightarrow V)$, so that the canonical grades are equivalently subsets of $\Sigma \subseteq V \Rightarrow V$, ordered by inclusion. A subset Σ corresponds to the $\mathscr{M}'$-subobject $\mathbf{S}[\Sigma] \rightarrowtail T$ given by

$\mathbf{S}[\Sigma]X = \{f : V \to V \times X \mid \pi_1 \circ f \in \Sigma\}$. Given such a subset, define $\mathrm{Cl}(\Sigma') = \{f : V \to V \times X \mid \forall v. \exists g \in \Sigma'. \pi_1(f\,v) = g\,v\}$. The right-skew monoidal category of canonical grades Σ has unit $\mathsf{J} = \{\mathrm{id}_V\}$ and tensor $\Sigma \boxdot \Sigma' = \Sigma \,\hat{\circ}\, \mathrm{Cl}(\Sigma')$, where $\Sigma \,\hat{\circ}\, \Sigma' = \{f' \circ f \mid f \in \Sigma, f' \in \Sigma'\}$. There is no left unitor for a left-skew monoidal structure, because $\mathsf{J} \boxdot \Sigma' = \mathrm{Cl}(\Sigma')$ is not in general equal to Σ'.

The failure of left-skew monoidality in this example can be traced back to the failure of $\mathscr{E}'$ to be closed under $(-) \cdot S$ for endofunctors $S \rightarrowtail T$. We had no such problem for the componentwise lifting of $(\mathscr{E}, \mathscr{M})$. When $(\mathscr{E}, \mathscr{M})$ is *stable* (Definition 4.6 below) and one restricts $[\mathscr{A}, \mathscr{D}]$ to cartesian natural transformations (and optionally further also to cartesian functors), then the componentwise lifting actually coincides with $(\mathscr{E}', \mathscr{M}')$, as we show in the next few lemmata. This then provides sufficient conditions for $\mathscr{E}'$ to be closed under $(-) \cdot S$. We give an example in which these conditions are satisfied in Example 4.10 below, where we actually obtain a monoidal structure on the category of grades.

Lemma 4.4. *If $\mathscr{A}$ has pullbacks, $m : S \Rightarrow G : \mathscr{A} \to \mathscr{D}$ is a cartesian natural transformation, and G is cartesian, then S is also cartesian.*

Proof. Every pullback square in $\mathscr{A}$, as on the left below, induces a cube in $\mathscr{D}$, as on the right below. Four of the faces of this cube are pullbacks because m is cartesian, and the face on the right is a pullback because G is cartesian. It follows that the left face is also a pullback.



□

Lemma 4.5. *If $\mathscr{A}$ has pullbacks, then any factorization system on $[\mathscr{A}, \mathscr{D}]_{\mathrm{cartnt}}$ (all functors, but only cartesian natural transformations) restricts to a factorization system on $[\mathscr{A}, \mathscr{D}]_{\mathrm{cart}}$ (cartesian functors and cartesian natural transformations).*

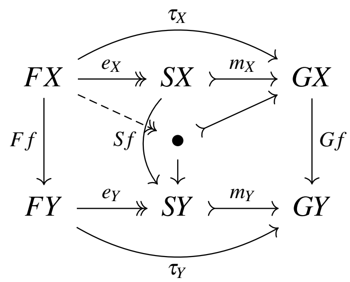
Proof. It suffices to show that $[\mathscr{A}, \mathscr{D}]_{\mathrm{cart}}$ is closed under factorizations of cartesian natural transformations. This a consequence of the previous lemma: if $\tau : F \Rightarrow G$ is a morphism in $[\mathscr{A}, \mathscr{D}]_{\mathrm{cart}}$ that factorizes as (S, e, m), then S is cartesian because G and m are. □

Definition 4.6. If $\mathscr{D}$ has pullbacks, then a factorization system $(\mathscr{E}, \mathscr{M})$ on $\mathscr{D}$ is *stable* when $\mathscr{E}$ is closed under pullbacks along arbitrary morphisms.

(The analogous property for $\mathscr{M}$ is true in every factorization system.)

Lemma 4.7. *Assume that $\mathscr{D}$ has pullbacks, and let $(\mathscr{E}, \mathscr{M})$ be a stable factorization system on $\mathscr{D}$. The componentwise lifting of $(\mathscr{E}, \mathscr{M})$ to $[\mathscr{A}, \mathscr{D}]$ restricts to a factorization system on $[\mathscr{A}, \mathscr{D}]_{\mathrm{cartnt}}$.*

Proof. We need to check that, if a cartesian natural transformation $\tau : F \Rightarrow G$ factorizes as (S, e, m) using $(\mathscr{E}, \mathscr{M})$ componentwise, then e and m are cartesian natural transformations. Given any $f : X \to Y$, we can consider the naturality square of τ for f, which is by assumption cartesian. It breaks into naturality squares of e and m for f. We can then pull back (SY, m_Y) along Gf and be certain that the resulting morphism is in $\mathscr{M}$. The unique morphism from FX to the pullback vertex $\bullet$ is a pullback of (FY, e_Y) along Sf by the pullback lemma, and is therefore in $\mathscr{E}$ by stability. We therefore have two factorizations of τ_X: one through SX and one through $\bullet$. Factorizations are unique up to isomorphism, and hence the naturality squares of both e and m are pullbacks.



We also need to check that the diagonal fill-ins of cartesian squares built using $(\mathscr{E},\mathscr{M})$ componentwise are cartesian. This holds because the pullback lemma provides a two-out-of-three property for cartesian natural transformations: if $m \circ d$ and m are cartesian, then d is also cartesian. $\square$

Lemma 4.7 enables us to restrict the componentwise lifting of a factorization system to $[\mathscr{A},\mathscr{D}]_{\mathrm{cartnt}}$. The following lemma enables us to restrict the factorization system $(\mathscr{E}',\mathscr{M}')$ defined at the beginning of this section.

Lemma 4.8. *Let $(\mathscr{E},\mathscr{M})$ be any factorization system on $[\mathscr{A},\mathscr{D}]$. If all natural transformations in $\mathscr{M}$ are cartesian, then $(\mathscr{E},\mathscr{M})$ restricts to a factorization system on $[\mathscr{A},\mathscr{D}]_{\mathrm{cartnt}}$.*

Proof. If $\tau = m \circ e$ and τ and m are cartesian, then e is cartesian by the pullback lemma. $\square$

Lemmata 4.7 and 4.8 provide two constructions of a factorization system on $[\mathscr{A},\mathscr{D}]_{\mathrm{cartnt}}$: the componentwise lifting of a factorization system on $\mathscr{D}$, and the factorization system $(\mathscr{E}',\mathscr{M}')$ of Lemma 4.2. We now show that the two coincide.

Proposition 4.9. *Let $(\mathscr{E},\mathscr{M})$ be a stable factorization system on a category $\mathscr{D}$ with pullbacks, and let $\mathscr{A}$ be a category with a terminal object. The factorization system $(\mathscr{E}',\mathscr{M}')$ on $[\mathscr{A},\mathscr{D}]$ from Lemma 4.2 and the componentwise lifting of $(\mathscr{E},\mathscr{M})$ to $[\mathscr{A},\mathscr{D}]$ both restrict to the same factorization system on $[\mathscr{A},\mathscr{D}]_{\mathrm{cartnt}}$.*

Proof. By stability of $(\mathscr{E},\mathscr{M})$, for each cartesian natural transformation e, having $e_1 \in \mathscr{E}$ is equivalent to having $e_X \in \mathscr{E}$ for all X. Hence when restricted to $[\mathscr{A},\mathscr{D}]_{\mathrm{cartnt}}$, the $\mathscr{E}'$-morphisms are exactly the $[\mathscr{A},\mathscr{D}]_{\mathrm{cartnt}}$-morphisms whose components are in $\mathscr{E}$, and similarly for $\mathscr{M}'$. $\square$

When $\mathscr{A}$ has pullbacks it follows, using Lemma 4.5, that the two factorization systems also restrict to the same factorization system on $[\mathscr{A},\mathscr{D}]_{\mathrm{cart}}$. When $\mathscr{A} = \mathscr{D}$, the latter forms a monoidal category with functor composition as tensor, and $\mathscr{E}'$ is closed under $(-)\cdot S$ for every cartesian endofunctor S. Hence we can construct canonical gradings of cartesian monads (monoids in $[\mathscr{D},\mathscr{D}]_{\mathrm{cart}}$) by Theorem 3.11. The grades of the canonical grading of a cartesian monad T are equivalently $\mathscr{M}$-subobjects of $T1$, by Lemma 4.2.

Example 4.10. Let us return to the factorization system $(\mathscr{E},\mathscr{M}) = (\mathbf{Surj},\mathbf{Inj})$ on $\mathscr{A} = \mathscr{D} = \mathbf{Set}$, which is stable. Let $(\mathscr{E}',\mathscr{M}')$ be the factorization system on $[\mathbf{Set},\mathbf{Set}]_{\mathrm{cart}}$ just discussed. Then $\mathscr{E}'$ is closed both under $(-)\cdot S$ and under $S\cdot(-)$ for any cartesian set functor S. Hence $\mathscr{M}'/T$ acquires a monoidal structure for any cartesian set monad T.

Let T be the list monad on **Set**, so TX is the set of lists over X, the unit is $\eta_X\, x = [x]$, and the multiplication is $\mu_X\,[xs_1,\ldots,xs_n] = xs_1 +\!\!+ \cdots +\!\!+ xs_n$, where $(+\!\!+)$ is concatenation of lists. This monad is cartesian. There is an isomorphism $T1 \cong \mathbb{N}$, so shapes are equivalently natural numbers (corresponding to the length of the list). Then $\mathscr{M}'$-subobjects of T are equivalently subsets $\Sigma \subseteq \mathbb{N}$. By the above, these form the canonical $\mathscr{M}'$-grading of T. The monoidal structure on these subsets is given by

$$\mathsf{J} = \{1\} \qquad \Sigma \boxdot \Sigma' = \{\textstyle\sum_{i=1}^{n} m_i \mid n \in \Sigma, m_1,\ldots,m_n \in \Sigma'\}$$

The graded monad $\mathsf{T}_{\mathscr{M}'}$ is given on objects by $\mathsf{T}_{\mathscr{M}'}\Sigma X = \{xs \mid |xs| \in \Sigma\}$, where $|xs|$ is the length of xs.

5 Algebraic operations

In models of computational effects, we usually do not just want a (strong) monad T; we also want to equip T with a collection of *algebraic operations* in the sense of Plotkin and Power [12]. The latter provide interpretations of the constructs that cause the effects. When modelling computations using a graded monad, we similarly want algebraic operations for the graded monad; such a notion of algebraic operation was introduced in [5]. In this section, we therefore investigate the problem of constructing algebraic operations for the graded monad $\mathsf{T}_{\mathscr{M}}$, given algebraic operations for the monad T.

Throughout this section, we assume a monoidal category $\mathbb{C} = (\mathscr{C},\mathsf{I},\otimes)$ that has finite products, for example, endofunctors on a category with finite products. When we write T^n below, we mean the product of n-many copies of T. We work only with normal (i.e., non-skew) monoidal categories in this section. The notion of algebraic operation for a graded monoid (e.g., a graded monad) that we use below works for monoidal categories, but the appropriate notion for skew monoidal categories would be more complicated. (It would use a list of grades e_i instead of a single grade e in the definition below.) Hence when we consider the canonical gradings below, we work under the assumption that they form a monoidal category (for example, when $\mathscr{E}$ is closed under $\otimes$ in both arguments).

The following definition generalizes the notion of algebraic operation for a monad to monoids.

Definition 5.1. Let $\mathsf{T} = (T,\eta,\mu)$ be a monoid in $\mathbb{C}$. An n-ary *algebraic operation* for T, where n is a natural number, is a morphism $\phi : T^n \to T$ such that

$$\begin{array}{ccccc} T^n \otimes T & \xrightarrow{\langle \pi_i \otimes T\rangle_i} & (T\otimes T)^n & \xrightarrow{\mu^n} & T^n \\ {\scriptstyle \phi\otimes T}\downarrow & & & & \downarrow{\scriptstyle\phi} \\ T\otimes T & & \xrightarrow{\qquad\mu\qquad} & & T \end{array}$$

Definition 5.2. Let $\mathsf{G} = (G,\eta,\mu) : \mathbb{G} \to \mathbb{C}$ be a $\mathbb{G}$-graded monoid in $\mathbb{C}$. A $(d_1,\ldots,d_n;d')$-ary algebraic operation for G, where $d_1,\ldots,d_n,d' \in \mathbb{G}$, is a natural transformation $\psi_e : \prod_i G(d_i \odot e) \Rightarrow G(d' \odot e)$ such that, for all $e,e' \in \mathscr{G}$,

$$\begin{array}{ccccccc} (\prod_i G(d_i\odot e))\otimes Ge' & \xrightarrow{\langle \pi_i\otimes Ge'\rangle_i} & \prod_i (G(d_i\odot e)\otimes Ge') & \xrightarrow{\prod_i \mu_{d_i\odot e,e'}} & \prod_i G((d_i\odot e)\odot e') & \xrightarrow{G\alpha} & \prod_i G(d_i\odot(e\odot e')) \\ {\scriptstyle\psi_e\otimes Ge'}\downarrow & & & & & & \downarrow{\scriptstyle\psi_{e\odot e'}} \\ G(d'\odot e)\otimes Ge' & & \xrightarrow{\qquad\mu_{d'\odot e,e'}\qquad} & & G((d'\odot e)\odot e') & \xrightarrow{G\alpha} & G(d'\odot(e\odot e')) \end{array}$$

Example 5.3. Let $\mathbb{C}$ be the cartesian monoidal category **Set**. Then an n-ary algebraic operation for a monoid T is a function $\phi : T^n \to T$ such that the multiplication of the monoid distributes over ϕ from the right. For example, if T is natural numbers with ordinary multiplication, then $\phi(x_1,\ldots,x_n) = x_1 + \cdots + x_n$ is an n-ary algebraic operation.

Definition 5.4. Let $(\mathbb{G}, \mathsf{G}, g)$ be an $\mathscr{M}$-grading of a monoid T, where $\mathscr{M}$ is a class of morphisms in $\mathscr{C}$. We say that a $(d_1, \ldots, d_n; d')$-ary algebraic operation ψ for G is a *grading* of an n-ary algebraic operation ϕ for T when the following diagram commutes for all $e \in \mathbb{G}$.

$$\begin{array}{ccc} \prod_i G(d_i \odot e) & \xrightarrow{\psi_e} & G(d' \odot e) \\ {\scriptstyle \prod_i g_{d_i \odot e}} \downarrow & & \downarrow {\scriptstyle g_{d' \odot e}} \\ T^n & \xrightarrow{\phi} & T \end{array}$$

Suppose that T is a monoid in $\mathbb{C}$, and that $(\mathscr{E}, \mathscr{M})$ is a factorization system on $\mathscr{C}$ such that $\mathscr{E}$ is closed under $(-) \otimes S$ for all $S \rightarrowtail T$. Then T has a canonical grading $\mathsf{T}_{\mathscr{M}} : \mathscr{M}/\mathsf{T} \to \mathbb{C}$ by Theorem 3.11. Suppose in addition that the skew monoidal category $\mathscr{M}/\mathsf{T}$ is actually monoidal (which is the case when $\mathscr{E}$ is closed also under $S \otimes (-)$ for all $S \rightarrowtail T$). We keep these assumptions without repeating them for the rest of this section.

Our goal in the rest of this section is to show that we can assign canonical grades to algebraic operations for T. To be more precise, let $\phi : T^n \to T$ is an n-ary algebraic operation for T, and let $R_1, \ldots, R_n$ be a list of grades ($\mathscr{M}$-subobjects of T). We show how to construct a grade R' and an algebraic operation

$$\psi : \textstyle\prod_i \mathsf{T}_{\mathscr{M}}(R_i \boxdot -) \to \mathsf{T}_{\mathscr{M}}(R' \boxdot -)$$

of arity $(R_1, \ldots, R_n; R')$ for $\mathsf{T}_{\mathscr{M}}$, such that ψ grades ϕ. The grade R' is in a sense canonical (see Theorem 5.6 below), and in fact every component of ψ is in $\mathscr{E}$.

To do this, we make the following two further assumptions about $\mathscr{E}$ for the rest of the section. Firstly, we assume that $\mathscr{E}$ contains the canonical morphisms $\langle \pi_i \otimes Y \rangle_i : (\prod_i X_i) \otimes Y \to \prod_i (X_i \otimes Y)$. This is the case in particular when $\otimes$ preserves finite products on the left (because $\mathscr{E}$ contains all isomorphisms); when $\otimes$ is composition of endofunctors this is automatically true. Secondly, we assume that $\mathscr{E}$ is closed under finite products, i.e. that $\prod_i e_i : \prod_i X_i \to \prod_i Y_i$ is in $\mathscr{E}$ whenever all of the morphisms $e_i : X_i \twoheadrightarrow Y_i$ are in $\mathscr{E}$. This is the case for all of the factorization systems we consider above.

The key lemma that enables us to construct ψ is the following, which characterizes algebraic operations for the canonical grading $\mathsf{T}_{\mathscr{M}}$ of T.

Lemma 5.5. *Let $\phi : T^n \to T$ be an n-ary algebraic operation for T, and let $R_1, \ldots, R_n, R'$ be $\mathscr{M}$-subobjects of T. There is a bijection between (1) morphisms $p : \prod_i R_i \to R'$ such that*

$$\begin{array}{ccc} \prod_i R_i & \xrightarrow{p} & R' \\ \downarrow & & \downarrow \\ T^n & \xrightarrow{\phi} & T \end{array}$$

and (2) $(R_1, \ldots, R_n; R')$-ary algebraic operations ψ for $\mathsf{T}_{\mathscr{M}}$ that grade ϕ.

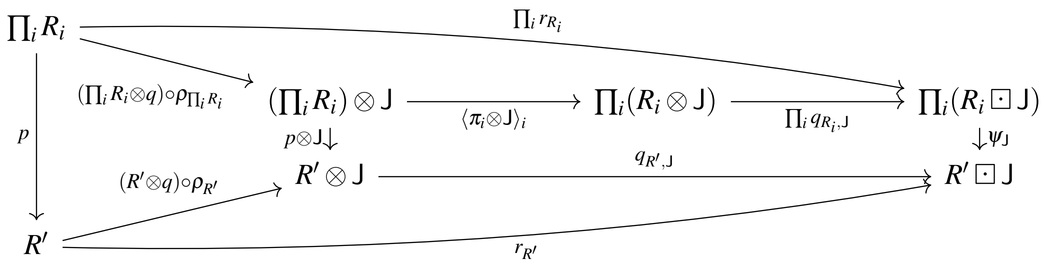
Proof. Given a morphism p as in (1), the following square commutes because ϕ is algebraic, and the square hence has a unique diagonal ψ_S. Further applications of orthogonality show that ψ is an algebraic operation. It is a grading of ϕ by definition.

$$\begin{array}{ccccc} (\prod_i R_i) \otimes S & \xrightarrow{\langle \pi_i \otimes S \rangle_i} & \prod_i (R_i \otimes S) & \xrightarrow{\prod_i q_{R_i,S}} & \prod_i (R_i \boxdot S) \\ {\scriptstyle p \otimes S} \downarrow & & & & \downarrow \\ R' \otimes S & & \psi_S & & T^n \\ {\scriptstyle q_{R',S}} \downarrow & & & & \downarrow {\scriptstyle \phi} \\ R' \boxdot S & & \longrightarrow & & T \end{array}$$

In the other direction, given ψ, we have a morphism p as follows; this p makes the diagram required for (1) commute because ψ is a grading of ϕ.

$$p : \prod_i R_i \xrightarrow{\prod_i r_{R_i}} \prod_i (R_i \boxdot \mathsf{J}) \xrightarrow{\psi_{\mathsf{J}}} R' \boxdot \mathsf{J} \xrightarrow{r_{R'}^{-1}} R'$$

From algebraicity of ψ it follows that this p makes the upper triangle of the above square commute and hence, by uniqueness of the diagonal, that ψ is the only grading of ϕ that induces this p. The construction of p from ψ is therefore injective. The following diagram chase shows that constructing a new p from the ψ constructed from a given p yields the same p, hence the constructions form a bijection.



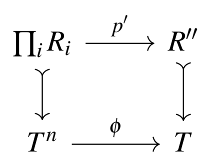
□

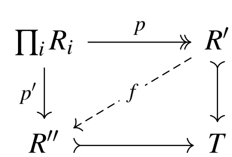
Now given an n-ary algebraic operation ϕ for T and a fixed tuple $R_1, \ldots, R_n$ of $\mathscr{M}$-subobjects of T, we construct the canonical R' by factorizing $\prod_i R_i \rightarrowtail T^n \xrightarrow{\phi} T$ as $\prod_i R_i \xrightarrow{p} R' \rightarrowtail T$. The preceding lemma then provides us with an $(R_1, \ldots, R_n; R')$-algebraic operation ψ for $\mathsf{T}_{\mathscr{M}}$.

Theorem 5.6. *Let $\phi : T^n \to T$ be an n-ary algebraic operation for T.*

1. *The construction above defines an $(R_1, \ldots, R_n; R')$-ary algebraic operation ψ for $\mathsf{T}_{\mathscr{M}}$, and ψ grades ϕ. Every component ψ_S is in $\mathscr{E}$.*
2. *For any $\mathscr{M}$-subobject $R'' \rightarrowtail T$ and $(R_1, \ldots, R_n; R'')$-ary algebraic operation ψ' for $\mathsf{T}_{\mathscr{M}}$, such that ψ' grades ϕ, there is a unique $f : R' \to R''$ in $\mathscr{M}/T$ such that $(f \boxdot S) \circ \psi_S = \psi'_S$ for all S.*

Proof. The first sentence of (1) is immediate from Lemma 5.5. Each ψ_S is in $\mathscr{E}$ because we have $\psi_S \circ e = e'$ for some $e, e' \in \mathscr{E}$ (this is the upper triangle in the definition of ψ_S, using the fact that p is in $\mathscr{E}$). This implies $\psi_S \in \mathscr{E}$ because $\mathscr{E}$-morphisms satisfy a two-out-of-three property. For (2), given ψ', we obtain from Lemma 5.5 a morphism $p' : \prod_i R_i \to R''$ making the diagram on the left below commute.





For a morphism $f : R' \to R''$ in $\mathscr{M}/T$, the condition that $(f \boxdot S) \circ \psi_S = \psi'_S$ for all S implies (using $S = \mathsf{J}$) that $f \circ p = p'$. The converse also holds, using orthogonality. Hence the conditions on the morphism f are equivalent to commutativity of the square on the right above. The outside of the square commutes and p is in $\mathscr{E}$, so there exists a unique f. □

Example 5.7. Consider the writer monad given by $T = M \times (-)$ from Example 3.12. Every $z \in M$ induces a unary algebraic operation $\phi_z : T \to T$, defined by $\phi_{z,X}(z', x) = (z \cdot z', x)$. When $\mathscr{M}$ is the class of componentwise injective natural transformations, the canonical $\mathscr{M}$-grading of T has subsets $\Sigma \subseteq M$

as grades, and $\mathsf{T}_{\mathscr{M}}\Sigma = \Sigma \times (-)$. Every input grade $P \subseteq M$ induces a canonical output grade $P'_z \subseteq M$ and algebraic operation $\psi_{z,\Sigma} : \mathsf{T}_{\mathscr{M}}(P \boxdot \Sigma) \Rightarrow \mathsf{T}_{\mathscr{M}}(P'_z \boxdot \Sigma)$, and these turn out to be:

$$P'_z = \{z \cdot z' \mid z' \in P\} \qquad \psi_{z,\Sigma,X}(z',x) = (z \cdot z', x)$$

Example 5.8. Let T be the list monad on **Set**. This has a binary algebraic operation $(+\!\!+) : T \times T \Rightarrow T$ that concatenates a pair of lists. As we explain in Example 4.10, subsets $\Sigma \subseteq \mathbb{N}$ provide a canonical grading of T. If P_1, P_2 are subsets of $\mathbb{N}$, then the grade we construct for the algebraic operation $(+\!\!+)$ as above is $P' = \{n_1 + n_2 \mid n_1 \in P_1, n_2 \in P_2\}$, and the algebraic operation for $\mathsf{T}_{\mathscr{M}}$ is the natural transformation $\mathsf{T}_{\mathscr{M}}(P_1 \boxdot -) \times \mathsf{T}_{\mathscr{M}}(P_2 \boxdot -) \Rightarrow \mathsf{T}_{\mathscr{M}}(P' \boxdot -)$ that maps (xs_1, xs_2) to $xs_1 +\!\!+ xs_2$.

6 Conclusion and future work

We have demonstrated that factorization systems provide a unifying framework for the grading of monads by subfunctors, in fact, monoids with subobjects. Skew monoidal categories turn out to be a more robust setting for this than monoidal categories, which means, among other things, that this framework will be directly applicable also to relative monads.

The abstract framework is pleasingly elegant, but for applications we would like obtain a stronger intuition for its reach. We intend to explore this first by working out the canonical gradings with (strong) subfunctors of further standard example (strong) monads from programming semantics, for the factorization systems considered in this paper and possibly others. Indeed, the examples may point to further factorization systems of interest. The outcomes of this exploration will hopefully lead to some new heuristics for the construction of graded monads for applications such as type-and-effect systems.

Programming semantics applications also suggest trying grading with subfunctors on (strong) lax monoidal functors ("applicative functors") and (strong) monads in **Prof** ("arrows"). Comonads can be graded with quotient functors.

Acknowledgements D.M. and T.U. were supported by the Icelandic Research Fund grants 196323-053 and 228684-051. This work started during T.U.'s visit to F.B. at LIPN as an invited professor in Sept. 2019, but was mostly stalled during the Covid pandemic.

References

[1] Tobias Fritz & Paolo Perrone (2018): *A Criterion for Kan Extensions of Lax Monoidal Functors*. arXiv eprint 1809.10481 [math.CT], doi:10.48550/arXiv.1809.10481.

[2] Soichiro Fujii, Shin-ya Katsumata & Paul-André Melliès (2016): *Towards a Formal Theory of Graded Monads*. In Bart Jacobs & Christof Löding, editors: *Proc. of 19th Int. Conf. on Foundations of Software Science and Computation Structures, FoSSaCS 2016*, *Lect. Notes in Comput. Sci.* 9634, Springer, Cham, pp. 513–530, doi:10.1007/978-3-662-49630-5_30.

[3] Ohad Kammar & Dylan McDermott (2018): *Factorisation Systems for Logical Relations and Monadic Lifting in Type-and-Effect System Semantics*. *Electron. Notes Theor. Comput. Sci.* 341, pp. 239–260, doi:10.1016/j.entcs.2018.11.012.

[4] Shin-ya Katsumata (2014): *Parametric Effect Monads and Semantics of Effect Systems*. In: *Proc. of 41st Ann. ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL '14*, ACM, New York, pp. 633–646, doi:10.1145/2535838.2535846.

[5] Shin-ya Katsumata, Dylan McDermott, Tarmo Uustalu & Nicolas Wu (2022): *Flexible Presentations of Graded Monads*. *Proc. ACM Program. Lang.* 6(ICFP), pp. 123:1–123:29, doi:10.1145/3547654.

[6] G. Max Kelly (1992): *On Clubs and Data-Type Constructors*. In Michael P. Fourman, Peter T. Johnstone & Andrew M. Pitts, editors: *Applications of Categories in Computer Science*, *London. Math. Soc. Lect. Note Series* 177, Cambridge Univ. Press, pp. 163–190, doi:10.1017/cbo9780511525902.010.

[7] Stephen Lack & Ross Street (2014): *Triangulations, Orientals, and Skew Monoidal Categories*. *Adv. Math.* 258, pp. 351–396, doi:10.1016/j.aim.2014.03.003.

[8] John M. Lucassen & David K. Gifford (1988): *Polymorphic Effect Systems*. In: *Proc. of 15th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL '88*, ACM, New York, pp. 47–57, doi:10.1145/73560.73564.

[9] Dylan McDermott, Maciej Piróg & Tarmo Uustalu (2020): *Degrading Lists*. In: *Proc. of 22nd Int. Symp. on Principles and Practice of Declarative Programming, PPDP '20*, ACM, New York, pp. 6:1–6:14, doi:10.1145/3414080.3414084.

[10] Paul-André Melliès (2012): *Parametric Monads and Enriched Adjunctions*. Manuscript. Available at `https://www.irif.fr/~mellies/tensorial-logic/8-parametric-monads-and-enriched-adjunctions.pdf`.

[11] Eugenio Moggi (1989): *Computational Lambda-Calculus and Monads*. In: *Proc. of 4th Ann. Symp. on Logic in Computer Science, LICS '89*, IEEE Press, Los Alamitos, CA, pp. 14–23, doi:10.1109/lics.1989.39155.

[12] Gordon Plotkin & John Power (2003): *Algebraic Operations and Generic Effects*. *Appl. Categ. Struct.* 11, pp. 69–94, doi:10.1023/a:1023064908962.

[13] Alexander Smirnov (2008): *Graded Monads and Rings of Polynomials*. *J. Math. Sci.* 151, pp. 3032–3051, doi:10.1007/s10958-008-9013-7.

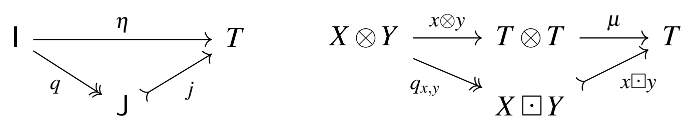
[14] Kornel Szlachányi (2012): *Skew-Monoidal Categories and Bialgebroids*. *Adv. Math.* 231(3–4), pp. 1694–1730, doi:10.1016/j.aim.2012.06.027.

A Proof of Theorem 3.9

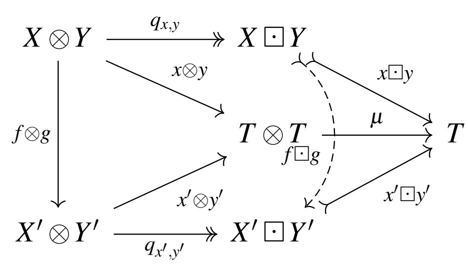
Given a monoidal category $(\mathscr{C}, \mathsf{I}, \otimes, \lambda, \rho, \alpha)$ with a monoid object (T, η, μ) and an orthogonal factorization system $(\mathscr{E}, \mathscr{M})$. We assume that $\mathscr{E}$ is closed under $(-) \otimes X$ for all $(X, x) \in \mathscr{M}/T$.

Our aim is to show $\mathscr{M}/T$ carries a left-skew monoidal category structure $((\mathsf{J}, j), \boxdot, \ell, r, a)$.

The unit (J, j) and tensor $(X \boxdot Y, x \boxdot y)$ of two objects (X, x), (Y, y) are defined as the factorizations shown in the diagrams below.

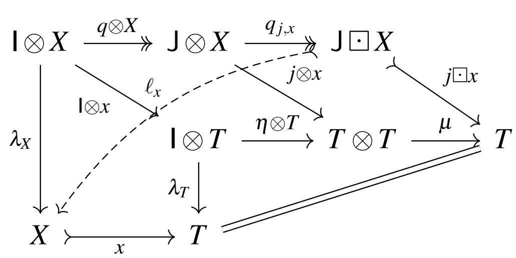


The functorial action of $\boxdot$ on two morphisms $f : (X, x) \to (X', x')$ and $g : (Y, y) \to (Y', y')$ is a morphism $f \boxdot g : (X \boxdot Y, x \boxdot y) \to (X' \boxdot Y', x' \boxdot y')$ defined as the diagonal fill-in of the commuting square below.

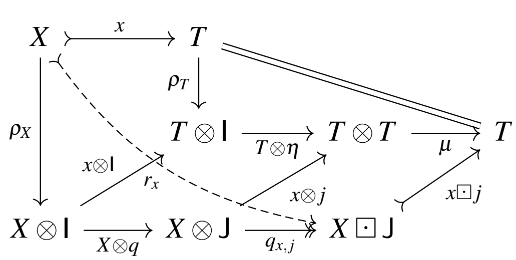


The left unitor ℓ and associator a are also defined as the diagonal fill-ins for suitable commuting squares. The right unitor is just a composition of morphisms.

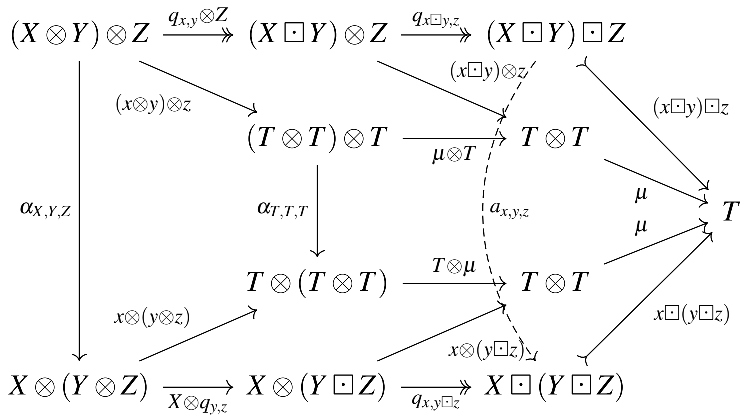
Definition of ℓ:



Definition of r:



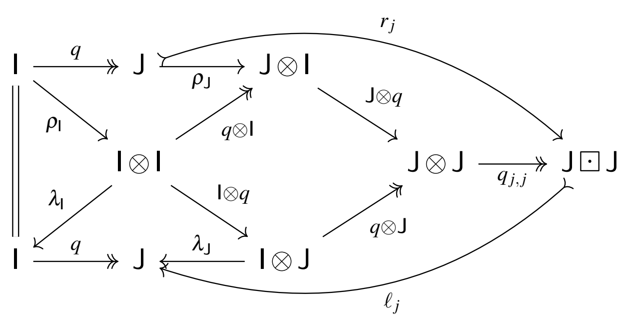
Definition of a:



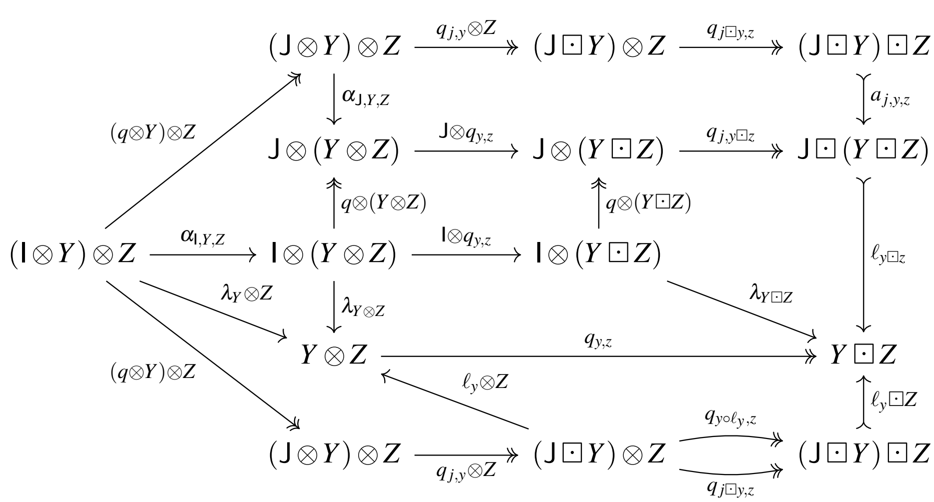
The proofs of functoriality of $\boxdot$ and naturality of ℓ, r and a are easy and omitted.

The equations (m1)–(m5) for ℓ, r, a are each proved from the respective equations of λ, ρ, α using the properties of $\boxdot$, ℓ, r, a arising from their construction (the two triangles that the fill-in breaks the square into). For each equation $lhs = rhs$, the two sides lhs and rhs are both shown to be the diagonal fill-in of a square of the form $s \circ e = s' \circ f$ where e is a suitable $\mathscr{E}$-morphism and s and s' are the common domain resp. codomain of lhs and rhs as morphisms in $\mathscr{M}/T$. Below are the diagram chases for the triangles $lhs \circ e = f = rhs \circ e$; the triangles $s' \circ lhs = s = s' \circ rhs$ are straightforward.

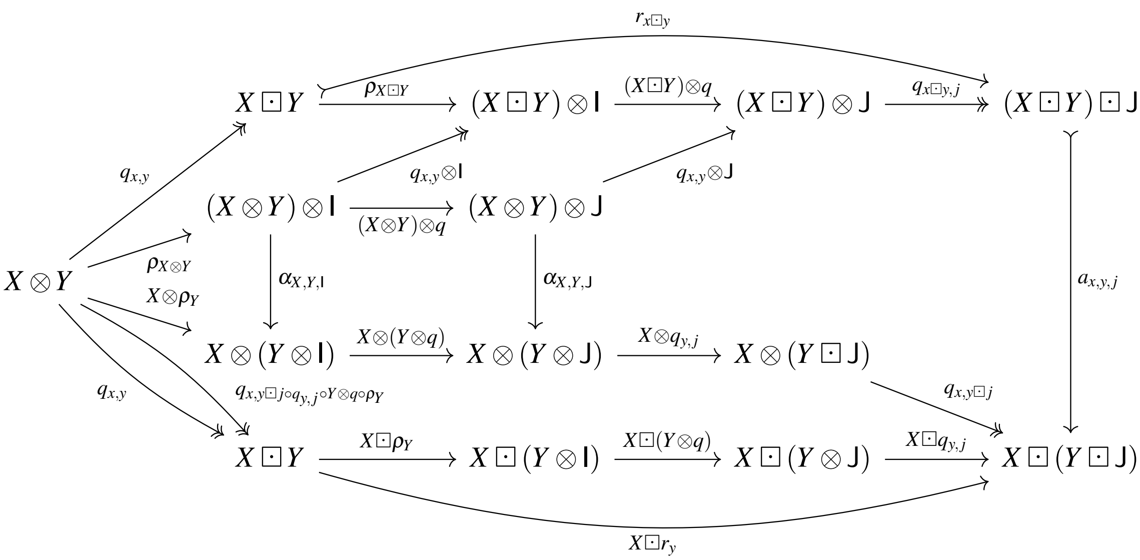
Proof of (m1):



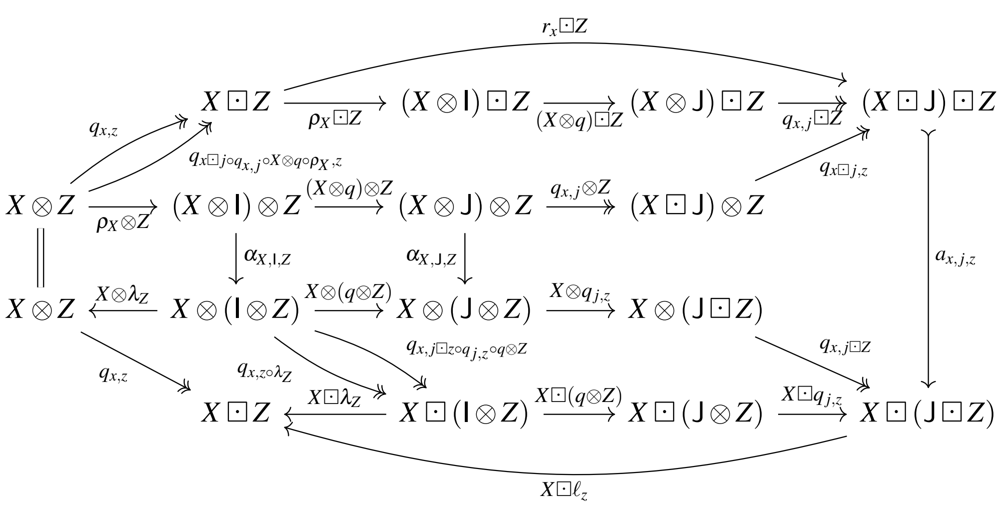
Proof of (m2):



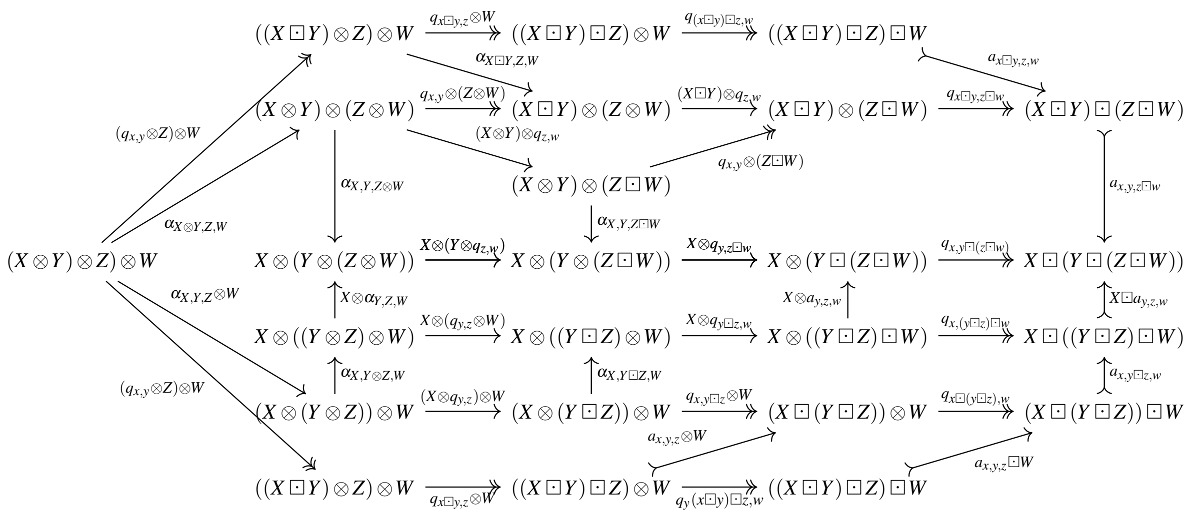
Proof of (m3):



Proof of (m4):



Proof of (m5):


$((X \boxdot Y) \otimes Z) \otimes W$
$((X \boxdot Y) \boxdot Z) \otimes W$
$((X \boxdot Y) \boxdot Z) \boxdot W$
$(X \otimes Y) \otimes (Z \otimes W)$
$(X \boxdot Y) \otimes (Z \otimes W)$
$(X \boxdot Y) \otimes (Z \boxdot W)$
$(X \boxdot Y) \boxdot (Z \boxdot W)$
$(X \otimes Y) \otimes (Z \boxdot W)$
$((X \otimes Y) \otimes Z) \otimes W$
$X \otimes (Y \otimes (Z \otimes W))$
$X \otimes (Y \otimes (Z \boxdot W))$
$X \otimes (Y \boxdot (Z \boxdot W))$
$X \boxdot (Y \boxdot (Z \boxdot W))$
$X \otimes ((Y \otimes Z) \otimes W)$
$X \otimes ((Y \boxdot Z) \otimes W)$
$X \otimes ((Y \boxdot Z) \boxdot W)$
$X \boxdot ((Y \boxdot Z) \boxdot W)$
$(X \otimes (Y \otimes Z)) \otimes W$
$(X \otimes (Y \boxdot Z)) \otimes W$
$(X \boxdot (Y \boxdot Z)) \otimes W$
$(X \boxdot (Y \boxdot Z)) \boxdot W$
$((X \boxdot Y) \otimes Z) \otimes W$
$((X \boxdot Y) \boxdot Z) \otimes W$
$((X \boxdot Y) \boxdot Z) \boxdot W$

Categorification of Negative Information using Enrichment

Andrea Censi Emilio Frazzoli Jonathan Lorand Gioele Zardini
Institute for Dynamic Systems and Control
Department of Mechanical and Process Engineering
ETH Zurich, Switzerland
{acensi,efrazzoli,jlorand,gzardini}@ethz.ch

In many engineering applications it is useful to reason about "negative information". For example, in planning problems, providing an optimal solution is the same as giving a feasible solution (the "positive" information) together with a proof of the fact that there cannot be feasible solutions better than the one given (the "negative" information). We model negative information by introducing the concept of "norphisms", as opposed to the positive information of morphisms. A "nategory" is a category that has "nom"-sets in addition to hom-sets, and specifies the interaction between norphisms and morphisms. In particular, we have composition rules of the form morphism + norphism → norphism. Norphisms do not compose by themselves; rather, they use morphisms as catalysts. After providing several applied examples, we connect nategories to enriched categtory theory. Specifically, we prove that categories enriched in de Paiva's dialectica categories **GC**, in the case $\mathbf{C} = \mathbf{Set}$ and equipped with a modified monoidal product, define nategories which satisfy additional regularity properties. This formalizes negative information categorically in a way that makes negative and positive morphisms equal citizens.

1 Introduction

1.1 Manipulation of negative information is important in applications of category theory

Our research group's background is in robotics and systems theory. In these fields, we have found that category theory can describe well many of the structures in our problems, but something is often missing: we find ourselves in the position of reasoning and writing algorithms that manipulate "negative information", but we do not know what is an appropriate categorical concept for it. We give some examples.

Robot motion planning can be formalized as the problem of finding a trajectory through an environment, respecting some constraint (e.g., avoiding obstacles). One can think of the robot configuration manifold $\mathbb{M}$ as a category where the objects are elements of the tangent bundle and the morphisms are the feasible paths according to the problem constraints. The output of planning problems has an intuitive representation in category theory, if the problem is feasible. A *path* planning algorithm is given two objects and must compute a *morphism* as a solution. A *motion* planning algorithm would compute a trajectory, which could be seen as a *functor* F from the manifold $[0,T]$ to M with $F(0) = A$ and $F(T) = B$. However, if the problem is infeasible–if no morphisms between two points can be found—if the algorithm must present a *certificate of infeasibility*–what is the equivalent concept in category theory?

In many cases, the problems are not binary (either a solution exists or not, either a proposition is true or not) but we care about the performance of solutions. For example, consider the case of the *weighted shortest path problem in dynamic programming*. The problem is to find a path through a graph that minimizes the sum of the weights of the edges on the path. In robotics, this can be used for planning problems, where the weights could represent the time, the distance, or the energy required by a robot to traverse an edge, and the nodes are either regions of space or, more generally, joint states of the world and environment. Proving that a path is optimal means producing the path *together with* a proof that there are no shorter paths. This is called a "certificate of optimality" and, like certificates of infeasibility, is negative information as it consists in negating the existence of a certain class of paths. Interestingly, one

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 22–40, doi:10.4204/EPTCS.380.2

can see algorithms such as Dijkstra's algorithm as constructing both positive and negative information at the same time, such that when a path is finally found, we are sure that there are no shorter ones [5].

In some cases, the negative information is a first-class citizen which is critical to the efficiency. Algorithms such as A* require the definition of *heuristic* functions, which is negative information: they provide a *lower bound* on the cost of a path between two points. And better heuristics make the algorithm faster. Again, we ask, what could be the categorical counterpart of heuristics?

In *co-design* [6, 2], a morphism $\mathbf{F} \to \mathbf{R}$ describes what functionality can be achieved with which resources. They are characterized as boolean profunctors, that is, monotone functions $\mathbf{F}^{\text{op}} \times \mathbf{R} \to \mathbf{Bool}$. The negative information would be a "nesign" problem that characterizes an impossibility. For example, if $\mathbf{F} = \mathbf{R} =$ Energy, we expect that in this universe we cannot find a realizable morphism d that satisfies $d(2J, 1J)$ (obtaining 2 Joules from 1 Joule). Can this be expressed as some sort of morphism? In which category does it live?

1.2 Our approach: "Categorification" of negative information

We briefly describe our thought process in finding a formalization for dealing with negative information.

One approach could have been to build structure on top of a category, at a higher level, using logic. We eschew this approach because of the belief that we should find a duality between positive and negative information that puts them "at the same level".

Our approach has been one in the spirit of "categorification": representing the negative information with a concrete structure for which to find axioms and inference rules.

An early influence in our thinking was the paper of Shulman about "proofs and refutations" [8]. What follows is a simplified explanation of one of the concepts of the paper. Consider a category where objects are propositions and morphisms $X \to Y$ are propositions $X \Rightarrow Y$ (with the particular case of $X \simeq (\top \to X)$). We can then consider the type $P(X \to Y)$ of *proofs* and the type $R(X \to Y)$ of *refutations*, which correspond to *positive* and *negative* information. According to intuitionist logic, $P(X \to Y) = (P(X) \to P(Y)) \times (R(Y) \to R(X))$: a proof of $X \Rightarrow Y$ is a way to convert a proof of X into a proof of Y together with a way to convert a refutation of Y into a refutation of X.

In that paper, proofs and refutations, positive and negative information, are treated *at the same level* but not symmetrically—proof and refutations have different semantics, and P and R map products and coproducts ($\vee$, $\wedge$) to different linear logic operators. This led to the idea that negative information should be at the same level of positive information: if positive information is represented by morphisms, then also the negative information should be described as "negative arrows" between objects, which we called *norphisms* (for negative morphisms).

We also realized that the positive/negative information duality we are looking for is richer than the structure of proofs/refutations in logic. In (classical/intuitionistic) logic, one expects the existence of either a proof of a proposition A, a refutation of A, or neither, but not both. Instead, in our formalization, norphisms are a more general notion, which can coexist with morphisms and give complementary information, as in the planning examples in the introduction.

An initial idea was to consider for each category a "twin" category, whose morphisms would be the norphisms we were looking for to represent the negative information; however, this idea failed. In the course of the paper, it will be clear that positive/negative information cannot be decoupled, because negative information cannot be composed independently of positive information. In the end, we unite them by viewing them as part of a single enrichement structure.

1.3 Plan of the paper

This paper follows an inductive exposition and is divided in two parts.

In the **first part** we provide the **motivation and several examples of representing negative information with "norphism" structure**. In Section 2 we consider the case of a thin category. In this simple setting we can already see that norphisms compose differently than morphisms, and that we need two composition rules for them. In Section 3 we state our main definition, that of a "nategory", and in Section 4 we show some canonical ways to build a nategory out of a category. In Sections 5 and 6 we discuss two examples, **Berg** and **DP**, which have norphism structures in which norphisms and morphisms are not mutually exclusive.

In the **second part** our goal is to provide **an elegant way to think of norphisms and their composition by using enriched category theory**. By doing so, we show that the additional structure of norphisms and their composition, rules which might initially appear "funky", is not an arbitrary structure, but rather it is as "natural" as the positive information of morphisms. In Section 7 we introduce the dialectica category **GSet** [3, 1] and define a monoidal product for it which is slightly different than the ones usually used as linear logic connectives. Then, in Section 8, we prove that **GSet**-enriched categories encode nategories which satisfy some additional compatibilities between morphisms and norphisms. These additional compatibilities are not satisfied in certain examples of interest to us, therefore we have refrained from including them directly in our definition of nategory.

2 Building intuition: the case of thin categories

To build an intuition about norphisms, we look at the case of "thin" categories, in which each hom-set contains at most one morphism. Thin categories are essentially pre-orders. To aid the interpretation, one can think of a pre-order as defining a reachability relation, in which a morphism $X \to Y$ represents "I can reach Y from X". Or, we can think of morphisms as (proof-irrelevant) implications: $X \to Y$ represents "I can prove Y from X". In a thin category, negative information is limited to indicate the refutation of positive information. Therefore, a norphism $n\colon X \dashrightarrow Y$ is equivalent to "There are no morphisms from X to Y". Particularly, this means "I cannot reach Y from X" or "I cannot prove Y from X".

We will later see that, in general, norphisms need not necessarily be mutually exclusive with morphisms. Still, this example is sufficient to get us started in appreciating how morphisms and norphisms compose differently. The composition rule for morphisms reads:

$$\frac{f\colon X \to Y \quad g\colon Y \to Z}{(f \mathbin{⨟} g)\colon X \to Z}\,. \tag{1}$$

Mimicking this, one could start with two norphisms $n\colon X \dashrightarrow Y$ and $m\colon Y \dashrightarrow Z$ and expect to be able to say something about a norphism $X \dashrightarrow Z$, with a composition rule of the form:

$$\frac{n\colon X \dashrightarrow Y \quad m\colon Y \dashrightarrow Z}{???\colon X \dashrightarrow Z}\,. \tag{2}$$

However, norphisms do not compose this way. In fact, one can derive the following rule:

$$\frac{o\colon X \dashrightarrow Z \qquad Y\colon \mathrm{Ob}_{\mathbf{C}}}{(n\colon X \dashrightarrow Y) \vee (m\colon Y \dashrightarrow Z)}\,. \tag{3}$$

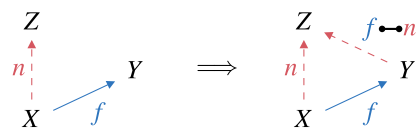
This rule is "the dual" of (1) in the same sense as these two axioms are dual:

$$\frac{\top}{X \to X}\,, \qquad \frac{X \dashrightarrow X}{\bot}\,, \tag{4}$$

that is, in the sense of flipping vertically and negating the propositions.

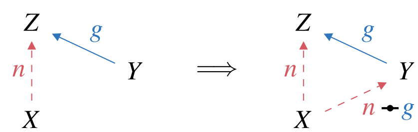
We read (3) as saying that if there is no morphism $X \to Z$, it is because, for every possible intermediate Y, there cannot be a morphism $X \to Y$ or $Y \to Z$. Note that composition goes in the "opposite" direction meaning that from one norphism, we get some information about the existence of one or two in a pair. The composition (3) is not constructive: from the "$\vee$", we do not know which side we can create. Indeed, this composition highlights the asymmetry between morphisms and norphisms: morphisms compose constructively by themselves (i.e., without taking into account norphisms); norphisms, instead, do not "compose", but rather "decompose" by themselves. To construct norphisms, we need to start from a norphism *and* a morphism that acts as a "catalyst".

When interpreting a thin category as a graph, if there is a norphism $n\colon X \dashrightarrow Y$, it means that for any Y, the path $X \to Y \to Z$ must be interrupted in one part or the other, because otherwise we would have a contradiction. Indeed, if we know that morphisms $f\colon X \to Y$ and $g\colon Y \to Z$ exist, then their composition $f \mathbin{⨾} g\colon X \to Z$ must exist, and therefore no norphism $n\colon X \dashrightarrow Z$ can exist. This observation can be turned around in a constructive way. Starting from a morphism $f\colon X \to Y$ and a norphism $n\colon X \dashrightarrow Z$ (i.e., morphisms and norphisms with the same source), we can infer a norphism $f \multimapdotboth n\colon Y \dashrightarrow Z$ (i.e., there cannot be a morphism $Y \to Z$):



$$\frac{Y \xleftarrow{f} X \overset{n}{\dashrightarrow} Z}{Y \overset{f \multimapdotboth n}{\dashrightarrow} Z}. \tag{5}$$

Symmetrically, starting from a morphism $g\colon Y \to Z$ and a norphism $n\colon X \dashrightarrow Z$ (i.e., morphisms and norphisms with the same target), we can infer a norphism $n \multimapdotinv f\colon X \dashrightarrow Y$:



$$\frac{X \overset{n}{\dashrightarrow} Z \xleftarrow{g} Y}{X \overset{n \multimapdotinv g}{\dashrightarrow} Y}. \tag{6}$$

Note that the new norphism is pointing in the "same direction" as the starting one, meaning that either source or target are preserved.

3 Describing negative information: *nategories*

In this section we make the notion of norphisms more precise, by defining the additional structure which a category must have in order to encode negative information.

Definition 1 (Nategory)**.** A locally small *nategory* $\mathbf{C}$ is a locally small category with the following additional structure. For each pair of objects $X, Y \in \mathrm{Ob}_{\mathbf{C}}$, in addition to the set of morphisms $\mathrm{Hom}_{\mathbf{C}}(X;Y)$, we also specify:

- A set of norphisms $\mathrm{Nom}_{\mathbf{C}}(X;Y)$.
- An *incompatibility relation*, which we write as a binary function

$$i_{XY}\colon \mathrm{Nom}_{\mathbf{C}}(X;Y) \times \mathrm{Hom}_{\mathbf{C}}(X;Y) \to \{\bot, \top\}. \tag{7}$$

For all triples X, Y, Z, in addition to the morphism composition function

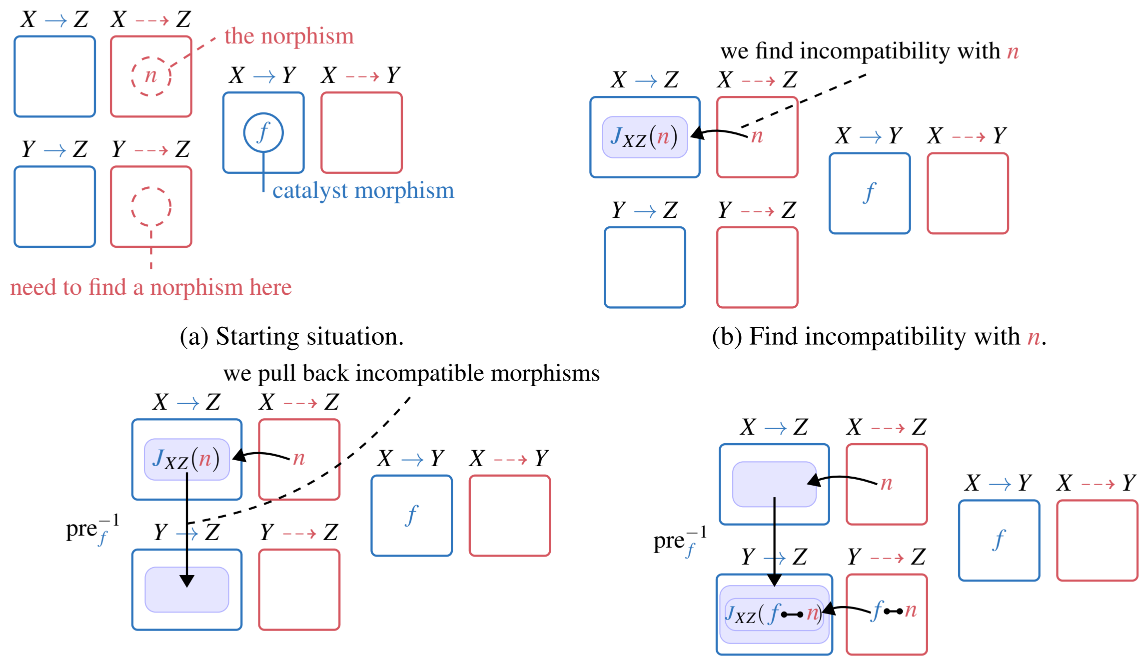
$$⨾_{XYZ}\colon \mathrm{Hom}_{\mathbf{C}}(X;Y) \times \mathrm{Hom}_{\mathbf{C}}(Y;Z) \to \mathrm{Hom}_{\mathbf{C}}(X;Z), \tag{8}$$

we require the existence of two norphism composition functions

$$\begin{aligned} \multimapdotboth_{XYZ}\colon \mathrm{Hom}_{\mathbf{C}}(X;Y) \times \mathrm{Nom}_{\mathbf{C}}(X;Z) \to \mathrm{Nom}_{\mathbf{C}}(Y;Z), \\ \multimapdotinv_{XYZ}\colon \mathrm{Nom}_{\mathbf{C}}(X;Z) \times \mathrm{Hom}_{\mathbf{C}}(Y;Z) \to \mathrm{Nom}_{\mathbf{C}}(X;Y), \end{aligned} \tag{9}$$

and we ask that they satisfy two "equivariance" conditions:

$$i_{YZ}(f \multimapdotboth n, g) \Rightarrow i_{XZ}(n, f ⨾ g), \tag{equiv-1}$$
$$i_{XY}(n \multimapdotinv g, f) \Rightarrow i_{XZ}(n, f ⨾ g). \tag{equiv-2}$$



(a) Starting situation.

(b) Find incompatibility with n.

(c) Pulling back incompatible morphisms.

(d) $J_{YZ}(f \mathbin{\bullet\!\!-\!\!\bullet} n) \subseteq \mathrm{pre}_f^{-1}(J_{XZ}(n))$.

Figure 1

We write $n : X \dashrightarrow Y$ to say that $n \in \mathrm{Nom}_{\mathbf{C}}(X;Y)$.

Definition 2 (Exact nategory)**.** If the two conditions (equiv-1) and (equiv-2) are satisfied with "$\Leftrightarrow$" instead of just "$\Rightarrow$", we say that the nategory is *exact*.

Condition (equiv-1) says that the norphism $f \mathbin{\bullet\!\!-\!\!\bullet} n$ can exclude the morphism g only if $f \fatsemi g$ is excluded by n. The idea is that such a g should not be excluded for any "additional reasons", but only on the grounds that $f \fatsemi g$ is excluded by n.

We draw some figures to develop further intuition (Fig. 1). Let J_{XY} denote the function which maps a norphism to the set of morphisms with which it is incompatible:

$$\begin{aligned} J_{XY} : \mathrm{Nom}_{\mathbf{C}}(X;Y) &\to \mathsf{Pow}(\mathrm{Hom}_{\mathbf{C}}(X;Y)), \\ n &\mapsto \{f \in \mathrm{Hom}_{\mathbf{C}}(X;Y) : i_{XY}(n, f)\}. \end{aligned} \tag{10}$$

We start in Fig. 1a with a norphism $n : X \dashrightarrow Z$ and a morphism $f : X \to Y$. In Fig. 1b we apply J_{XZ} to find the set of incompatible morphisms $J_{XZ}(n)$. In Fig. 1c we use the precomposition map

$$\begin{aligned} \mathrm{pre}_f : \mathrm{Hom}_{\mathbf{C}}(Y;Z) &\to \mathrm{Hom}_{\mathbf{C}}(X;Z), \\ g &\mapsto f \fatsemi g, \end{aligned} \tag{11}$$

to obtain the set of morphisms

$$\mathrm{pre}_f^{-1}(J_{XZ}(n)). \tag{12}$$

These are to be prohibited because when pre-composed with f they give a morphism that is forbidden by n. Now, in principle, it could be that our norphism inference is so powerful that $f \mathbin{\bullet\!\!-\!\!\bullet} n$ manages to exclude all of these:

$$J_{YZ}(f \mathbin{\bullet\!\!-\!\!\bullet} n) = \mathrm{pre}_f^{-1}(J_{XZ}(n)). \tag{13}$$

In general, we are happy with the composition operation if it excludes part of those (but not more):

$$J_{YZ}(f \mathbin{\bullet\!\!-\!\!\bullet} n) \subseteq \mathrm{pre}_f^{-1}(J_{XZ}(n)). \tag{14}$$

It can readily be checked that (14) is equivalent to (equiv-1). Similarly, (equiv-2) is equivalent to requiring

$$J_{YZ}(n \leftarrow\!\!\bullet\, g) \subseteq \mathrm{post}_g^{-1}(J_{XZ}(n)), \tag{15}$$

where post_g is the map "post-composition with g".

4 Canonical nategory constructions

Here are three canonical constructions that allow us to get a nategory out of a category in a more or less straightforward way:

1. Setting the norphism sets to be empty (Example 3);
2. Setting the norphism sets to be singletons that negate the entire respective hom-sets (Example 4);
3. Setting the norphism sets to be the powerset of the respective hom-sets (Example 5).

Example 3 (A nategory with no norphisms)**.** For any category $\mathbf{C}$, let

$$\mathrm{Nom}_{\mathbf{C}}(X;Y) := \emptyset. \tag{16}$$

For all pairs X, Y the function i_{XY} is uniquely defined as it has an empty domain. The functions $\bullet\!\!\to$, $\leftarrow\!\!\bullet$ also have empty domains. The conditions (equiv-1) and (equiv-2) are trivially verified. A nategory with no norphisms is just a category.

Example 4 (Singleton norphism sets negating all morphisms)**.** In this construction, we turn a category into a nategory by making the choice that a norphism is a witness for the fact that the corresponding hom-set is empty. For any category $\mathbf{C}$, let

$$\mathrm{Nom}_{\mathbf{C}}(X;Y) := \{\bullet\}, \tag{17}$$

and for any pair X, Y and any $f \colon X \to Y$ let

$$i_{XY}(\bullet, f) = \top. \tag{18}$$

In this case, the element $\bullet$ is a witness for "$\mathrm{Hom}_{\mathbf{C}}(X;Y)$ is empty". Next, we need to define the two maps:

$$\bullet\!\!\to \colon \mathrm{Hom}_{\mathbf{C}}(X;Y) \times \mathrm{Nom}_{\mathbf{C}}(X;Z) \to \mathrm{Nom}_{\mathbf{C}}(Y;Z), \tag{19}$$

$$\leftarrow\!\!\bullet \colon \mathrm{Nom}_{\mathbf{C}}(X;Z) \times \mathrm{Hom}_{\mathbf{C}}(Y;Z) \to \mathrm{Nom}_{\mathbf{C}}(X;Y). \tag{20}$$

The choice is forced, as there is only one norphism in the codomains. We obtain:

$$f \bullet\!\!\to \bullet = \bullet, \tag{21}$$

$$\bullet \leftarrow\!\!\bullet\, g = \bullet. \tag{22}$$

The conditions (equiv-1) and (equiv-2) are easily verified because i_{XY} always evaluates to $\top$.

Example 5 (Norphism sets are subsets of hom-sets)**.** For any category $\mathbf{C}$, let

$$\mathrm{Nom}_{\mathbf{C}}(X;Y) = \mathsf{Pow}(\mathrm{Hom}_{\mathbf{C}}(X;Y)). \tag{23}$$

Set the incompatibility relation as

$$i_{XY}(n, f) = f \in n. \tag{24}$$

Define the composition operations as

$$f \bullet\!\!\to n = \mathrm{pre}_f^{-1}(n), \tag{25}$$

$$n \leftarrow\!\!\bullet\, g = \mathrm{post}_g^{-1}(n), \tag{26}$$

where pre_f and post_g are the pre- and post-composition maps

$$\begin{aligned} \mathrm{pre}_f \colon \mathrm{Hom}_{\mathbf{C}}(Y;Z) &\to \mathrm{Hom}_{\mathbf{C}}(X;Z), \\ g &\mapsto f \mathbin{⨟} g, \end{aligned} \tag{27}$$

$$\begin{aligned} \mathrm{post}_g \colon \mathrm{Hom}_{\mathbf{C}}(X;Y) &\to \mathrm{Hom}_{\mathbf{C}}(X;Z), \\ f &\mapsto f \mathbin{⨟} g. \end{aligned} \tag{28}$$

Let's check the condition (equiv-1): $i_{YZ}(f \multimap n, g) \Rightarrow i_{XZ}(n, f \mathbin{⨟} g)$. Using our definitions, we get

$$g \in f \multimap n \quad \Rightarrow \quad f \mathbin{⨟} g \in n. \tag{29}$$

Expanding the left-hand side we find

$$g \in \mathrm{pre}_f^{-1}(n) \quad \Rightarrow \quad f \mathbin{⨟} g \in n. \tag{30}$$

Another expansion shows that both sides are the same:

$$f \mathbin{⨟} g \in n \quad \Rightarrow \quad f \mathbin{⨟} g \in n. \tag{31}$$

Checking condition (equiv-2) is analogous. Note that this nategory is exact.

Finally, we provide an example of a nategory that we will use later as a counter-example.

Example 6 (Very weak composition operations)**.** For any category **C**, as in the previous example, use subsets of morphisms as the norphisms

$$\mathrm{Nom}_{\mathbf{C}}(X;Y) = \mathsf{Pow}(\mathrm{Hom}_{\mathbf{C}}(X;Y)), \tag{32}$$

and set the incompatibility relation as

$$i_{XY}(n, f) = f \in n. \tag{33}$$

However, define the composition operations as

$$f \multimap n = \emptyset, \tag{34}$$
$$n \multimapinv g = \emptyset. \tag{35}$$

The equivariance conditions are still satisfied. For example condition (equiv-1),

$$i_{YZ}(f \multimap n, g) \Rightarrow i_{XZ}(n, f \mathbin{⨟} g), \tag{36}$$

becomes

$$g \in \emptyset \;\Rightarrow\; f \mathbin{⨟} g \in \emptyset, \tag{37}$$

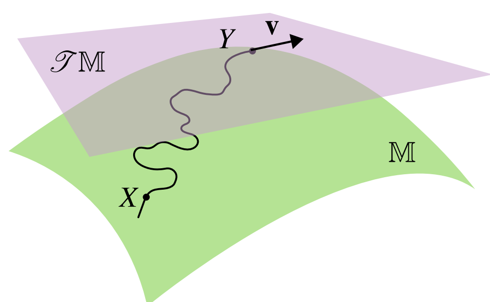
which is vacuously satisfied, because the premise is always false.

5 Example: hiking on the Swiss mountains

In this section we present an example of planning, giving a more concrete description of the path planning problems mentioned in the introduction. We describe **Berg**, a category whose morphisms are hiking paths of various difficulty on a mountain. We then consider the problem of finding paths of minimum length.

Definition 7 (**Berg**)**.** Let $h\colon \mathbb{R}^2 \to \mathbb{R}$ be a C^1 function, describing the elevation of a mountain. The set with elements $\langle a, b, h(a,b)\rangle$ is a manifold $\mathbb{M}$ that is embedded in $\mathbb{R}^3$. Let $\sigma = [\sigma_{\mathrm{L}}, \sigma_{\mathrm{U}}] \subset \mathbb{R}$ be a closed interval of real numbers. The category $\mathbf{Berg}_{h,\sigma}$ is specified as follows:

1. An object X is a pair $\langle \mathbf{p}, \mathbf{v} \rangle \in \mathscr{T}\mathbb{M}$, where $\mathbf{p} = \langle \mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z \rangle$ is the position, $\mathbf{v}$ is the velocity, and $\mathscr{T}\mathbb{M}$ is the tangent bundle of the manifold.
2. Morphisms are C^1 paths $f\colon [0, \tau] \to \mathbb{M}$ on the manifold satisfying evident boundary conditions (here $\tau \in \mathbb{R}$ may vary). We also define, formally by decree, that for each $\langle \mathbf{p}, \mathbf{v} \rangle$ there is a *trivial path* "$[0,0] \to \mathbb{M}$" which is C^1, has trace $\mathbf{p}$, and velocity $\mathbf{v}$.
   At each point $\mathbf{p} = f(t)$ of a path we define the *steepness* via the formula
   $$s(\langle \mathbf{p}, \mathbf{v} \rangle) := \mathbf{v}_z / \sqrt{\mathbf{v}_x^2 + \mathbf{v}_y^2}, \tag{38}$$
   where $\mathbf{v} = \frac{d}{dt} f(t)$. We choose as morphisms only the paths that have the steepness values contained in the interval σ:
   $$\mathrm{Hom}_{\mathbf{Berg}_{h,\sigma}}(X;Y) = \{f \text{ is a } C^1 \text{ path from } X \text{ to } Y \text{ and } s(f) \subseteq \sigma\}, \tag{39}$$
3. Morphism composition is given by concatenation of paths.
4. Identity morphisms are given by trivial paths.



The steepness interval σ allows considering different categories on the same mountain, with possible hikes varying in difficulty, measured via minimum/maximum steepness. For example, a good hiker can handle $\sigma = [-0.57, 0.57]$ (positive/negative 30° slope). If $\sigma = [-0.57, 0]$, we are only allowed to climb down. If $\sigma = [0, 0]$, we can only walk along isoclines.

Interpretation of norphisms in Berg What might a norphism be in this case?

One possibility is to let a norphism $n\colon X \dashrightarrow Y$ mean "there exists no path from X to Y". This is a simple choice that is similar to Example 4 and that makes morphisms and norphisms mutually exclusive.

We can obtain a more useful theory by letting norphisms carry information that is *complementary* to morphisms by interpreting them as *lower bounds* on distances. To see how this can work, let the set of norphisms be the real numbers completed by positive infinity:

$$\mathrm{Nom}_{\mathbf{Berg}}(X;Y) := \mathbb{R}_{\geq 0} \cup \{+\infty\}. \tag{40}$$

Let length(f) be the length of the path (according to the manifold metric). Then we interpret a norphism $n\colon X \dashrightarrow Y$ as a witness of "for all paths $f\colon X \to Y$, we have $\mathrm{length}(f) \geq n$". The case $n = \infty$ negates any path from X to Y. The incompatibility relation i_{XY} can be written as follows:

$$i_{XY}(n, f) = \mathrm{length}(f) < n. \tag{41}$$

To say that a path f is optimal means saying that f is feasible *and* that length(f) is a norphism:

$$\frac{f\colon X \to Y \quad \mathrm{length}(f)\colon X \dashrightarrow Y}{f \text{ is optimal}}. \tag{42}$$

Figure 2: Composition of morphisms and norphisms in the case of paths and lengths.

Composition rules for norphisms Next, we define the following two composition rules

$$\begin{aligned} f \mathbin{\bullet\!-\!\bullet} n &= \max\{n - \mathrm{length}(f), 0\}, \\ n \mathbin{-\!\bullet} g &= \max\{n - \mathrm{length}(g), 0\}, \end{aligned} \tag{43}$$

which are the equivalent of (5) and (6). See Fig. 2. Our reasoning is as follows. If for example f is a path from X to Y, and we know that going from X to Z has a distance of at least n, then any path from Y to Z must be at least $n - \mathrm{length}(f)$ long. In this case,

$$J_{XZ}(f \mathbin{\bullet\!-\!\bullet} n) = \{g \colon \mathrm{length}(g) < \max\{n - \mathrm{length}(f), 0\}\}. \tag{44}$$

If $n < \mathrm{length}(f)$, then $J_{XZ}(f \mathbin{\bullet\!-\!\bullet} n)$ is empty, which differs from

$$\mathrm{pre}_f^{-1}(J_{XY}(n)) = \{g \colon \mathrm{length}(g) + \mathrm{length}(f) < n\}. \tag{45}$$

The nategory is not exact. However, since

$$\{g \colon \mathrm{length}(g) < \max\{n - \mathrm{length}(f), 0\}\} \subseteq \{g \colon \mathrm{length}(g) + \mathrm{length}(f) < n\}, \tag{46}$$

the nategory satisfies (equiv-1). The check for (equiv-2) is analogous.

Example 8. As a variant of the above, if we set

$$\mathrm{Nom}_{\mathbf{Berg}}(X;Y) \coloneqq \mathbb{R} \cup \{+\infty\}, \tag{47}$$

and define the composition operations as

$$\begin{aligned} f \mathbin{\bullet\!-\!\bullet} n &= n - \mathrm{length}(f), \\ n \mathbin{-\!\bullet} g &= n - \mathrm{length}(g), \end{aligned} \tag{48}$$

then the nategory is exact. Indeed, for this case one has

$$\begin{aligned} J_{XZ}(f \mathbin{\bullet\!-\!\bullet} n) &= \{g \colon \mathrm{length}(g) < n - \mathrm{length}(f)\}, \\ &= \{g \colon \mathrm{length}(g) + \mathrm{length}(f) < n\} \\ &= \mathrm{pre}_f^{-1}(J_{XY}(n)). \end{aligned} \tag{49}$$

Example 9. We may also think of a variation in which the norphisms are integers:

$$\mathrm{Nom}_{\mathbf{Berg}}(X;Y) \coloneqq \mathbb{Z} \cup \{+\infty\}. \tag{50}$$

In this case we are limited to express constraints of the type

$$\mathrm{length}(f) \geq 0,\ \mathrm{length}(f) \geq 1,\ \mathrm{length}(f) \geq 2, \ldots \tag{51}$$

We then define the composition rules as

$$\begin{aligned} f \mathbin{\bullet\!-\!\bullet} n &= \mathsf{floor}(n - \mathrm{length}(f)), \\ n \mathbin{-\!\bullet} g &= \mathsf{floor}(n - \mathrm{length}(g)). \end{aligned} \tag{52}$$

In this case, (equiv-1) is satisfied, however our nategory is not exact, because in general

$$\begin{aligned} J_{XZ}(f \mathbin{\bullet\!-\!\bullet} n) &= \{g \colon \mathrm{length}(g) < \mathsf{floor}(n - \mathrm{length}(f))\} \\ &\subsetneq \{g \colon \mathrm{length}(g) < n - \mathrm{length}(f)\} \\ &= \mathrm{pre}_f^{-1}(J_{XY}(n)), \end{aligned} \tag{53}$$

since $\mathsf{floor}(n - \mathrm{length}(f)) + \mathrm{length}(f) \leq n$. An analogous reasoning applies to (equiv-2). We note that using $\mathsf{round}(-)$ or $\mathsf{ceil}(-)$ in (52) would violate (equiv-1) and (equiv-2).

Norphism schemas So far, we have not discussed heuristics for actually choosing a set Nom for each pair of objects in **Berg**. Here are some different ways.

1. **Non-negativity of lenths**. Since path lengths cannot be negative, for all pair of objects X, Y we can say that we have a norphism
$$0\colon X \dashrightarrow Y. \tag{54}$$
If these are our only norphisms, we are providing no new information about paths.
2. **Bound based on distance in $\mathbb{R}^3$**. Any path along the mountain cannot be shorter than the distance of a straight line ("as the crow flies"). Therefore, for two objects $\langle \mathbf{p}^1, \mathbf{v}^1\rangle$, $\langle \mathbf{p}^2, \mathbf{v}^2\rangle$, we might choose the distance $\|\mathbf{p}^1 - \mathbf{p}^2\|$ in $\mathbb{R}^3$
$$\|\mathbf{p}^1 - \mathbf{p}^2\|\colon \langle \mathbf{p}^1, \mathbf{v}^1\rangle \dashrightarrow \langle \mathbf{p}^2, \mathbf{v}^2\rangle. \tag{55}$$
as a norphism.
3. **Bound based on geodesic distance**. More accurate bounds are given by taking geodesic distance as our norphisms. This is defined using the metric $d_{\mathbb{M}}$ of the manifold:
$$d_{\mathbb{M}}(\mathbf{p}^1, \mathbf{p}^2)\colon \langle \mathbf{p}^1, \mathbf{v}^1\rangle \dashrightarrow \langle \mathbf{p}^2, \mathbf{v}^2\rangle. \tag{56}$$
4. **Bound based on steepness interval**. A different kind of norphism is to encode steepness information, and relate it to the steepness of paths, instead of their length. Given two objects $\langle \mathbf{p}^1, \mathbf{v}^1\rangle$, $\langle \mathbf{p}^2, \mathbf{v}^2\rangle$, we can use one of the following bounds
$$\frac{\mathbf{p}^1_z - \mathbf{p}^2_z < 0}{|\mathbf{p}^1_z - \mathbf{p}^2_z|/\sigma_{\mathrm{U}}\colon \langle \mathbf{p}^1, \mathbf{v}^1\rangle \dashrightarrow \langle \mathbf{p}^2, \mathbf{v}^2\rangle}, \tag{57}$$
$$\frac{\mathbf{p}^1_z - \mathbf{p}^2_z > 0}{|\mathbf{p}^1_z - \mathbf{p}^2_z|/\sigma_{\mathrm{L}}\colon \langle \mathbf{p}^1, \mathbf{v}^1\rangle \dashrightarrow \langle \mathbf{p}^2, \mathbf{v}^2\rangle}. \tag{58}$$

6 Example: co-design

The next example revolves around the construction of norphisms for the category of design problems **DP** [2, 6]; this is called $\mathbf{Feas_{Bool}}$ in [6]. The objects of **DP** are posets. The morphisms are design problems (also referred to as feasibility relations or boolean profunctors). A *design problem* (DP) $d\colon \mathbf{P} \nrightarrow \mathbf{Q}$ is a monotone map of the form $d\colon \mathbf{P}^{\mathrm{op}} \times \mathbf{Q} \to_{\mathbf{Pos}} \mathbf{Bool}$, where $\mathbf{P}, \mathbf{Q}$ are arbitrary posets and **Bool** denotes the poset with elements $\{\bot, \top\}$, with $\bot \leq \top$.

The semantics for a DP is that it describes a process which provides a certain functionality, by requiring certain resources. A design problem d is a monotone map, since lowering the requested functionalities will not require more resources, and increasing the available resources will not provide less functionalities.

Morphism composition is defined as follows. Given DPs $d\colon \mathbf{P} \nrightarrow \mathbf{Q}$ and $e\colon \mathbf{Q} \nrightarrow \mathbf{R}$, their composite is
$$\begin{aligned} (d \mathbin{⨟} e)\colon \mathbf{P}^{\mathrm{op}} \times \mathbf{R} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p, r\rangle &\mapsto \bigvee_{q \in \mathbf{Q}} d(p,q) \wedge e(q,r). \end{aligned} \tag{59}$$

For any poset $\mathbf{P}$, the identity DP $\mathrm{id}_{\mathbf{P}}\colon \mathbf{P} \nrightarrow \mathbf{P}$ is the monotone map
$$\begin{aligned} \mathrm{id}_{\mathbf{P}}\colon \mathbf{P}^{\mathrm{op}} \times \mathbf{P} &\to_{\mathbf{Pos}} \mathbf{Bool}, \\ \langle p_1, p_2\rangle &\mapsto p_1 \preceq_{\mathbf{P}} p_2. \end{aligned} \tag{60}$$

Interpretation of norphisms in DP Given that the morphisms of **DP** are feasibility relations, we expect that the norphisms of **DP** ("nesign problems" NP), should be *infeasibility* relations. We define a nesign problem $n\colon \mathbf{F} \nrightarrow \mathbf{R}$ to be a monotone map $n\colon \mathbf{F}\times\mathbf{R}^{\mathrm{op}} \to \mathbf{Bool}$, and we interpret $n(f,r) = \top$ to mean that it is *not* possible to produce f from r. The idea is that if $\langle f_1, r_1\rangle$ is infeasible, then $f_1 \preceq f_2$ implies that $\langle f_2, r_1\rangle$ is also infeasible and $r_2 \preceq r_1$ implies that $\langle f_1, r_2\rangle$ is also infeasible. Note that the source poset of a nesign problem is the $^{\mathrm{op}}$ of the source poset for a design problem.

Compatibility of morphisms and norphisms Consider a DP $d\colon \mathbf{F} \nrightarrow \mathbf{R}$ and a NP $n\colon \mathbf{F} \nrightarrow \mathbf{R}$. The compatibility relation between DP and NP should ensure that there are no contradictions. We ask that, for any pair of functionality/resources $\langle f, r\rangle$, it cannot happen that they are declared feasible by the DP ($d(f,r)$) *and* declared infeasible by the NP ($n(f,r)$):

$$i_{\mathbf{FR}}(n,d) = \exists f \in \mathbf{F}, r \in \mathbf{R}\colon\ d(f,r) \wedge n(f,r). \tag{61}$$

Composition rules for norphisms Given a NP $n\colon \mathbf{P} \nrightarrow \mathbf{Q}$ and a DP $d\colon \mathbf{R} \nrightarrow \mathbf{Q}$, one can compose them to get a NP $n \multimap d\colon \mathbf{P} \nrightarrow \mathbf{R}$:

$$(n \multimap d)(p,r) = \bigvee_{q\in\mathbf{Q}} n(p,q) \wedge d(r,q). \tag{62}$$

And given a DP $d\colon \mathbf{Q} \nrightarrow \mathbf{P}$ and a NP $n\colon \mathbf{Q} \nrightarrow \mathbf{R}$, one can compose them to get a NP $d \multimap n\colon \mathbf{P} \nrightarrow \mathbf{R}$:

$$(d \multimap n)(p,r) = \bigvee_{q\in\mathbf{Q}} d(q,p) \wedge n(q,r). \tag{63}$$

The composition rules satisfy (equiv-1) and (equiv-2) and are exact, as may easily be checked.

Example 10. Consider the posets $\mathbf{P} = \langle \mathbb{N}_{[\text{kg pears}]}, \leq\rangle$, $\mathbf{Q} = \langle \mathbb{R}_{\geq 0,[\text{CHF}]}, \leq\rangle$, and $\mathbf{R} = \langle \mathbb{N}_{[\text{kg raisins}]}, \leq\rangle$. Consider the design problem $d\colon \mathbf{R} \nrightarrow \mathbf{Q}$ and the nesign problem $n\colon \mathbf{P} \nrightarrow \mathbf{Q}$ given, respectively, by the (in)feasibility relations

$$\frac{d(r,q)}{r\cdot 10 \leq q}, \qquad \frac{n(p,q)}{p\cdot 5 > q}.$$

These say that it is possible to buy raisins at 10 CHF/kg or more, and never possible to buy pears at less than 5 CHF/kg. We can evaluate the composition $(n \multimap d)\colon \mathbf{P} \nrightarrow \mathbf{R}$ in a particular point to understand its meaning. For instance:

$$\begin{aligned}(n \multimap d)(10,4) &= \bigvee_{q\in\mathbf{Q}} n(10,q) \wedge d(4,q)\\ &= \bigvee_{q\in\mathbf{Q}} (40 \leq q < 50) = \top.\end{aligned}$$

This equation is saying that we cannot get 10 kilos of pears from 4 kilos of raisins. The rationale is that, if I could, then I would be able to start with 40 CHF and use d to get 4 kilos of raisins, which I could then use to obtain 10 kilos of pears. But this would contradict the norphism n, because $n(10,40) = \top$ holds and this means that it is infeasible to exchange 40 CHF for 10 kilos of pears.

Norphism schemas Considerations about how to define norphisms might follow from specific knowledge about particular designs that we know are (in)feasible, as well as from more general principles of physics or information theory. One very general rule that is arguably valid across all fields: in this universe, physically realizable designs can never produce strictly more of the same resource than one started with. This rule can be encoded as a norphism. For each object $\mathbf{P}$, we postulate a NP

$$n_{\mathbf{P}}\colon \mathbf{P} \nrightarrow \mathbf{P}, \tag{64}$$

such that

$$n_{\mathbf{P}}(q,p) = p \prec_{\mathbf{P}} q, \tag{65}$$

where $p \prec_{\mathbf{P}} q = (p \preceq_{\mathbf{P}} q) \wedge (p \neq q)$.

Interestingly, starting from any morphism

$$d \colon \mathbf{F} \nrightarrow \mathbf{R}, \tag{66}$$

one can directly obtain two NPs that go in the opposite direction, $\mathbf{R} \nrightarrow \mathbf{F}$. These are

$$(n_{\mathbf{R}} \multimap d)(r,f) = \bigvee_{r' \in \mathbf{R}} n_{\mathbf{R}}(r,r') \wedge d(f,r'), \tag{67}$$

$$(d \multimapdot n_{\mathbf{F}})(r,f) = \bigvee_{f' \in \mathbf{F}} d(f',r) \wedge n_{\mathbf{F}}(f',f). \tag{68}$$

which gives two impossibility results. The first states infeasibility because, while it is possible to get f from r' via d for a certain r', it is not possible to obtain r from r'. The second states infeasibility because, while it is possible to get f' from r via d for a certain f', it is not possible to obtain f' from f. In this nategory, we see that *positive information induces negative information* in the other direction.

7 The category GSet

The dialectica construction **GC** is due to De Paiva [3, 4], and its instantiation in the case $\mathbf{C} = \mathbf{Set}$ has been studied from a "questions and answers" perspective, for example in [1]. We will focus on **GSet**, however our discussion is also interesting for other cases of the **GC** construction.

Definition 11 (GSet). An object of **GSet** is a tuple

$$\langle Q, A, C \rangle, \tag{69}$$

where Q and A are sets, and $C : Q \to_{\mathbf{Rel}} A$ is a relation.

A morphism $\mathbf{r} \colon \langle Q_1, A_1, C_1 \rangle \to_{\mathbf{GC}} \langle Q_2, A_2, C_2 \rangle$ is a pair of maps

$$\mathbf{r} = \langle r_\flat, r^\sharp \rangle, \tag{70}$$

$$r_\flat : Q_1 \leftarrow_{\mathbf{Set}} Q_2, \tag{71}$$

$$r^\sharp : A_1 \to_{\mathbf{Set}} A_2, \tag{72}$$

that satisfy the property

$$\forall q_2 : Q_2 \quad \forall a_1 : A_1 \quad r_\flat(q_2)\, C_1\, a_1 \;\Rightarrow\; q_2\, C_2\, r^\sharp(a_1). \tag{73}$$

Morphism composition is defined component-wise

$$(\mathbf{r} \mathbin{\fatsemi} \mathbf{s})_\flat = s_\flat \mathbin{\fatsemi} r_\flat, \tag{74}$$

$$(\mathbf{r} \mathbin{\fatsemi} \mathbf{s})^\sharp = r^\sharp \mathbin{\fatsemi} s^\sharp, \tag{75}$$

and satisfies (73) via composition of implications.

The identity at $\langle Q, A, C \rangle$ is $\mathrm{id}_{\langle Q, A, C \rangle} = \langle \mathrm{id}_Q, \mathrm{id}_A \rangle$.

Remark 12. Our notation was chosen to facilitate a "questions and answers" interpretation [1]. In this perspective, an object of **GSet** is a "problem": a relation C between a set of questions Q and a set of answers A. For a particular question $q \in Q$ and answer $a \in A$, $q\,C\,a$ means that the answer is correct for the question. A morphism $\mathbf{r} \colon \langle Q_1, A_1, C_1 \rangle \to_{\mathbf{GC}} \langle Q_2, A_2, C_2 \rangle$ is a *reduction* of problem 2 to problem 1, in the sense that we can use a solution to problem 1 to solve problem 2. The idea is that we start from a question q_2 and transform it to a question $q_1 = r_\flat(q_2)$ of the first problem. Assuming we can find an answer a_1 to q_1, we can then transform it in an answer of the second problem $a_2 = r^\sharp(a_1)$. The condition (73) ensures that the answer so produced is correct for the second problem.

We now rewrite the objects of **GSet** in a slightly different way. Instead of

$$C : Q \to_{\mathbf{Rel}} A, \tag{76}$$

we can write this relation as a boolean function

$$\kappa\colon Q \times A \to \{\bot, \top\}. \tag{77}$$

Letting **Bool** denote the category with two objects "$\bot$" and "$\top$" and a single non-identity morphism $\Rightarrow\colon \bot \to_{\mathbf{Bool}} \top$, we can rewrite (77) again as

$$\kappa\colon Q \times A \to \mathrm{Ob}_{\mathbf{Bool}}. \tag{78}$$

And then condition (73) can be rewriten as a dependent function

$$r^* : \{q_2 : Q_2, a_1 : A_1\} \to \kappa_1(r_\flat(q_2), a_1) \to_{\mathbf{Bool}} \kappa_2(q_2, r^\sharp(a_1)). \tag{79}$$

The value $r^*(q_2, a_1)$ is a morphism in **Bool** that witnesses an implication.

Remark 13. One idea that we find interesting is to replace **Bool** with some other category **B** (on which we may wish to place suitable assumptions) and can consider maps of the form

$$\kappa\colon Q \times A \to \mathrm{Ob}_{\mathbf{B}}. \tag{80}$$

If **B** is some category whose morphisms are "proofs", the analogue of (79) chooses a proof which is just one among many possible proofs (and such proofs might themselves be ordered by relevance or other criteria).

7.1 A monoidal product for GSet

The categories **GC** have a very rich structure. In particular they provide models of linear logic with four distinct monoidal products $\otimes$, $⅋$, $\oplus$, and &. We define here a monoidal product $\sqcup$ for **GSet** which one might say is "in between" the the multiplicative connectives $\otimes$ and $⅋$ (which are denoted $\oslash$ and $\Box$, respectively, in [3, 4]).

Definition 14 (Monoidal product $\sqcup$)**.** On objects,

$$\langle Q_1, A_1, \kappa_1 \rangle \sqcup \langle Q_2, A_2, \kappa_2 \rangle = \langle Q_1^{A_2} \times Q_2^{A_1}, A_1 \times A_2, \kappa_1 \sqcup \kappa_2 \rangle, \tag{81}$$

where

$$\kappa_1 \sqcup \kappa_2 : \langle \langle q_1, q_2 \rangle, \langle a_1, a_2 \rangle \rangle \mapsto \kappa_1(q_1(a_2), a_1) \vee \kappa_2(q_2(a_1), a_2). \tag{82}$$

The product of morphisms $\mathbf{r}\colon \langle Q_1, A_1, \kappa_1 \rangle \to \langle Q_3, A_3, \kappa_3 \rangle$ and $\mathbf{s}\colon \langle Q_2, A_2, \kappa_2 \rangle \to \langle Q_4, A_4, \kappa_4 \rangle$ is

$$\mathbf{r} \sqcup \mathbf{s}\colon \langle Q_1^{A_2} \times Q_2^{A_1}, A_1 \times A_2, \kappa_1 \sqcup \kappa_2 \rangle \to \langle Q_3^{A_4} \times Q_4^{A_3}, A_3 \times A_4, \kappa_3 \sqcup \kappa_4 \rangle, \tag{83}$$

with

$$(\mathbf{r} \sqcup \mathbf{s})_\flat = \langle s^\sharp \,⨟\, - \,⨟\, r_\flat, r^\sharp \,⨟\, - \,⨟\, s_\flat \rangle, \tag{84}$$

$$(\mathbf{r} \sqcup \mathbf{s})^\sharp = r^\sharp \times s^\sharp, \tag{85}$$

$$(\mathbf{r} \sqcup \mathbf{s})^*\colon \langle \langle q_3, q_4 \rangle, \langle a_1, a_2 \rangle \rangle \mapsto r^*(s^\sharp \,⨟\, q_3(a_2), a_1) \vee s^*(r^\sharp \,⨟\, q_3(a_1), a_2). \tag{86}$$

The monoidal unit is

$$1_\sqcup = \langle \{\bullet\}, \{\bullet\}, \bot \rangle, \qquad \bot\colon \langle \bullet, \bullet \rangle \mapsto \bot. \tag{87}$$

For the associator and unitors we make the canonical choices, which are easily inferred from their signatures. We refrain from writing them out explicitly here. For reasons of space we also omit the the proof that $\langle \mathbf{GSet}, \sqcup \rangle$ is indeed a monoidal category.

Remark 15. In the generalization where we replace **Bool** with some category **B**, the operation $\vee$ and the object $\bot$ in **Bool** which are used in the above definition would be replaced by suitable substitutes in **B**.

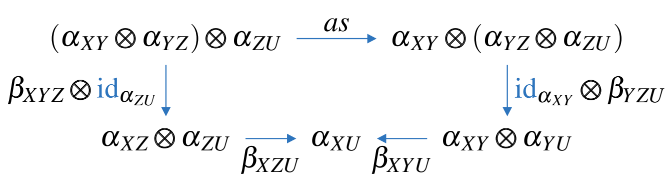
8 Describing nategories using enrichment

We recall the following standard definition of enriched category [7], for easy reference and to fix notation.

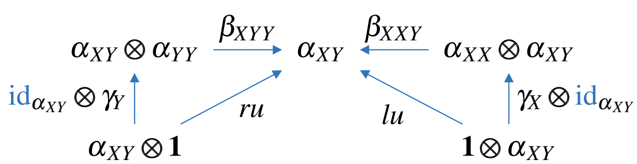
Definition 16 (Enriched category)**.** Let $\langle \mathbf{V}, \otimes, \mathbf{1}, as, lu, ru \rangle$ be a monoidal category.
A **V**-enriched category **E** is a tuple $\langle \mathrm{Ob}_{\mathbf{E}}, \alpha, \beta, \gamma \rangle$, where

1. $\mathrm{Ob}_{\mathbf{E}}$ is a collection of objects.
2. α is a function such that, for all pairs of objects $X, Y \in \mathrm{Ob}_{\mathbf{E}}$, its value α_{XY} is an object of **V**, called a *hom-object*.
3. β is a function such that, for all $X, Y, Z \in \mathrm{Ob}_{\mathbf{E}}$, there exists a morphism β_{XYZ} of **V**
$$\beta_{XYZ} \colon \alpha_{XY} \otimes \alpha_{YZ} \to_{\mathbf{V}} \alpha_{XZ}, \tag{88}$$
called a *composition morphism*.
4. γ is a function such that, for each $X \in \mathrm{Ob}_{\mathbf{E}}$, there exists a morphism of **V**
$$\gamma_X \colon \mathbf{1} \to_{\mathbf{V}} \alpha_{XX}, \tag{89}$$
called an *identity-choosing morphism*.

Moreover, for any $X, Y, Z, U \in \mathrm{Ob}_{\mathbf{E}}$, the following diagrams must commute.



(90)



(91)

Recall that specifying the data of an ordinary (locally small) category is equivalent to specifying a category enriched in the monoidal category $\mathbf{P} = \langle \mathbf{Set}, \times, 1 \rangle$. In this case, both the enriched category and the ordinary category have the same objects, the hom-objects of the enriched category correspond to the hom-sets of the ordinary category, the composition morphisms encode the composition operations, and the identity-choosing morphisms select an element of each of the hom-sets of the type α_{XX}, corresponding to identity morphisms. The diagrams (90) and (91) encode the associativity of the composition operations, and that the identity morphisms act neutrally for composition.

The proof of our main result below follows a similar pattern – we show that to specify a nategory which satisfies some additional conditions it is sufficient to specify a category enriched in the monoidal category $\langle \mathbf{GSet}, \sqcup \rangle$. We will denote $\langle \mathbf{GSet}, \sqcup \rangle$ by **PN**, which stands for "positive" and "negative".

Proposition 17. A **PN**-enriched category provides the data necessary to specify a nategory. However, not all nategories can be specified by the data of a **PN**-enriched category, because the nategory produced has the following additional properties, which encode a covariant and a contravariant "action" of morphisms on norphisms.
Identities act neutrally:

$$\mathrm{id} \bullet\!\!\rightarrow n = n, \tag{neut-1}$$
$$n \rightarrow\!\!\bullet\, \mathrm{id} = n. \tag{neut-2}$$

Compatibility with composition:

$$(f \fatsemi g) \multimap n = g \multimap (f \multimap n), \qquad \text{(covar)}$$
$$n \multimapinv (g \fatsemi h) = (n \multimapinv h) \multimapinv g. \qquad \text{(contravar)}$$

The actions commute:

$$f \multimap (n \multimapinv h) = (f \multimap n) \multimapinv h. \qquad \text{(comm)}$$

These conditions are not satisfied by all nategories.

Proof. Suppose somebody has provided us with a **PN**-enriched category $\mathbf{E} = \langle \mathrm{Ob}_{\mathbf{E}}, \alpha, \beta, \gamma \rangle$. Using this data we will describe a nategory **C** with the above-stated properties.

For the objects of **C**, we set $\mathrm{Ob}_{\mathbf{C}} := \mathrm{Ob}_{\mathbf{E}}$.

For every pair of objects $X, Y \in \mathrm{Ob}_{\mathbf{C}}$, we have an object α_{XY} of **PN**. This is a tuple

$$\alpha_{XY} = \langle Q, A, \kappa \rangle, \tag{92}$$

which we interpret as

$$\alpha_{XY} = \langle \mathrm{Nom}_{\mathbf{C}}(X;Y), \mathrm{Hom}_{\mathbf{C}}(X;Y), i_{XY} \rangle, \tag{93}$$

thereby setting $\mathrm{Nom}_{\mathbf{C}}(X;Y) := Q$, $\mathrm{Hom}_{\mathbf{C}}(X;Y) := A$, and $i_{XY} := \kappa$.

Next, for each $X \in \mathrm{Ob}$ we have an identity-choosing morphism

$$\gamma_X : \mathbf{1}_{\mathbf{PN}} \to_{\mathbf{PN}} \alpha_{XX}. \tag{94}$$

Because $\mathbf{1}_{\mathbf{PN}} = \langle \{\bullet\}, \{\bullet\}, \bot \rangle$, this is a morphism

$$\gamma_X : \langle \{\bullet\}, \{\bullet\}, \bot \rangle \to_{\mathbf{PN}} \langle \mathrm{Nom}_{\mathbf{C}}(X;X), \mathrm{Hom}_{\mathbf{C}}(X;X), i_{XX} \rangle, \tag{95}$$

which consists of three functions $\langle r_\flat, r^\sharp, r^* \rangle$. The forward map $r^\sharp : \{\bullet\} \to \mathrm{Hom}_{\mathbf{C}}(X;X)$ chooses our (candidate) identity morphism, so we set $\mathrm{id}_X := r^\sharp(\bullet)$. The backward map $r_\flat : \mathrm{Nom}_{\mathbf{C}}(X;X) \to \{\bullet\}$ is uniquely determined and does not carry any information. As for r^*, it is a dependent function of the type

$$r^* : \{q_2 : \mathrm{Nom}_{\mathbf{C}}(X;X), a_1 : \{\bullet\}\} \to \bot(r_\flat(q_2), a_1) \to_{\mathbf{Bool}} i_{XX}(q_2, r^\sharp(a_1)). \tag{96}$$

Evaluated at $q_2 = n$ and $a_1 = \bullet$, we have

$$r^*(n, \bullet) : \bot \to_{\mathbf{Bool}} i_{XX}(n, \mathrm{id}_X). \tag{97}$$

Because $\bot$ is an initial object in **Bool**, such a morphism always exists, no matter what the right-hand side is. Therefore, this condition does not carry any additional information.

Now let us fix three objects X, Y, Z and consider the composition morphism

$$\beta_{XYZ} : \alpha_{XY} \otimes_{\mathbf{PN}} \alpha_{YZ} \to_{\mathbf{PN}} \alpha_{XZ}. \tag{98}$$

Rewriting the hom-objects as tuples and using abbreviated notation we have

$$\beta_{XYZ} : \langle \mathrm{N}_{XY}, \mathrm{H}_{XY}, i_{XY} \rangle \otimes_{\mathbf{PN}} \langle \mathrm{N}_{YZ}, \mathrm{H}_{YZ}, i_{YZ} \rangle \to_{\mathbf{PN}} \langle \mathrm{N}_{XZ}, \mathrm{H}_{XZ}, i_{XZ} \rangle. \tag{99}$$

Expanding using the definition of $\otimes_{\mathbf{PN}}$ we find

$$\beta_{XYZ} : \langle {\mathrm{N}_{XY}}^{\mathrm{H}_{YZ}} \times {\mathrm{N}_{YZ}}^{\mathrm{H}_{XY}}, \mathrm{H}_{XY} \times \mathrm{H}_{YZ}, i_{XY} \sqcup i_{YZ} \rangle \to_{\mathbf{PN}} \langle \mathrm{N}_{XZ}, \mathrm{H}_{XZ}, i_{XZ} \rangle. \tag{100}$$

Such a morphism corresponds to three maps $\langle s_\flat, s^\sharp, s^* \rangle$. The forward map $s^\sharp$ has type

$$s^\sharp : \mathrm{Hom}_{\mathbf{C}}(X;Y) \times \mathrm{Hom}_{\mathbf{C}}(Y;Z) \to \mathrm{Hom}_{\mathbf{C}}(X;Z), \tag{101}$$

and we use it to define morphism composition "$\mathbin{;}_{\mathbf{C}}$" in our nategory. The backward map $s_\flat$ has type

$$s_\flat \colon \mathrm{N}_{XZ} \to {\mathrm{N}_{XY}}^{\mathrm{H}_{YZ}} \times {\mathrm{N}_{YZ}}^{\mathrm{H}_{XY}}, \tag{102}$$

which, after splitting into two maps (using the universal property of the product) and currying, specifies two maps

$$\multimapinv \colon \mathrm{N}_{XZ} \times \mathrm{H}_{YZ} \to \mathrm{N}_{XY}, \tag{103}$$
$$\multimapboth \colon \mathrm{H}_{XY} \times \mathrm{N}_{XZ} \to \mathrm{N}_{YZ}. \tag{104}$$

As for the dependent function s^*, given $n : \mathrm{N}_{XZ}$, $f : \mathrm{H}_{XY}$, $g : \mathrm{H}_{YZ}$ we have

$$s^*(n, \langle f, g\rangle) \colon (i_{XY} \sqcup i_{YZ})(\langle (n \multimapinv -), (- \multimapboth n)\rangle, \langle f, g\rangle) \to_{\mathbf{Bool}} i_{XZ}(n, f \mathbin{;} g). \tag{105}$$

Expanding more,

$$s^*(n, \langle f, g\rangle) \colon i_{XY}(n \multimapinv g, f) \vee i_{YZ}(f \multimapboth n, g) \to_{\mathbf{Bool}} i_{XZ}(n, f \mathbin{;} g), \tag{106}$$

which is equivalent to having two maps

$$s^*_1(n, \langle f, g\rangle) \colon i_{XY}(n \multimapinv g, f) \to_{\mathbf{Bool}} i_{XZ}(n, f \mathbin{;} g), \tag{107}$$
$$s^*_2(n, \langle f, g\rangle) \colon i_{YZ}(f \multimapboth n, g) \to_{\mathbf{Bool}} i_{XZ}(n, f \mathbin{;} g). \tag{108}$$

These witness the implications which give us (equiv-1) and (equiv-2).

Next we move to the commutative diagrams in the definition of enriched category. As a general observation, we note that for diagrams in **GSet** we only need to consider commutativity on the level of "forward maps" and "backward maps" respectively. We do not need to worry about the conditions (73), because for any two parallel morphisms this condition is the same, and hence "commutativity" is trivially satisfied.

Conditions from the associativity diagram for enriched categories We now consider the diagram (90), in the case of **PN**. On the level of "forward" maps, this commutative diagram encodes that morphism composition must be associative. One the level of "backward" maps, it implies that the following diagram must commute:

$$\begin{array}{ccccc}
\left(Q_{XY}^{A_{YZ}} \times Q_{YZ}^{A_{XY}}\right)^{A_{ZU}} \times Q_{ZU}^{A_{XY}\times A_{YZ}} & & \xleftarrow{\;as_\flat\;} & & Q_{XY}^{A_{YZ}\times A_{ZU}} \times \left(Q_{YZ}^{A_{ZU}} \times Q_{ZU}^{A_{YZ}}\right)^{A_{XY}} \\
\uparrow{\scriptstyle (\beta_{XYZ} \sqcup \mathrm{id}_{ZU})_\flat} & & & & \uparrow{\scriptstyle (\mathrm{id}_{XY} \sqcup \beta_{YZU})_\flat} \\
Q_{XZ}^{A_{ZU}} \times Q_{ZU}^{A_{XZ}} & \xleftarrow[\beta_{XZU\flat}]{} & Q_{XU} & \xrightarrow[\beta_{XYU\flat}]{} & Q_{XY}^{A_{YU}} \times Q_{YU}^{A_{XY}}
\end{array}$$

Let us look at the two different routes through this diagram. For the left-hand route, note that

$$\begin{aligned}
(\beta_{XYZ} \sqcup \mathrm{id}_{ZU})_\flat &= \langle {\mathrm{id}_U}^\sharp \mathbin{;} (-) \mathbin{;} \beta_{XYZ\flat}, {\beta_{XYZ}}^\sharp \mathbin{;} (-) \mathbin{;} \mathrm{id}_{ZU\flat}\rangle \\
&= \langle (-) \mathbin{;} \beta_{XYZ\flat}, {\beta_{XYZ}}^\sharp \mathbin{;} (-)\rangle,
\end{aligned} \tag{109}$$

and so

$$\begin{aligned}
\beta_{YZU\flat} \mathbin{;} (\beta_{XYZ} \sqcup \mathrm{id}_{ZU})_\flat \colon Q_{XU} &\to Q_{XZ}^{A_{ZU}} \times Q_{ZU}^{A_{XZ}} \to \left(Q_{XY}^{A_{YZ}} \times Q_{YZ}^{A_{XY}}\right)^{A_{ZU}} \times Q_{ZU}^{A_{XY}\times A_{YZ}} \\
q &\mapsto \langle q \multimapinv (-), (-) \multimapboth q\rangle \mapsto \langle q \multimapinv (-) \mathbin{;} \beta_{XYZ\flat}, {\beta_{XYZ}}^\sharp \mathbin{;} (-) \multimapboth q\rangle.
\end{aligned} \tag{110}$$

For the right-hand route, note that

$$\begin{aligned}(\mathrm{id}_{XY} \sqcup \beta_{YZU})_\flat &= \langle {\beta_{YZU}}^\sharp \fatsemi (-) \fatsemi \mathrm{id}_{XY_\flat}, {\mathrm{id}_{XY}}^\sharp \fatsemi (-) \fatsemi \beta_{YZU_\flat}\rangle \\ &= \langle {\beta_{YZU}}^\sharp \fatsemi (-), (-) \fatsemi \beta_{YZU_\flat}\rangle,\end{aligned} \tag{111}$$

and so

$$\begin{aligned}\beta_{XYU_\flat} \fatsemi (\mathrm{id}_{XY} \sqcup \beta_{YZU})_\flat \colon Q_{XU} &\to Q_{XY}^{A_{YU}} \times Q_{YU}^{A_{XY}} \to Q_{XY}^{A_{YZ}\times A_{ZU}} \times \left(Q_{YZ}^{A_{ZU}} \times Q_{ZU}^{A_{YZ}}\right)^{A_{XY}} \\ q &\mapsto \langle q \mathbin{-\!\bullet} (-), (-) \mathbin{\bullet\!-\!\bullet} q\rangle \mapsto \langle {\beta_{YZU}}^\sharp \fatsemi q \mathbin{-\!\bullet} (-), (-) \mathbin{\bullet\!-\!\bullet} q \fatsemi \beta_{YZU_\flat}\rangle.\end{aligned} \tag{112}$$

Instead of now applying $as_\flat$ directly, which is an obvious map but messy to write down, we evaluate the functions we obtained from our calculations for the left- and right-hand routes. Given $\langle f, g, h\rangle \in A_{XY} \times A_{YZ} \times A_{ZU}$, evaluating the two components of (110) we find

$$(q \mathbin{-\!\bullet} (-) \fatsemi \beta_{XYZ_\flat})(h) = \beta_{XYZ_\flat}(q \mathbin{-\!\bullet} h) \colon \langle g, f\rangle \mapsto \langle (q \mathbin{-\!\bullet} h) \mathbin{-\!\bullet} g, f \mathbin{\bullet\!-\!\bullet} (q \mathbin{-\!\bullet} h)\rangle, \tag{113}$$

and

$$({\beta_{XYZ}}^\sharp \fatsemi (-) \mathbin{\bullet\!-\!\bullet} q)(\langle f, g\rangle) = (f \fatsemi g) \mathbin{\bullet\!-\!\bullet} q, \tag{114}$$

respectively.

For the right-hand route, evaluating (112) gives

$$({\beta_{YZU}}^\sharp \fatsemi q \mathbin{-\!\bullet} (-))(\langle g, h\rangle) = q \mathbin{-\!\bullet} (g \fatsemi h), \tag{115}$$

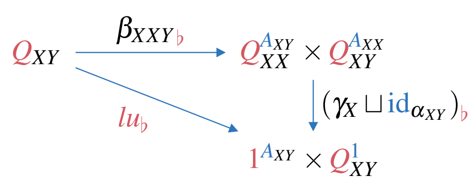
and

$$((-) \mathbin{\bullet\!-\!\bullet} q \fatsemi \beta_{YZU_\flat})(f) = \beta_{YZU_\flat}(f \mathbin{\bullet\!-\!\bullet} q) \colon \langle h, g\rangle \mapsto \langle (g \mathbin{\bullet\!-\!\bullet} q) \mathbin{-\!\bullet} f, g \mathbin{\bullet\!-\!\bullet} (f \mathbin{\bullet\!-\!\bullet} q)\rangle. \tag{116}$$

By comparing the two routes, we obtain the conditions

$$\begin{aligned}(f \fatsemi g) \mathbin{\bullet\!-\!\bullet} q &= g \mathbin{\bullet\!-\!\bullet} (f \mathbin{\bullet\!-\!\bullet} q), \\ q \mathbin{-\!\bullet} (g \fatsemi h) &= (q \mathbin{-\!\bullet} h) \mathbin{-\!\bullet} g, \\ f \mathbin{\bullet\!-\!\bullet} (q \mathbin{-\!\bullet} h) &= (f \mathbin{\bullet\!-\!\bullet} q) \mathbin{-\!\bullet} h.\end{aligned} \tag{117}$$

Conditions from the unitality diagrams for enriched categories Consider the right-hand portion of the diagram (91), now for the case of **PN**. On the level of forward maps, this diagram encodes the condition that $\mathrm{id}_X \fatsemi a = a$ for any morphism $a \colon X \to Y$. On the level of backward maps, it amounts to the commutative diagram



Computing the right-hand route, we have

$$\begin{aligned}(\gamma_X \sqcup \mathrm{id}_{\alpha_{XY}})_\flat \colon Q_{XX}^{A_{XY}} \times Q_{XY}^{A_{XX}} &\to 1^{A_{XY}} \times Q_{XY}^1 \\ \langle \varphi_{XX}, \varphi_{XY}\rangle &\mapsto \langle !, {\gamma_X}^\sharp \fatsemi \varphi_{XY} \fatsemi \mathrm{id}_{\alpha_{XY}\flat}\rangle = \langle !, \bullet \mapsto \varphi_{XX}(\mathrm{id}_X)\rangle,\end{aligned} \tag{118}$$

and

$$\begin{aligned}((\gamma_X \sqcup \mathrm{id}_{\alpha_{XY}}) \fatsemi \beta_{XXY})_\flat \colon Q_{XY} &\to Q_{XX}^{A_{XY}} \times Q_{XY}^{A_{XX}} \to 1^{A_{XY}} \times Q_{XY}^1 \\ q &\mapsto {\beta_{XXY}}^\sharp(q) = \langle q \mathbin{-\!\bullet} (-), (-) \mathbin{\bullet\!-\!\bullet} q\rangle \\ &\mapsto \langle {\mathrm{id}_{\alpha_{XY}}}^\sharp \fatsemi q \mathbin{-\!\bullet} (-) \fatsemi \gamma_{X_\flat}, {\gamma_X}^\sharp \fatsemi (-) \mathbin{\bullet\!-\!\bullet} q \fatsemi \mathrm{id}_{\alpha_{XY}\flat}\rangle \\ &= \langle !, \bullet \mapsto (\mathrm{id}_X \mathbin{\bullet\!-\!\bullet} q)\rangle,\end{aligned} \tag{119}$$

which, when compared with $lu_\flat \colon q \mapsto \langle !, \bullet \mapsto q \rangle$, gives the condition

$$\mathrm{id} \mathbin{\bullet\!\!-\!\!\bullet} q = q. \tag{120}$$

The left-hand portion of the diagram (91) may be treated analogously and gives rise the to the conditions

$$a \mathbin{⨟} \mathrm{id}_Y = a \qquad \text{and} \qquad q \mathbin{-\!\bullet\!-} \mathrm{id} = q. \tag{121}$$

□

Remark 18. Exact nategories are those in which the implications in (equiv-1) and (equiv-2) are in fact equivalences. In the above proof, this corresponds to the morphisms (107) and (108) in **Bool** being identities.

(17) begs the question as to why we don't include the additional properties stated there as part of our definition of what a nategory is. Our reason is that these properties fail for examples of interest to us.

For example, in applications it is normal that physical measurements and numerical representations on a computer are given only to a certain accuracy. This motivates (9), where we only allow integer values for norphisms. However, in that example, the properties (contravar) and (covar) are not satisfied. To see this, consider (contravar)and consider morphisms g and h in **Berg** with $\mathrm{length}(g) = \mathrm{length}(h) = 1.5$. For a norphism q of the appropriate signature we have

$$q \mathbin{-\!\bullet\!-} (g \mathbin{⨟} h) = \mathsf{floor}(q - \mathrm{length}(g \mathbin{⨟} f)) = \mathsf{floor}(q - 3) \tag{122}$$

on the one hand, and

$$(q \mathbin{-\!\bullet\!-} h) \mathbin{-\!\bullet\!-} g = \mathsf{floor}(\mathsf{floor}(q - 1.5) - 1.5) \tag{123}$$

on the other. If we choose $q = 10$, for instance, the previous expressions evaluate to 7 and 6, respectively.

As a different example, the properties (neut-1) and (neut-2) fail in Example 6. Indeed, there $\mathrm{id}_X \mathbin{\bullet\!\!-\!\!\bullet} n = \emptyset$ and $n \mathbin{-\!\bullet\!-} \mathrm{id}_Y = \emptyset$, even though, in general, we would have $n \neq \emptyset$.

9 Conclusions

This work showed that we can encode negative information in a categorical manner such that norphisms (negative arrows) and morphisms (positive arrows) are equal citizens in the theory. Norphisms and morphisms are, in general, not mutually exclusive; they give complementary information.

We have seen how, in the category **Berg**, norphisms can represent negative results such as lower bounds on distances between two locations. A path planning algorithm must construct a morphism (a path) *and* construct a norphism (a bound) to prove that the path is optimal. We have also seen how, in the category **DP**, norphisms can represent design impossibility results.

After defining nategories as categories with extra structure, we showed a way to encode this new concept using categories enriched in the dialecta category **GSet**. This approach, however, introduces some compatibility properties for norphism composition that we do not wish to include in our general notion of nategory. Future work includes exploring if nategories might be recovered, on the nose, via a different enrichment, as well as studying various typical categorical concepts in the context of nategories. We would also be very happy to discover further interesting examples.

Acknowledgments

The authors would like to thank David I. Spivak for fruitful discussions. We also thank Valeria de Paiva, Brendan Fong, and David Yetter for helpful comments. Gioele Zardini was supported by the Swiss National Science Foundation, under NCCR Automation, grant agreement 51NF40_180545.

References

[1] Andreas Blass (1995): *Questions and answers—a category arising in linear logic, complexity theory, and set theory*. *Advances in linear logic* 222, pp. 61–81, doi:10.48550/ARXIV.MATH/9309208.

[2] Andrea Censi, Jonathan Lorand & Gioele Zardini (2022): *Applied Compositional Thinking for Engineers*. Available at `https://bit.ly/3qQNrdR`. Work in progress book.

[3] Valeria De Paiva (1989): *The dialectica categories*. *Categories in Computer Science and Logic* 92, pp. 47–62, doi:10.48456/tr-213.

[4] Valeria De Paiva (1989): *A dialectica-like model of linear logic*. In: *Category Theory and Computer Science*, Springer, pp. 341–356, doi:10.1007/BFb0018360.

[5] Daniel Delling, Peter Sanders, Dominik Schultes & Dorothea Wagner (2009): *Engineering route planning algorithms*. In: *Algorithmics of large and complex networks*, Springer, pp. 117–139, doi:10.1007/978-3-642-02094-0_7.

[6] Brendan Fong & David I Spivak (2018): *Seven sketches in compositionality: An invitation to applied category theory*. *arXiv preprint arXiv:1803.05316*, doi:10.48550/ARXIV.1803.05316.

[7] Gregory Maxwell Kelly (1982): *Basic concepts of enriched category theory*. 64, CUP Archive.

[8] Michael Shulman (2022): *Affine logic for constructive mathematics*. *The Bulletin of Symbolic Logic*, pp. 1–51, doi:10.1017/bsl.2022.28.

A Category of Surface-Embedded Graphs

Malin Altenmüller[1] Ross Duncan[1,2]

malin.altenmuller@strath.ac.uk ross.duncan@strath.ac.uk

[1]Department of Computer and Information Sciences, University of Strathclyde,
26 Richmond Street, Glasgow, G1 1XH, UK

[2]Quantinuum, Terrington House, 13–15 Hills Road, Cambridge CB2 1NL, UK

We introduce a categorical formalism for rewriting surface-embedded graphs. Such graphs can represent string diagrams in a non-symmetric setting where we guarantee that the wires do not intersect each other. The main technical novelty is a new formulation of double pushout rewriting on graphs which explicitly records the boundary of the rewrite. Using this boundary structure we can augment these graphs with a rotation system, allowing the surface topology to be incorporated.

1 Introduction

String diagrams [17] are a graphical formalism to reason about monoidal categories. Equational reasoning in *symmetric* string diagrams can be implemented as graph or hyper-graph rewriting subject to various side conditions to capture the precise flavour of the monoidal category intended [5, 6, 7, 14, 2, 1]. We want to use string diagrammatic reasoning for monoidal categories which are not necessarily symmetric. Informally, the lack of symmetry is often stated as “the wires cannot cross” – but what does that mean when the string diagram is a graph or other combinatorial object? Where is this “crossing” taking place? To make sense of this we must move beyond the situation where only the connectivity matters and add some topological information.

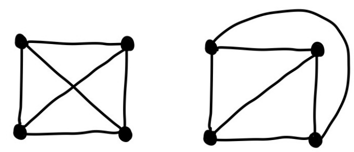
In this paper we make two steps in that direction. Firstly we borrow a tool from topological graph theory – rotation systems – and use it to define a category of graphs which are embedded in some surface. Secondly, we introduce a new refinement of double pushout rewriting [9] which is adapted to this category. This refinement was motivated by the desire to do rewriting on rotation systems, however it works on conventional directed graphs equally well, and removes many annoyances encountered when using standard techniques from algebraic graph rewriting in string diagrams. This is an important step towards formalising non-symmetric string diagrams and their rewriting theory.

Our motivation is also twofold. From the abstract point of view, non-symmetric string diagrams can capture a larger class of theories, including both the symmetric case and the braided monoidal one. A more practical motivation comes from the area of quantum computing, where string diagrams are often used to model quantum circuits [3], their connectivity restrictions imposed by the qubit architecture [4] require a theory without implicit SWAP gates, and can involve circuits defined on quite complex surfaces.

Curiously, Joyal and Street’s original work [12] formalised monoidal categories as plane embedded diagrams, and used the plane to carry the categorical structure. Our work goes in the opposite direction: to recover the topology from the combinatorial structure.

Graph Embeddings Graphs can be drawn on surfaces, and the same graph be drawn different ways on the same surface, as shown below.

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 41–62, doi:10.4204/EPTCS.380.3



If, like the one on the right, the drawing does not intersect itself then it defines an *embedding* of the graph into the surface. If a graph can be embedded in the plane (or equivalently surface of the sphere) it is called *planar*. However in this work we will be concerned with graphs with a given embedding into some closed compact surface, which need not be the plane.

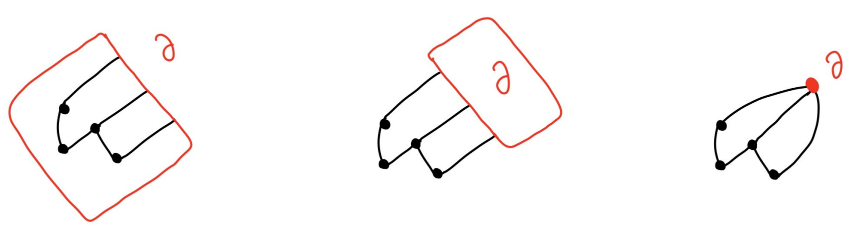
Dealing with lines and points as submanifolds of some surface (up to homeomorphism) is quite unwieldy, so we use a combinatorial representation of graph embeddings called *rotation systems*. A rotation system imposes an order on the edges incident at vertex (called a *rotation*). The rotation information at each vertex is enough to fix the embedding of the graph into some surface, as it defines the faces of the embedding uniquely. This is a well studied topic in graph theory and we refer to the literature for more details [10].

Theorem 1.1. *A rotation system determines the embedding of a connected graph into a minimal surface up to homeomorphism [11, 8, 10].*

Note that different rotation systems for the same graph may have different minimal surfaces, which need not be the plane.

Boundary Graphs and Partitioning Spans When using string diagrams, graphs as usually defined are not the most natural object; rather, we often think about *open* graphs which have "half-edges" or "dangling wires" which represent the domain and codomain of the morphism in question. The half-edges therefore provide the interface along which morphisms compose, and also where substitutions can be made in rewriting. Unfortunately half-edges don't work particularly well with double pushout rewriting, necessitating various workarounds encoding the "wires" as special vertices in a graph [7] or hypergraph [1]. This in turn leads to its own complications when we consider the identity morphism, and other natural transformations which are naturally "all string"; equations which should be trivial are no longer so. Surface embedded graphs suggest a different approach to this question.

Naively, when picturing a rewrite on a surface embedded graph, we picture a disc-like region of the surface which is removed and replaced. The edges which cross the boundary of this region define the interface and we naturally require that the removed disc and its replacement should have the same interface. From the outside, this disc is homeomorphic to a point, so it can be treated as if it was a vertex equipped with a rotation system. However, the perspective from inside and outside the region are completely equivalent, so we can dually view the rest of the graph as a single vertex connected to the interior of the disc. We think of and draw a graph with boundary in three different ways:



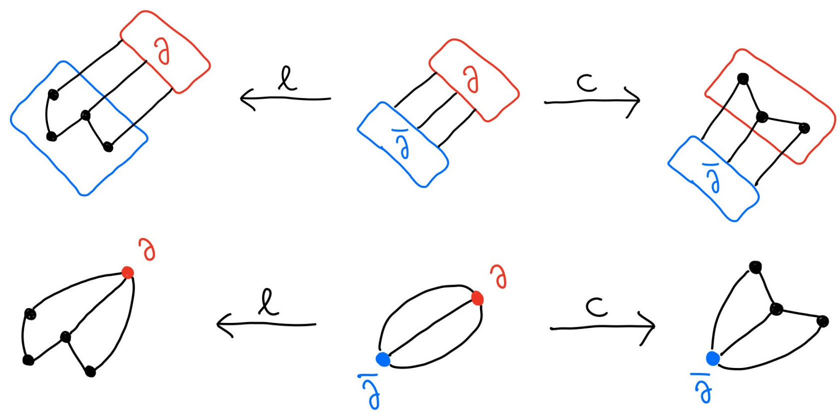


Figure 1: Example of a partitioning span, drawn in a region- and a vertex-style (edge directions are omitted for readability)

On the left, the graph is depicted as a region of the surface with its outside being the rest of the surface. In the middle, graph and its surrounding are both regions of the surface, and on the right we have drawn the boundary as a vertex with the interconnecting edges attached.

This leads naturally to our notion of *boundary graph*: we contract both subgraphs on either side of the boundary to points, leaving a two-vertex graph whose edges specify the connections crossing the boundary. Boundary graphs form the vertex of *partitioning spans*, which specify the whole graph as the two parts, as shown in Figure 1; the pushout of a partitioning span is the original graph.

This formalism allow us to use a simple definition of graph, although our morphisms are now built from partial functions, which introduces some complications around the required injectivity properties to preserve the type of the vertices, which is essential if these graphs are to be interpreted as string diagrams.

Limitations The astute reader will have noted that Theorem 1.1 applies only to connected graphs. To specify an embedding of a disconnected graph a rotation system does not suffice. We would also need to take into account the relationship between components and faces of the graph. We have made no attempt to do so here.

2 A Suitable Category of Graphs

In this section we will introduce a category of directed graphs without reference to any topological structure. The main difficulty here is arriving at the correct notion of graph morphism: our intent here is that the graphs represent terms in some monoidal category – i.e. string diagrams – and the morphisms represent *embeddings* of subterms. This implies that certain structures should be preserved which conventional graph rewriting does not worry about. Our choices here are also influenced by the variant of double-pushout rewriting we will define in the next section. In later sections we will show how to incorporate the plane topology by adding rotation systems.

A total graph is a functor $G : (\bullet \rightrightarrows \bullet) \to \mathbf{Set}$. Concretely, such a graph is a pair of sets V and E, of *vertices* and *edges* respectively, and a pair of functions s and t assigning *source* and *target* vertices to each edge.

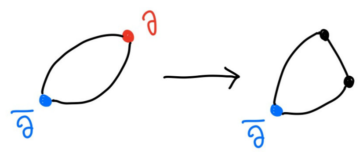
$$E \overset{s}{\underset{t}{\rightrightarrows}} V.$$

In the functor category $[\bullet \rightrightarrows \bullet, \mathbf{Set}]$, a morphism of graphs is a pair of functions $f_V : V \to V', f_E : E \to E'$, such that the following squares commute:

$$\begin{array}{ccc} E & \xrightarrow{f_E} & E' \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle s'} \\ V & \xrightarrow[f_V]{} & V' \end{array} \qquad \begin{array}{ccc} E & \xrightarrow{f_E} & E' \\ {\scriptstyle t}\downarrow & & \downarrow{\scriptstyle t'} \\ V & \xrightarrow[f_V]{} & V' \end{array} \tag{1}$$

Sadly for us, this simple and elegant definition will not suffice.

We want to consider graph morphisms which can replace vertices with subgraphs, and therefore forget these vertices, as shown below:



To achieve this we could operate in a subcategory of $[\bullet \rightrightarrows \bullet, \mathbf{Pfn}]$, the category of partial graphs and maps, with only the total graphs as objects. However this is not quite enough. Commutation of the naturality squares (1) in this category is strict, meaning it includes equality of the domains of definition. Therefore if a morphism forgets a vertex it must also forget all the incident edges at that vertex. This is no use. We address this issue by using the poset enrichment of **Pfn**, and work in the category $[\bullet \rightrightarrows \bullet, \mathbf{Pfn}]_{\leq}$ of functors and *lax* natural transformations:

$$\begin{array}{ccc} E & \xrightarrow{f_E} & E' \\ {\scriptstyle s}\downarrow & \leq & \downarrow{\scriptstyle s'} \\ V & \xrightarrow[f_V]{} & V' \end{array} \qquad \begin{array}{ccc} E & \xrightarrow{f_E} & E' \\ {\scriptstyle t}\downarrow & \leq & \downarrow{\scriptstyle t'} \\ V & \xrightarrow[f_V]{} & V' \end{array} \tag{2}$$

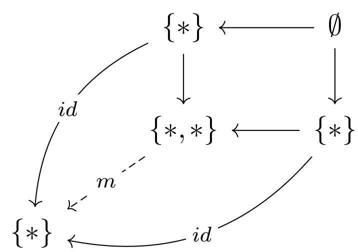
The lax commutation allows the vertex component of a morphism to be undefined at some vertex v while its incident edges may be preserved. However, if an edge is "forgotten" then its source and target vertices must also be so. We'll need more, but let's take $[\bullet \rightrightarrows \bullet, \mathbf{Pfn}]_{\leq}$ as our ambient category for now.

Proposition 2.1. *The category* **Pfn** *of sets and partial functions has pushouts.*

Proof. Given a span $L \xleftarrow{l} B \xrightarrow{c} C$, the elements of the pushout are the same as in for **Set**, but restricted to a subset $B' \subseteq B$, with both $l(b)$ and $c(b)$ defined for $b \in B'$. This is the only way the square commutes for elements in B', and the universal property of the pushout can be derived from **Set**. □

Proposition 2.2. *The category* **Inj** *of sets and injective functions does not have pushouts.*

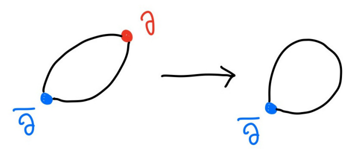
Proof. If pushouts in **Inj** exist, they have to coincide with those of **Set**. Consider the span $\{*\} \longleftarrow \emptyset \longrightarrow \{*\}$, and commuting squares:



In the square all morphisms are injective, but the mediating map out of the pushout $m : \{*,*\} \to \{*\}$ is not. □

We would like to be able to accommodate two further properties in our notion of graph morphism: Firstly, since vertices represent morphisms of a monoidal category, their type should be preserved. Secondly, we want to specify when a morphism is a graph *embedding*, which requires an injectivity property. Merely asking for injectivity of the vertex and edge component is not enough though, our setup requires the edge component to be non-injective, i.e. to represent the identity morphism (or similar circumstances):

Example 2.3. A graph morphism with a non-injective edge component:



Both of the above requirements turn out to be properties of the connection points between vertices and their incident edges, called *flags*:

Definition 2.4. Given a graph (V,E,s,t) its set of *flags* is defined

$$F = \{(e,s(e)) : e \in E\} + \{(e,t(e)) : e \in E\}$$

Given a graph morphism $f : G \to G'$ there is an induced *flag map*, $f_F : F \to F'$,

$$f_F = (f_E \times f_V) + (f_E \times f_V)$$

Note that the flag map is in general a partial map: it is undefined on (e,v), whenever f_V is undefined on v. Whenever f_F is injective we say that f is *flag injective*.

Flag injectivity allows edges to be combined but prevents a morphism from decreasing a vertex degree in the process. However, nothing said so far forbids a morphism from increasing the degree of a vertex: we require a notion of *flag surjectivity*. Given $f : G \to G'$, it doesn't suffice to require the flag map f_F to be surjective, since in general G' will contain more vertices than G, and hence more flags. The resulting definition is unfortunately unintuitive.

Definition 2.5. Let $f : G \to G'$ be a morphism between two total graphs; we say that f is *flag surjective* if the two diagrams below commute laxly,

$$\begin{array}{ccc} V & \xrightarrow{f_V} & V' \\ {\scriptstyle s^{-1}}\downarrow & \geq & \downarrow{\scriptstyle s'^{-1}} \\ P(E) & \xrightarrow[P(f_E)]{} & P(E') \end{array} \qquad \begin{array}{ccc} V & \xrightarrow{f_V} & V' \\ {\scriptstyle t^{-1}}\downarrow & \geq & \downarrow{\scriptstyle t'^{-1}} \\ P(E) & \xrightarrow[P(f_E)]{} & P(E') \end{array} \tag{3}$$

where s^{-1} and t^{-1} are the preimage maps of s and t respectively, and P is the powerset functor.

If a flag surjective morphism f is defined on a vertex v, it will ensure that all edges attached to $v' = f_V(v)$ are in the image of f_E, thus no additional edges can be attached to v' in the process. An example of a morphism which is not flag surjective can be found in Figure 9 in Appendix B. We'll call a morphism which is both flag injective and flag surjective a *flag bijection*. This is quite a strong property; it's almost enough to make the vertex map injective, but not quite.

Lemma 2.6. *Let $f : G \to G'$ be a flag bijection, and suppose that $f_V(v_1) = f_V(v_2)$ and both are defined; then* $\deg v_1 = \deg v_2 = 0$.

Proof. Let $v' = f_V(v_1) = f_V(v_2)$; since f is flag injective, the set of flags at v' must contain (the image of) the disjoint union of the flags at v_1 and v_2; hence $\deg v' \geq \deg v_1 + \deg v_2$. Since (by (2)) f_E is defined on all the flags at v_1, flag surjectivity implies that $\deg v_1 \geq \deg v'$, and similarly for v_2. Hence $\deg v' = \deg v_1 = \deg v_2 = 0$. □

Lemma 2.7. *Let G and G' be total graphs, and let $f : G \to G'$ be a flag bijection. For all $v \in V$, if $f_V(v)$ is defined, then f_E defines a bijection between the flags at v and those incident at $f_V(v)$; in consequence* $\deg v = \deg f_V(v)$.

Proof. Let $v' = f_V(v)$. The edges incident at v are given by the disjoint union of $s^{-1}(v)$ and $t^{-1}(v)$, and likewise at v'. Since f is flag injective, f_E is injective on the subset of flags defined by v. Since f is flag surjective all the flags at v' are in the image of $f_E(s^{-1}(v)) + f_E(t^{-1}(v))$. Note that since $f_V(v)$ is defined then f_E is defined for all $e \in s^{-1}(v)$ and all $e \in t^{-1}(v)$ by Eq. (2). Hence we have a bijection as required. □

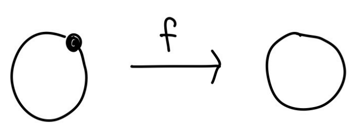
Lemma 2.8. *Let $f : G \to H$ and $g : H \to J$ be flag bijections; then $g \circ f$ is a flag bijection.*

Proof. See Appendix A. □

By the preceding lemma, and by observing that the identity is a flag bijection, we may conclude that the flag bijections define a wide subcategory of $[\bullet \rightrightarrows \bullet, \mathbf{Pfn}]_{\leq}$, which we will call $\mathcal{B}$.

Example 2.3 suggests a confounding special case: the vertex of a self loop can be forgotten. Here is another one:

Example 2.9. Let G be the (unique) total graph with one vertex and one edge; let G' be the (unique) partial graph with no vertices and a single edge. Define $f : G \to G'$ by $f_V = \emptyset$ and $f_E = \mathrm{id}_{\mathbf{1}}$. This is a valid flag bijection in $\mathcal{B}$.



While it is tempting to restrict to the subcategory defined by the total graphs, and ban such monsters by fiat, they do occur quite naturally in the rewrite theory we propose, albeit in quite restricted circumstances. So they must be tamed. To do so, we extend the definition of graph with *circles*: closed edges which have neither a source nor a target vertex[1]. Unfortunately the definition of graph morphism will get more complex and the resulting category is no longer a functor category, as we shall now see.

[1]This notion of graph has a long history; see, for example, the work of Kelly and Laplaza on compact closed categories [13].

Definition 2.10. A *graph with circles* is a 5-tuple $G = (V, E, O, s, t)$ where (V, E, s, t) is a total graph and O is a set of *circles*. For notational convenience we define the set of *arcs* as the disjoint union $A = E + O$.

A morphism $f : G \to G'$ between two graphs with circles consists of two (partial) functions $f_V : V \rightharpoonup V'$ as above, and $f_A : A \to A'$, satisfying the conditions listed below. Note that any such f_A factors as four maps,

$$\begin{array}{cc} f_E : E \to E' & f_{EO} : E \to O' \\ f_{OE} : O \to E' & f_O : O \to O' \end{array}$$

The following conditions must be satisfied:

1. $f_A : A \to A'$ is total;
2. the component $f_{OE} : O \to E'$ is the empty function;
3. the pair (f_V, f_E) forms a flag surjection between the underlying graphs in $\mathcal{B}$.

If, additionally, the following three conditions are satisfied, we call the morphism an *embedding*:

4. $f_V : V \rightharpoonup V'$ is injective;
5. the component f_O is injective;
6. the pair (f_V, f_E) forms a flag bijection between the underlying graphs.

It's worth noticing that if some f_A maps an edge e to a circle, then $f_E(e)$ is undefined, but $f_{EO}(e)$ is defined. This, by the lax naturality property, implies that f_V is undefined on both $s(e)$ and $t(e)$. Various examples and non-examples of morphisms and embeddings of graphs with circles can be found in Appendix B.

Lemma 2.11. *Defining composition point-wise, the composite of two morphisms of graphs with circles is again such a morphism. Additionally, if both morphisms are embeddings, their composition is an embedding as well.*

Proof. See Appendix A. □

We finally have introduced all the necessary structure to define our suitable category of graphs.

Definition 2.12. Let $\mathcal{G}$ be the category whose objects are graphs with circles, and whose arrows are morphisms as per Definition 2.10.

There is an obvious and close relationship between $\mathcal{G}$ and the category of partial graphs and flag bijections, $\mathcal{B}$. We can make this precise.

Definition 2.13. We define a forgetful functor $U : \mathcal{G} \to \mathcal{B}$ by

$$\begin{array}{ccc} U : (V, E, O, s, t) & \longmapsto & (V, E, s, t) \\ {\scriptstyle U:(f_V, f_A)} \downarrow & \mapsto & \downarrow {\scriptstyle (f_V, f_E)} \\ U : (V', E', O', s', t') & \longmapsto & (V', E', s', t') \end{array}$$

Example 2.14. Returning to Example 2.9, we see how this degenerate case fits in to the framework. We start with G, the unique total graph with a single vertex and a single edge (and no circles). There a single valid way to erase the vertex in $\mathcal{G}$.

Firstly observe that $G' = (\emptyset, \{e\}, \emptyset, \emptyset, \emptyset)$ as in the earlier example is not an object in $\mathcal{G}$. However $G'' = (\emptyset, \emptyset, \{e\}, \emptyset, \emptyset)$ is a valid graph, and the map $f : G \to G''$ which is undefined on the vertex and sends the edge to the circle is a valid morphism, indeed the only one.

Finally observe that the image of UG'' is the empty graph and Uf is the empty function.

The term “graph with circles” is unacceptably cumbersome, so henceforth we will simply say “graph” and refer to $\mathcal{G}$ as the category of graphs. In practice the circles are rarely important, although we will devote a disappointingly large amount of this paper to them.

3 DPO Rewriting in the Suitable Category

Double pushout rewriting [9] is an approach to formalising equational theories over graphs by rewriting. Each equation is formalised as a rewrite rule $L \Rightarrow R$, and the substitution $G[R/L]$ is computed via a double pushout as shown below.

$$\begin{array}{ccccc} L & \xleftarrow{l} & B & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & \lrcorner & \downarrow & \llcorner & \downarrow \\ G & \longleftarrow & C & \longrightarrow & H \end{array}$$

The upper span embeds a *boundary graph* B into both L and R; ensures that both graphs have the same connectivity, and hence that R can validly replace L. The map $m : L \to G$ is the *match*, an embedding of L into G, which shows where the rewrite will occur. The first pushout square is completed by C, the *context graph*; it is basically G with L removed. In the DPO approach, C is computed as a *pushout complement*. Finally the graph $H = G[R/L]$ is the graph resulting from performing the rewrite $L \Rightarrow R$ in G; it is computed as a pushout.

In the algebraic graph literature the notion of adhesive category [15, 16] is commonly used, as DPO rewriting behaves well in such categories. However, adhesivity is not suitable for our purposes, since the monomorphisms of $\mathcal{G}$ don’t play any special role in our formalism. We will instead consider a specific case of maps in the DPO diagram only, and in that context show the existence of pushouts and the existence and uniqueness of pushout complements, which are similar properties to those of an adhesive categories. The key to this approach is to recognise that B and C are in some sense partial graphs, as to a lesser extent are L and R; our handling of this partiality is one of the main novelties of this paper.

Notation 3.1. Almost every map in this section is an embedding of a small object into a larger one. Wherever unambiguous to do so, we will treat these embeddings as actual inclusions so, for example, we may write $m_E(e) = e$ despite the fact that the domain and codomain of the map are different graphs.

In our approach the graphs L and R that make up a rewrite rule have an additional distinguished vertex, the *boundary vertex* ∂, which represents the rest of the world, from the perspective of L (or R). The incident edges at ∂ represent the interface between L and the rest of the graph it occurs in. The context graph C also has a distinguished vertex, the dual boundary $\bar{\partial}$ which represents its interface. In our formalism, the graph B in the middle exists only to say that these interfaces must be compatible.

Definition 3.2. A *boundary graph* is a graph with exactly two vertices, ∂ and $\bar{\partial}$ (called respectively the *boundary* and *dual boundary* vertices), where $s(e) \neq t(e)$ for all its edges e, and there are no circles.

Definition 3.3. A *partitioning span* is a span $L \xleftarrow{l} B \xrightarrow{c} C$ in $\mathcal{G}$, where B is a boundary graph, the vertex component l_V is defined on ∂ and undefined on $\bar{\partial}$ and, dually, c_V is undefined on ∂ and defined on $\bar{\partial}$. Further, we require l and c to be embeddings.

An example of a partitioning span and its pushout in $\mathcal{G}$ is depicted in Figure 2. The name *partitioning span* arises from the fact that each of the maps out of the boundary graph replaces one half of it. Hence each graph has two regions, connected via the edges present in the boundary graph.

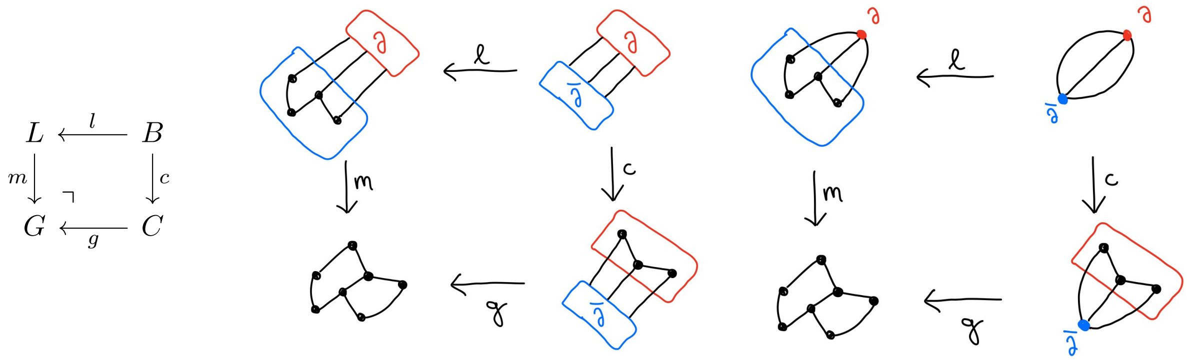


Figure 2: Pushout of the partitioning span from Figure 1, drawn in two different ways

Lemma 3.4. *Let $L \xleftarrow{l} B \xrightarrow{c} C$ be a partitioning span and suppose that $e = l_E(e_1) = l_E(e_2)$ in L for distinct e_1 and e_2 in E_B. Then e is a self-loop at ∂ in L and for all other $e_3 \neq e_1 \neq e_2$ we have $e \neq l_E(e_3)$. The same holds mutatis mutandis for C.*

Proof. By flag bijectivity all the flags at ∂ must be preserved, including distinct flags for $l_E(e_1)$ and $l_E(e_2)$. By hypothesis these two edges are identified so necessarily $s_B(e_1) = \partial$ and $s_B(e_2) = \bar{\partial}$ or vice versa. Hence e is a self loop. Suppose further that $l_E(e_3) = e$; then l is not flag bijective, which is a contradiction. □

Self-loops in partitioning spans indicate that the boundary is connected back to itself without an intervening vertex. This is responsible for the failure of injectivity on edges and gives rise to degeneracy when constructing pushouts. We can study them using a dual perspective.

Definition 3.5. The *pairing graph* for a partitioning span $L \xleftarrow{l} B \xrightarrow{c} C$ is a labelled directed graph whose vertices are E_B; each vertex receives a *polarity*: $+$ if $s_B(e) = \partial$, $-$ if $s_B(e) = \bar{\partial}$. We draw a *blue* edge between e_1 and e_2 if $l_E(e_1) = l_E(e_2)$ i.e. if e_1 and e_2 form self-loop in L; similarly we draw a *red* edge between e_1 and e_2 if they form a self-loop in C. Blue edges are directed from positive to negative polarity; red edges the reverse.

An example of a pairing graph is shown in Figure 3. The pairing graph is always bipartite: it's immediate from the definition that vertices of the same polarity are never connected. Further, due to Lemma 3.4, each vertex can have a maximum of one edge of each colour incident to it. In consequence every connected component is just a path, possibly of length zero, possibly a cycle. From these properties, we have the following immediate corollary.

Corollary 3.6. *Let $\mathcal{P}$ be the pairing graph of the partitioning span $L \xleftarrow{l} B \xrightarrow{c} C$; then each connected component p of $\mathcal{P}$ determines an edge-disjoint path on B. For those components which* *are not cycles, if the first vertex of p is positive, then the path starts at ∂; if negative the path starts at $\bar{\partial}$. Conversely, if the last vertex of p is positive, the path ends at $\bar{\partial}$ and vice versa.*

The reader may already suspect that when we form the pushout of a partitioning span, the components of the pairing graph determine which edges in B will be identified. This is indeed the case; it forms an intermediate result (Lemma A.3) in the proof of the next theorem.

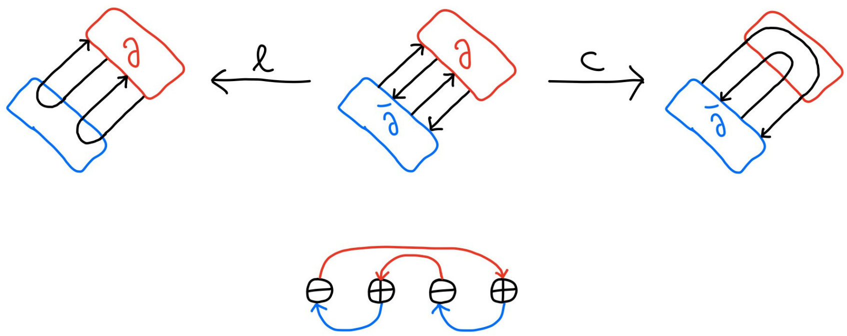


Figure 3: Example of a partitioning span with its pairing graph

Theorem 3.7. *In $\mathcal{G}$, pushouts of partitioning spans exist. Further, the maps into the pushout are embeddings.*

Proof. See Appendix A. □

Since pushouts of partitioning spans are the basis of the rewrite theory we wish to pursue, for the rest of the paper the term "pushout" should be understood to imply "of partitioning span".

We now move on to the other required ingredient for DPO rewriting: pushout complements. Just as we did with partitioning spans and pushouts, we will introduce a specific kind of embedding for which the complement must exist.

Definition 3.8. A *boundary embedding* is a pair of maps $B \xrightarrow{l} L \xrightarrow{m} G$ in $\mathcal{G}$, where B is a boundary graph, where : (i) $l_V(\partial)$ is defined but $l_V(\bar{\partial})$ is undefined; and (ii) $(m_V \circ l_V)(\partial)$ is undefined. Further, L has to be a connected graph, and m an embedding.

Definition 3.9. Given a boundary embedding $B \xrightarrow{l} L \xrightarrow{m} G$ we can immediately construct half a pairing graph $\mathcal{P}$, consisting of only the blue edges using the mapping $l : B \to L$. The *re-pairing problem* is to construct the other half (the red edges) so that the connected components map to the edges of G (cf. Lemma A.3). See Figure 5 for examples.

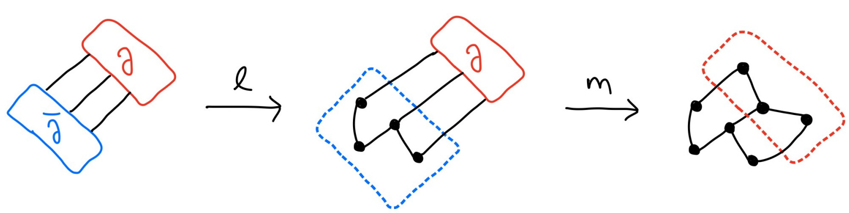


Figure 4: Example of a boundary embedding

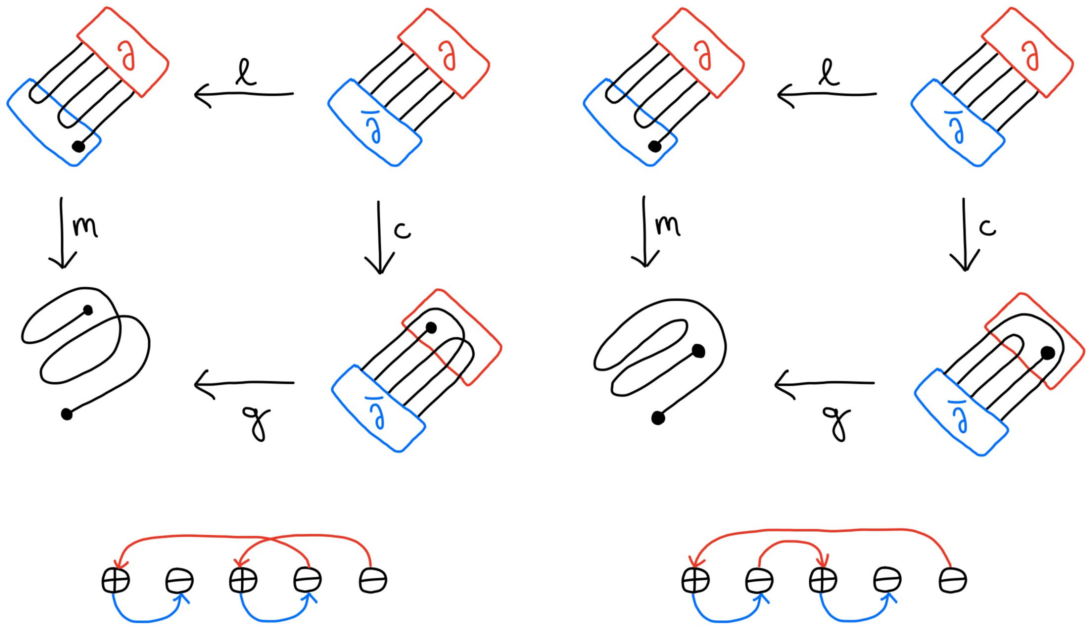


Figure 5: Two different solutions to the same re-pairing problem, together with the corresponding pairing graphs.

Lemma 3.10. *Given a boundary embedding $B \xrightarrow{l} L \xrightarrow{m} G$ a solution to the re-pairing problem always exists, but it is not necessarily unique.*

Proof. See Appendix A. □

Theorem 3.11. *In $\mathcal{G}$, pushout complements of boundary embeddings exist, and give rise to partitioning spans.*

Proof. We'll use the boundary embedding $B \xrightarrow{l} L \xrightarrow{m} G$ to construct the complement C such that $L \xleftarrow{l} B \xrightarrow{c} C$ is a partitioning span, and show that G is indeed the pushout of this span.

Let C have vertex set $V_C = (V_G \setminus V_L) + \{\bar{\partial}\}$. We'll construct the edge set, and the source and target maps, in three steps.

1. Let E_C contain all the edges of the induced subgraph of G defined by the vertices V_C, and define the source and target maps on those edges correspondingly.
2. Let O_C contain $O_G \setminus m_O^{-1}(O_G)$.
3. Finally we add the edges between $\bar{\partial}$ and the rest of the graph, and simultaneously define the map $c : B \to C$. Let $\mathcal{P}$ be a solution to the re-pairing problem given by $B \xrightarrow{l} L \xrightarrow{m} G$. If in $\mathcal{P}$ there is a red edge between e_1 and e_2 in create a self-loop e at $\bar{\partial}$ and set $c(e_1) = c(e_2) = e$. If there is any vertex e in $\mathcal{P}$ which has no incident red edge, add e to E_C; if its polarity is positive set
$$s_C(e) = (s_G \circ m_E \circ l_E)(e) \qquad t_C(e) = \bar{\partial}$$
and if the polarity is negative, the source and target are reversed. We define $c_E(e) = e$.

The resulting span $L \xleftarrow{l} B \xrightarrow{c} C$ is evidently partitioning, and by construction has G as its pushout, as a consequence of Lemma A.3 □

Theorem 3.12. *In $\mathcal{G}$, pushout complements are unique up to the solution of the re-pairing problem.*

Proof. Suppose that both $B \xrightarrow{c} C \xrightarrow{g} G$ and $B \xrightarrow{c'} C' \xrightarrow{g'} G$ are pushout complements for the boundary embedding $B \xrightarrow{l} L \xrightarrow{m} G$. Observe that given the boundary embedding, a solution to the re-pairing problem determines the map $c : B \to C$ and vice versa. Let's assume for now a that $\mathsf{im}(c) = \mathsf{im}(c')$ and hence they both correspond to the same pairing graph.

Since m is an embedding, it follows that every part of C not in $\mathsf{im}(c)$ is preserved isomorphically in G, and similarly for C'. Since we have assumed $\mathsf{im}(c) = \mathsf{im}(c')$ this implies that $C \simeq C'$.

Further, observe that different solution of the re-pairing also have the same number of edges, and hence produce the same number of self loops at $\bar{\partial}$. Hence the difference between different solutions is just the labels on the edges incident at $\bar{\partial}$. □

4 A Category of Rotation Systems

Despite some suggestive illustrations, up to this point we have operated in a purely combinatorial setting, but now we introduce some topological information in the form of rotation systems. A rotation system for a graph determines an embedding of the graph into a surface by fixing a cyclic order of the incident edges, or more precisely the flags, at every vertex.

We augment our category of graphs with this extra structure, in the form of *cyclic lists of flags* for each vertex, and strengthen the property of flag surjectivity (Equation 3). The requisite categorical properties for DPO rewriting will follow more or less immediately from those of the underlying category of directed graphs.

Definition 4.1. Let $\mathsf{CList} : \mathbf{Set} \to \mathbf{Set}$ be the functor where $\mathsf{CList}\, X$ is the set of circular lists whose elements are drawn from X.

Definition 4.2. A *rotation system* R for a graph with circles (V, E, O, s, t) is a total function $\mathsf{inc} : V \to \mathsf{CList}\, E$ such that :

- $e \in \mathsf{inc}(s(e))$
- $e \in \mathsf{inc}(t(e))$
- $t^{-1}(v) + s^{-1}(v) \cong \mathsf{inc}(v)$ (when considering $\mathsf{inc}(v)$ as a set)

We call $\mathsf{inc}(v)$ the *rotation* at v.

Note that $\mathsf{inc}(v)$ is actually a cyclic ordering on the set of flags at v.

Definition 4.3. A homomorphism of rotation systems $f : R \to R'$ is a $\mathcal{G}$-morphism (f_A, f_V) between the underlying graphs, satisfying the following additional condition.

$$\begin{array}{ccc} V & \xrightarrow{f_V} & V' \\ {\scriptstyle \mathsf{inc}}\downarrow & \geq & \downarrow{\scriptstyle \mathsf{inc}'} \\ \mathsf{CList}\, E & \xrightarrow[\mathsf{CList}\, f_E]{} & \mathsf{CList}\, E' \end{array} \qquad (4)$$

This condition requires the preservation of the edges ordering on vertices where f_V is defined; it implies flag surjectivity (Equation 3). Morphisms therefore either preserve a vertex with its rotation exactly, or forget about it.

Definition 4.4. Let $\mathcal{R}$ be the category whose objects are tuples (V,E,O,s,t,inc) where (V,E,O,s,t) is an object of the category of graphs, $\mathcal{G}$ (see Def. 2.10) and inc is a rotation system for this graph. The morphisms of $\mathcal{R}$ are homomorphisms of rotation systems.

There is an evident forgetful functor $U' : \mathcal{R} \to \mathcal{G}$; this is especially clean since the morphisms of $\mathcal{R}$ are just $\mathcal{G}$-morphisms which satisfy an additional condition. Further, since we demand require the inc structure to be preserved exactly, pushouts and complements are very easily defined here.

Definition 4.5. In $\mathcal{R}$, objects B, spans $L \xleftarrow{l} B \xrightarrow{c} C$ and composites $B \xrightarrow{l} L \xrightarrow{m} G$ are respectively *boundary graphs*, *partitioning spans*, and *boundary embeddings* if their underlying graphs in $\mathcal{G}$ satisfy those definitions (respectively Definitions 3.2, 3.3, and 3.8).

Lemma 4.6. *In $\mathcal{R}$ pushouts of partitioning spans exist.*

Proof. The pushout candidate is the one in the underlying category (see Theorem 3.7), together with the rotation system:

$$\mathsf{inc}_G(v) = \begin{cases} \mathsf{inc}_C(v), & \text{if } v \in V_C \\ \mathsf{inc}_L(v), & \text{if } v \in V_L \end{cases}$$

The vertex set of the pushout is the disjoint union of vertices from both input graphs, $V_G = (V_L + V_C) \setminus V_B$. Therefore, by the mediating map from Theorem 3.7, inc_G is indeed the pushout of the rotations. □

Lemma 4.7. *In $\mathcal{R}$ pushout complements of boundary embeddings exist, and are unique up to the solution of the re-pairing problem.*

Proof. This follows from the underlying construction in $\mathcal{G}$; see Theorem 3.12. Note that the rotation for every vertex of C is determined by either those of G or of B, so there is no choice about the additional structure. □

Remark 4.8. We must sound a cautionary note about the "up to" in the preceding statement. While in $\mathcal{G}$ pushout complements that arise from different pairing graphs are essentially the same, this is not so in $\mathcal{R}$. Since the rotation around $\bar{\partial}$ is preserved exactly by $c : B \to C$, different choices for which edges to merge as self loops will result in different local topology at $\bar{\partial}$. In particular it can happen that a re-pairing problem can have planar and non-planar solutions; see Figure 5 for an example.

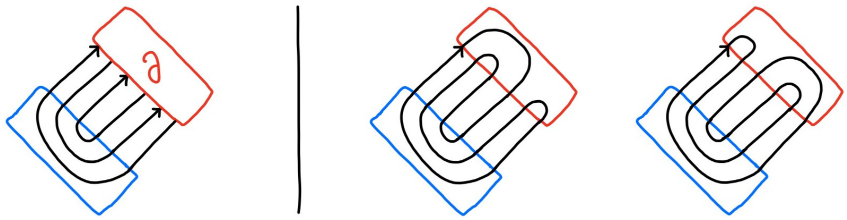
With that caveat noted, since $\mathcal{R}$ has pushouts and their complements, specialised to the setting where the rewrite rules explicitly encode the connectivity at their boundary, we can use it as a setting for DPO rewriting of surface embedded graphs.

Remark 4.9. As illustrated in Figure 5, we have adopted a particular convention for drawing the pairing graphs: the vertices are drawn in a row, with the red edges above and the blue edges below. If the vertices are drawn in an order compatible with $\mathsf{inc}_B(\partial)$ then the blue edges (partly) reproduce the local topology at ∂ in L. Any edge crossings imply the region around ∂ is not planar. This is sufficient but not necessary for L to be non-planar. Isomorphic statements can be made for $\bar{\partial}$ in C.

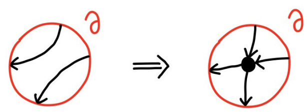
5 On Planarity

Since the graphs of $\mathcal{R}$ are equipped with rotation systems they carry information about their topology along with them. As the previous section showed, $\mathcal{R}$ admits DPO rewriting, but we might ask for more, for example, to maintain a topological invariant. Concretely, we might ask: if L, R, and G are all plane embedded is $G[R/L]$ also plane? We have already seen, in Remark 4.8 above, that the re-pairing problem can have topologically distinct solutions.

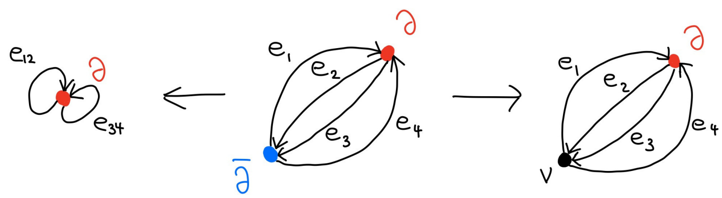
Focussing on plane graphs, it's possible that the re-pairing problem has distinct plane solutions, for example :



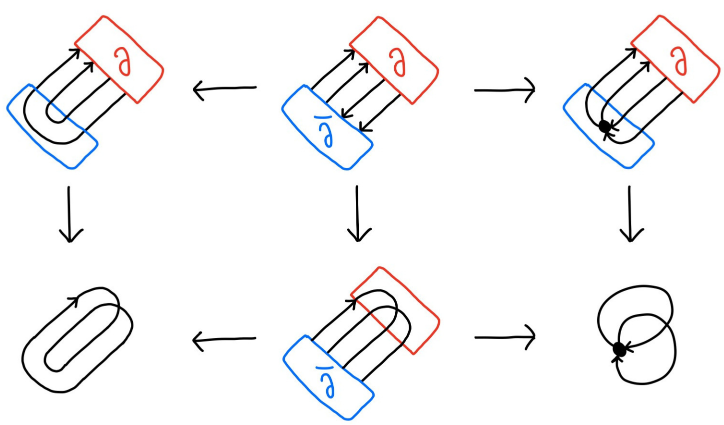
It may also occur that there is no plane solution. Consider a rewrite rule:



This is a legitimate rewrite rule for plane graphs, and expanding it into a span for the top of a double pushout diagram makes sense.

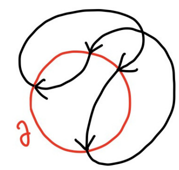


Further it's clear that the left-hand side can be embedded into a circle, which is trivially plane. However when we apply the rewrite something goes wrong.



In this example we match the left hand side of the rule to the graph with one circle and no vertices, and compute the complement and the result of substituting the right hand side into the context graph. When computing the pushout complement of this boundary embedding, we notice that there is only one solution to the re-pairing problem, and that this solution is not plane. In a setting where all embeddings are plane this is an unwanted case.

However in this case, solving the re-pairing problem and computing the pushout complement already alerts us to the problem, since this graph is not plane. Notice first that the graph G not carrying a rotation system itself. The circle is seemingly plane, but with the flags in L fixed, there is no way this circle can be drawn on the plane without edges crossing.



Secondly, observe that the right hand side of the rewrite plays no role: all the toplogical information is in the boundary embedding. Thus we have a checkable condition to detect when a rewrite will fail to preserve the surface. We might hope for a necessary and sufficient condition, or a stronger result allowing us to compute how matching a given boundary into a graph alters the surface it is embedded in.

6 Conclusions and Further Work

In this work we have made some significant progress towards a purely combinatorial formalisation of surface embedded strings diagrams. Along the way we have introduced a new representation for symmetric string diagrams and PROPs which removes several annoyances of earlier approaches.

An obvious next step, already underway, is to formalise string diagrams using the graph representation described here. Unlike the situation we have discussed in this paper, a morphism in a monoidal category is not a closed surface – it has a boundary, and it has wires which impinge on that boundary. Fortunately, the technology of boundary vertices developed in Section 3 can be easily adapted for this purpose. At this point one could generalise to the situation of a diagram on a surface with multiple boundaries.

However to build a complete theory of diagrams on surfaces we must address two major topological questions. The first was already described in Section 5: the preservation of planarity by rewrites. The second was briefly mentioned in the introduction: disconnected graphs. Minimally we must record the relationships between components and faces of other components, and consider how these relationships change under rewrites. Many other details arise, such as the orientation of circles.

A much simpler modification to the theory would be to consider the undirected case. This is relatively easy, since undirected graphs can be obtained by a forgetful functor from the directed ones. However some details also change. For example the repairing problem has more solutions in the undirected setting than the directed. However we expect no major difficulties here.

Finally, a computerised implementation of this representation would be most helpful for experiments and applications.

Acknowledgements We would like to thank Tim Ophelders for helpful remarks on the different solutions of the re-pairing problem, and the anonymous reviewers for their comments.

References

[1] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski & Fabio Zanasi (2016): *Rewriting modulo symmetric monoidal structure*. In: *LiCS'16 Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 707–719, doi:10.1145/2933575.2935316.

[2] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski & Fabio Zanasi (2017): *Confluence of Graph Rewriting with Interfaces*. In: *Proceedings of the 26th European Symposium on Programming Languages and Systems (ESOP'17)*, doi:10.1007/978-3-662-54434-1_6.

[3] Bob Coecke, Ross Duncan, Aleks Kissinger & Quanlong Wang (2015): *Generalised Compositional Theories and Diagrammatic Reasoning*. In G. Chirabella & R. Spekkens, editors: *Quantum Theory: Informational Foundations and Foils*, Springer, doi:10.1007/978-94-017-7303-4_10.

[4] Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons & Seyon Sivarajah (2019): *On the qubit routing problem*. In Wim van Dam & Laura Mancinska, editors: *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 135, pp. 5:1–5:32, doi:10.4230/LIPIcs.TQC.2019.5. Available at `http://drops.dagstuhl.de/opus/volltexte/2019/10397`.

[5] Lucas Dixon & Ross Duncan (2009): *Graphical Reasoning in Compact Closed Categories for Quantum Computation*. *Annals of Mathematics and Artificial Intelligence* 56(1), pp. 23–42, doi:10.1007/s10472-009-9141-x.

[6] Lucas Dixon, Ross Duncan & Aleks Kissinger (2010): *Open Graphs and Computational Reasoning*. In: *Proceedings DCM 2010*, *Electronic Proceedings in Theoretical Computer Science* 26, pp. 169–180, doi:10.4204/EPTCS.26.16.

[7] Lucas Dixon & Aleks Kissinger (2013): *Open Graphs and Monoidal Theories*. *Math. Structures in Comp. Sci.* 23(2), pp. 308–359, doi:10.1017/S0960129512000138.

[8] J. R. Edmonds (1960): *A combinatorial representation for polyhedral surfaces*. *Notices of the American Mathematical Society* 7(A646).

[9] Hartmut Ehrig, Karsten Ehrig, Ulrike Prange & Gabriele Taentzer (2006): *Fundamentals of Algebraic Graph Transformation*. Monographs in Theoretical Computer Science, Springer Berlin Heidelberg, doi:10.1007/3-540-31188-2.

[10] Jonathan L. Gross & Thomas W. Tucker (2001): *Topological Graph Theory*. Dover.

[11] Lothar Heffter (1891): *Über das Problem der Nachbargebiete*. *Mathematische Annalen* 38(4), pp. 477–508, doi:10.1007/BF01203357.

[12] A. Joyal & R. Street (1991): *The Geometry of Tensor Categories I*. *Advances in Mathematics* 88, pp. 55–113. doi:10.1016/0001-8708(91)90003-P.

[13] G.M. Kelly & M.L. Laplaza (1980): *Coherence for Compact Closed Categories*. *Journal of Pure and Applied Algebra* 19, pp. 193–213, doi:10.1016/0022-4049(80)90101-2.

[14] Aleks Kissinger & Vladimir Zamdzhiev (2015): *Equational Reasoning with Context-Free Families of String Diagrams*. In Francesco Parisi-Presicce & Bernhard Westfechtel, editors: *Graph Transformation*, *Lecture Notes in Computer Science* 9151, Springer International Publishing, pp. 138–154, doi:10.1007/978-3-319-21145-9_9.

[15] S. Lack & P. Sobocinski (2003): *Adhesive categories*. Technical Report BRICS RS-03-31, BRICS, Department of Computer Science, University of Aarhus.

[16] Stephen Lack & Pawel Sobocinski (2005): *Adhesive and quasiadhesive categories*. *Theoretical Informatics and Applications* 39(3), pp. 511 – 545, doi:10.1051/ita:2005028.

[17] Peter Selinger (2011): *A survey of graphical languages for monoidal categories*. In Bob Coecke, editor: *New structures for physics*, *Lecture Notes in Physics* 813, Springer, pp. 289–355, doi:10.1007/978-3-642-12821-9_4.

A Proofs

From Section 2

Lemma 2.8 Let $f : G \to H$ and $g : H \to J$ be flag bijections; then $g \circ f$ is a flag bijection.

Proof. For flag injectivity, we assume injectivity of the flag maps induced by f and g. If f_V is undefined, so is the flag map. Consider flags (e,v) and (e',v') where $(f_E \times f_V)$ is defined, $v = s(e)$, $v' = s(e')$, and assume $g_F(f_F(e,v)) = g_F(f_F(e',v'))$. Because f is a flag surjection and defined on the given flags, Equation 3 holds strictly on v and v'. Therefore we get: $f_E(e) = s_H(f_V(v))$ and $f_E(e') = s_H(f_V(v'))$. This lets us apply flag injectivity of g to get $f_F(e,v) = f_F(e',v')$, and flag injectivity of f to reach $(e,v) = (e',v')$. The argument applies equally to the target map.

For flag surjectivity we assume lax commutation of Equation 3 for f and g and show that the composite diagram also commutes laxly.

$$
\begin{array}{ccccc}
V_G & \xrightarrow{f_V} & V_H & \xrightarrow{g_V} & V_J \\
{\scriptstyle s_G^{-1}}\downarrow & \geq & {\scriptstyle s_H^{-1}}\downarrow & \geq & \downarrow{\scriptstyle s_J^{-1}} \\
P(E_G) & \xrightarrow[P(f_E)]{} & P(E_H) & \xrightarrow[P(g_E)]{} & P(E_J)
\end{array}
\qquad
\begin{array}{ccccc}
V_G & \xrightarrow{f_V} & V_H & \xrightarrow{g_V} & V_J \\
{\scriptstyle s_G^{-1}}\downarrow & & \geq & & \downarrow{\scriptstyle s_J^{-1}} \\
P(E_G) & \xrightarrow[P(f_E)]{} & P(E_H) & \xrightarrow[P(g_E)]{} & P(E_J)
\end{array}
$$

In the case of either f_V or g_V being undefined, the composite $(g_V \circ f_V)$ is also undefined and the diagram commutes laxly immediately. If both f_V and g_V are defined, both their diagrams commute strictly, and by diagram gluing, their composite does as well. □

Lemma 2.11 Defining composition point-wise, the composite of two morphisms of graphs with circles is again such a morphism. Additionally, if both morphisms are embeddings, their composition is an embedding as well.

Proof. Let $f : G \to G'$ and $g : G' \to G''$ be two morphisms; then $g \circ f = ((g_{V'} \circ f_V),(g_{A'} \circ f_A))$; since composition of partial functions is associative, we need only check that the four properties of Definition 2.10 are preserved.

Conditions 1 and 4 follow from the properties of partial functions, and condition 6 (which includes condition 3) follows from Lemma 2.8. Observe that

$$
\begin{aligned}
(g \circ f)_O &= [g_{EO}, g_O] \circ (f_{OE} + f_O) \\
&= (g_{EO} \circ f_{OE}) + (g_O \circ f_O) \\
&= (g_{EO} \circ \emptyset) + (g_O \circ f_O) \\
&= g_O \circ f_O
\end{aligned}
$$

hence $(g \circ f)_O$ is injective since f_O and g_O are, satisfying condition 5. Finally, by a similar argument we have

$$
\begin{aligned}
(g \circ f)_{EO} &= [g_{EO}, g_O] \circ (f_E + f_{EO}) \\
&= (g_{EO} \circ f_E) + (g_O \circ f_{EO}) \\
&= (\emptyset \circ f_E) + (g_O \circ \emptyset) \\
&= \emptyset
\end{aligned}
$$

hence the remaining condition 2 is satisfied. □

From Section 3

Theorem 3.7 In $\mathcal{G}$, pushouts of partitioning spans exist. Further, the maps into the pushout are embeddings.

Proof. The proof will proceed via several intermediate results. First we will explicitly define the pushout candidate $L \underset{m}{\rightarrow} G \underset{g}{\leftarrow} C$ (Definition A.1), show the constructed object G is a valid graph, (Lemmas A.2 and A.4), show that m and g are indeed embeddings in $\mathcal{G}$ (Lemma A.5), and finally show that the required universal property holds in in $\mathcal{G}$ (Lemma A.6). This suffices to prove the theorem. □

Definition A.1. Given the partitioning span $L \overset{l}{\leftarrow} B \overset{c}{\rightarrow} C$, we define the *pushout candidate* $L \underset{m}{\rightarrow} G \underset{c}{\leftarrow} C$ as follows.

We construct the underlying sets and functions by pushout in **Pfn**,

$$\begin{array}{ccc} V_L & \overset{l_V}{\longleftarrow} & \{\partial, \bar{\partial}\} \\ {\scriptstyle m_V}\downarrow & \lrcorner & \downarrow{\scriptstyle c_V} \\ V_G & \underset{g_V}{\longleftarrow} & V_C \end{array} \qquad \begin{array}{ccc} A_L & \overset{l_A}{\longleftarrow} & E_B \\ {\scriptstyle m_A}\downarrow & \lrcorner & \downarrow{\scriptstyle c_A} \\ A_G & \underset{g_A}{\longleftarrow} & A_C \end{array} \tag{5}$$

so explicitly we have

$$V_G = (V_C + V_L) \setminus V_B \qquad A_G = (A_L + A_C)/{\sim}$$

where $\sim$ is the least equivalence relation such that $l_E(e) = c_E(e)$ for $e \in E_B$. Next we define the source map by

$$s_G(e') = \begin{cases} s_L(e), \text{ if } e' = m_A(e) \text{ and } s_L(e) \text{ is defined and } s_L(e) \neq \partial \\ s_C(e), \text{ if } e' = g_A(e) \text{ and } s_C(e) \text{ is defined and } s_C(e) \neq \bar{\partial} \\ \text{undefined otherwise.} \end{cases} \tag{6}$$

for all $e' \in A_G$. The target map t_G is defined similarly. (Strictly speaking we have defined s and t on all of A; they will be restricted to E when we have defined that.) Finally we divide the arcs into edges and circles by setting

$$E_G = \{e \in A_G : \text{ both } s_G(e) \text{ and } t_G(e) \text{ are defined } \} \tag{7}$$

$$O_G = A_G \setminus E_G \tag{8}$$

There are two properties that need to be checked to ensure that the definition above yields a valid graph. The source and target maps should be well-defined partial functions; and all arcs should either have two end points (i.e. they are edges) or none (they are circles).

Lemma A.2. *Equation* (6) *defines a partial function: if* $s_G(e')$ *is defined, it is single-valued.*

Proof. There are two things to check. First we show that if the first or second clause of the definition applies it is single valued. We then show that at most one of those clauses can apply.

Suppose that in L we have distinct e_1, e_2 such that $m_A(e_1) = m_A(e_2)$ and $s_L(e_1) \neq \partial$. Since they are distinct in L and identified in G, we must have distinct $e_1, e_2 \in B$ such that $c_A(e_1) = c_A(e_2)$ in C. By Lemma 3.4 this gives a self-loop at $\bar{\partial}$ in C, which in turn implies that $s_L(e_2) = \partial$. Hence

L provides at most one candidate source vertex for every edge in G, and a similar argument can be made for C.

Now suppose $m_A(e_1) = g_A(e_2)$, and that $s_L(e_1) \neq \partial$ and $s_C(e_2) \neq \bar{\partial}$. Since the edges are identified in G they are both present in B. Since $s_L(e_1) \neq \partial$ we have $s_B(e_1) = \bar{\partial}$, from which $s_C(e_1) = \bar{\partial}$. Since $s_C(e_2) \neq \bar{\partial}$, e_1 and e_2 are distinct in C. Therefore we must have e_1 and e_2 identified in L; therefore, by Lemma 3.4, e_1 must be a self-loop at ∂ which contradicts our original assumption. Therefore there is at most one candidate source vertex and the map s_G is well defined in (6). □

The preceding argument applies equally to the target map t_G.

Lemma A.3. *Let $\mathcal{P}$ be the pairing graph of the partitioning span $L \xleftarrow{l} B \xrightarrow{c} C$, and let G be its pushout candidate.*

1. *Suppose e and e' are edges in B; if e and e' are in the same component of $\mathcal{P}$ then $(m_A \circ l_E)(e) = (m_A \circ l_E)(e')$.*
2. *Let e be any arc in A_G; then its preimage in B is either empty or is exactly one connected component of $\mathcal{P}$.*

Proof. (1) Suppose that e and e' are the same component of $\mathcal{P}$. We use induction on the length of the path from e to e' in $\mathcal{P}$. If the path is length zero, then $e = e'$ and the property holds trivially. Otherwise, let e'' be the predecessor of e'. By induction, and (5), we have

$$(m_A \circ l_E)(e'') = (m_A \circ l_E)(e) = (g_A \circ c_E)(e) = (g_A \circ c_E)(e'')$$

Since e' and e'' are adjacent in $\mathcal{P}$ we must have either $l_E(e') = l_E(e'')$ or $c_E(e') = c_E(e'')$ depending on the colour of the edge. From this the result follows.

(2) Let $e \in A_G$ and suppose that $e_1 \in (m_A \circ l_E)^{-1}(e)$ in B. Either e_1 is a component on its own, or it has a neighbour e_2. By the definition of $\mathcal{P}$ either $l_E(e_1) = l_E(e_2)$ or $c_E(e_1) = c_E(e_2)$ depending on the colour of the edge. Therefore we have

$$(m_A \circ l_E)(e_2) = (m_A \circ l_E)(e_1) = e$$

so e_2 is also in the pre-image of e. By induction, the entire component containing e_1 must also be included in the pre-image.

For the converse, recall that $A_G = (A_L + A_C)/\sim$ where $\sim$ is the least equivalence relation such that $l_E(e_i) = c_E(e_i)$ for $e_i \in E_B$. Therefore if distinct e' and $e'' \in E_B$ both belong to the preimage of $e \in A_G$, there necessarily exists a chain of equalities

$$l_E(e') = l_E(e_1), \quad c_E(e_1) = c_E(e_2), \quad l_E(e_2) = l_E(e_3), \quad \ldots, \quad c_E(e_n) = c_E(e'')$$

to place them in the same equivalence class. Such a chain of equalities precisely defines a path from e' to e'' in $\mathcal{P}$, hence if two edges of B are identified in the pushout, they belong to the same component in the pairing graph. □

Lemma A.4. *Let G be the pushout candidate defined above. For all arcs $e \in A_G$ either both $s_G(e)$ and $t_G(e)$ are defined or neither is.*

Proof. Consider the preimage of e in B; if it is empty then e is simply included in G from either L or C, along with both its end points.

Otherwise, by Lemma A.3, e corresponds to a connected component p of the pairing graph $\mathcal{P}$. By corollary 3.6 such components can be either line graphs or closed loops. If p is a closed loop, for all $e_i \in p$ we have

$$s_L(l_E(e_i)) = t_L(l_E(e_i)) = \partial \qquad \text{and} \qquad s_C(c_E(e_i)) = t_C(c_E(e_i)) = \bar{\partial}$$

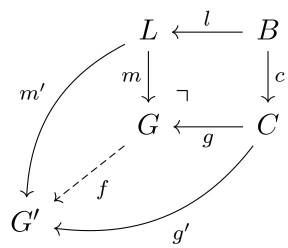
so, by (6), neither $s_G(e)$ nor $t_G(e)$ is defined. If, on the other hand, p forms a path $e_1, e_2, \ldots, e_n$, its ends provide the source and target. Specifically, if e_1 positive in $\mathcal{P}$ then $s_C(c_E(e_1)) \neq \bar{\partial}$ and if it is negative $s_L(l_E(e_1)) \neq \partial$; if e_n is positive $t_L(l_E(e_n)) \neq \partial$, and if e_n is negative $t_C(c_E(e_n)) \neq \bar{\partial}$.

Hence $s_G(e)$ is defined if and only if $t_G(e)$ is defined. Therefore the division of A_G into edges and circles is correct and G is indeed a valid graph. □

Lemma A.5. *The arrows of the cospan* $L \xrightarrow[m]{} G \xleftarrow[g]{} C$ *defined by the pushout candidate are embeddings in* $\mathcal{G}$.

Proof. We will show the result for m; the proof for g is the same. Note that Properties 4 and 1 are automatic from the underlying pushouts in **Pfn**. Since the graph B has no circles, the m_O component is injective by construction (Property 5) and since no arc gets a source or target in G unless its preimage had one, the component m_{OE} is empty as required (Property 2). Finally we have to show that the induced map (m_V, m_E) is a flag bijection. First note that if $m_E(e)$ is undefined then e is necessarily a self-loop at ∂, and $m_V(\partial)$ is always undefined, so the squares (2) commute. Otherwise if $(f_V \circ s_L)(e)$ is defined then the square commutes directly by the definition of s_G above, and similarly for t_G. Finally for all $v \neq \partial \in V_L$, we have that $m_V(v)$ is defined. By the definition of s_G and t_G, e is a flag at v if and only if $m_E(e)$ is a flag at $m_V(v)$. Flag injectivity and flag surjectivity follow immediately. Hence m is an embedding in $\mathcal{G}$. □

Lemma A.6. *the cospan* $L \xrightarrow{m} G \xleftarrow{g} C$ *has the required universal property.*



Proof. Since the underlying sets and functions are constructed via pushout the required mediating map $f = (f_V, f_A)$ exists; we need to show that it is a morphism of $\mathcal{G}$. Property 1 follows from m' and g' satisfying it as well. For the f_{OE} to be empty (Property 2), use the fact that m'_{OE} and g'_{OE} are empty for circles in L and C, because they are morphisms in $\mathcal{G}$. The remaining case for a circle to appear in G is as the pushout of some edges in B being identified in one instance of the re-pairing problem. In this case, because the outer square has to commute for the edge component, these edges have to be identified, and hence form a circle, in G', too. This makes f_{OE} empty. For flag surjectivity between the underlying graphs (Property 3), observe that the vertex set V_G is the disjoint union of vertex sets V_L and V_C. Because m' and g' are valid morphisms in $\mathcal{G}$, they are flag surjective, and therefore so is f. □

This completes the proof of Theorem 3.7.

Lemma 3.10 Given a boundary embedding $B \xrightarrow{l} L \xrightarrow{m} G$ a solution to the re-pairing problem always exists, but it is not necessarily unique.

Proof. Note that any half-pairing graph has connected components of at most two vertices, linked by a (blue) edge from a positive vertex to a negative one. Define the component of an arc by

$$k(a) = (m_A \circ l_A)^{-1}(a) \text{ for all } a \in A_G$$

Note that this defines a partition of the set $E_B \simeq \sum_{a \in A_G} k(a)$, and each (non-empty) $k(a)$ determines a connected component of the solution to the re-pairing problem. We'll abuse notation and use $k(a)$ to also denote the subgraph of the half-pairing graph whose vertices are $k(a)$. There are two cases depending whether a is a circle or an edge.

1. Suppose $a \in O_G$; we form a closed loop involving all $e \in k(a)$, by adding red edges as follows. Pick a degree-one positive vertex p follow the incident blue edge to the negative vertex n; now pick another a degree-one positive vertex p' which is not connected to n. Add a red edge from n to p'. Repeat the process starting from p'. When no more vertices remain, close the loop by adding a red edge from the final negative vertex back to p. Since a is a circle, $k(a)$ necessarily contains an even number of vertices, so closing the loop is always possible.

2. The case when a is an edge is slightly more complex because edges have end points; $k(a)$ may contain zero, one, or two degree-zero vertices depending how many of its end points are defined by vertices in L. We will connect the vertices as previously, but in a line, rather than a loop. Since we can only add red edges, and only one at each vertex, the degree-zero vertices will necessarily be the end points of this line.

□

B Examples

This collection captures some of the corner cases we do or do not want to allow as morphisms in the category of graphs with circles as described in Definition 2.10.

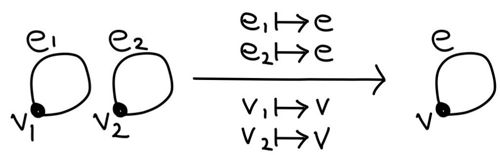


Figure 6: A flag surjective, but not flag bijective map. This is a valid of $\mathcal{G}$.

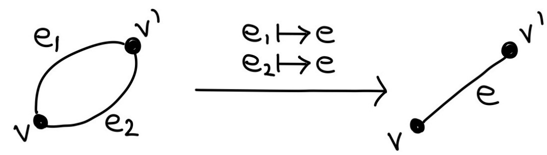


Figure 7: An example of a flag surjective but not flag bijective morphism of $\mathcal{G}$.

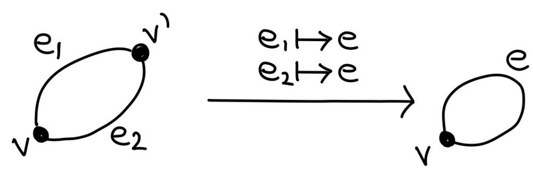


Figure 8: An example of an embedding (hence a flag bijective morphism) in $\mathcal{G}$.

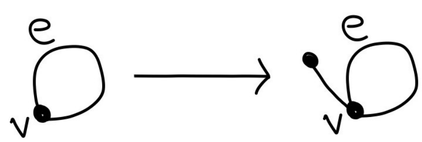


Figure 9: This map is not flag surjective and therefore not a valid morphism in $\mathcal{G}$.

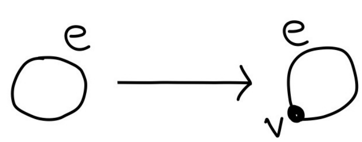


Figure 10: This example is not a valid morphism in $\mathcal{G}$ because it does not respect Condition 2.

Coend Optics for Quantum Combs

James Hefford
University of Oxford, UK

Cole Comfort
University of Oxford, UK

We compare two possible ways of defining a category of 1-combs, the first intensionally as coend optics and the second extensionally as a quotient by the operational behaviour of 1-combs on lower-order maps. We show that there is a full and bijective on objects functor quotienting the intensional definition to the extensional one and give some sufficient conditions for this functor to be an isomorphism of categories. We also show how the constructions for 1-combs can be extended to produce polycategories of n-combs with similar results about when these polycategories are equivalent. The extensional definition is of particular interest in the study of quantum combs and we hope this work might produce further interest in the usage of optics for modelling these structures in quantum theory.

1 Introduction

The traditional way in which physical systems are modelled is by considering a state space which evolves according to processes which act on that space. For example, a quantum circuit is traditionally viewed in terms of linear operators being applied to a Hilbert space; electrical circuits in terms of certain operators acting on phase space; probabilistic theories in terms of stochastic maps acting on probability spaces.

This approach has proven to be amenable to categorical analysis. For example, the ZX-calculus [15, 32], graphical affine algebra [6, 5, 18] and markov categories [20] have all been successful in formally modelling these respective classes of systems using the theory of monoidal categories. Moreover, categorical quantum mechanics [1, 14, 21] and the framework of generalised/operational probabilistic theories [2, 11] provide semantics for modelling more general quantum-like theories.

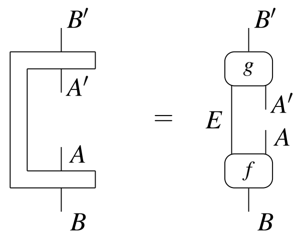
However, the approach of modelling systems merely in terms of the action of operators on the state space may not fully capture the behaviour of the system. When the collection of operators is itself regarded as the state space, this traditional approach gives little insight into the evolution of this new, "higher order" state space. What is missing is a theory of second order processes, a theory of processes which themselves act on (first order) processes. Or indeed a theory of n^{th} order processes which act on $(n-1)^{\text{th}}$ order processes.

In the theory of quantum circuits, the domain which we are chiefly interested in this paper, these higher order processes are known as quantum supermaps [9, 24]. In their most abstract formulation, it is known that there exist quantum supermaps, such as the quantum `switch`, which go beyond the standard quantum circuit model by not possessing a factorisation as a circuit with definite causal ordering of gates and no time-loops [12, 7]. Yet there is a class of supermaps which can be adequately modelled by "circuits with holes" [8, 10] where one has a framework quantum circuit with slots that can be filled with first-order maps. Indeed, all second-order deterministic single-party supermaps on quantum channels possess a factorisation as a circuit [9, 19, 24].

In this paper we restrict our attention to these "circuits with holes", otherwise known as n-combs

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 63–76, doi:10.4204/EPTCS.380.4

[8, 10]. For example, 1-combs are often drawn suggestively as diagrams of the form:



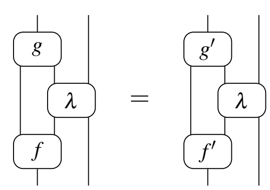
(1)

Some care is needed though to make these drawings rigorous and to demonstrate that a suitable (possibly symmetric monoidal) category of combs can be defined. In much of the quantum literature it is assumed that the base category of first order processes is compact closed, or at least embeds into one. In this case it is possible to bend input and output wires to express combs as maps without holes and use the drawing (1) in an unambiguous way; for example, see [8, 24]. Outside of the quantum literature there are approaches to defining comb diagrams without the assumption of closure [29, 27], but it is not clear when this coincides with the quantum definition.

In this article we focus on defining categories of combs without any assumptions of closure on the category of first order processes. In Section 2, we compare two constructions which take an arbitrary symmetric monoidal category and produce a symmetric monoidal category of combs. Both of these constructions represent a comb as a pair of morphisms (f,g) from the theory of first order processes, quotiented by their behaviour on first-order processes.

The first construction, which we define in Subsection 2.1, $\mathsf{Comb} : \mathsf{SymMonCat} \to \mathsf{SymMonCat}$, quotients combs by their extensional behaviour: two combs are equal when they produce the same output on all first-order inputs. In other words this identifies two combs when they appear to be the same when probed with all first order processes λ:

$(f,g) \sim_{\text{comb}} (f',g')$ when



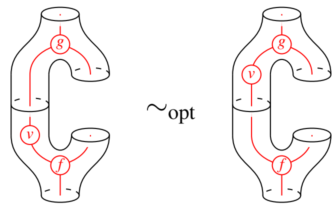
$\forall \lambda$

This equivalence relation has been discussed before [16] [1] and is perhaps the one that would be most immediate to those studying quantum theory.

The second construction, which we review in Subsection 2.2, is that of the category of coend optics, $\mathsf{Optic} : \mathsf{SymMonCat} \to \mathsf{SymMonCat}$ (which we shall henceforth just call optics) [13, 26, 28, 25]. Optics are used to encompass bidirectional data accessors familiar to the computer science community such as lenses, prisms and grates, amongst many others. Their usage to model combs and more general "circuits with holes" has been described in [29, 27]. In contrast to the previous construction this quotients the combs by their intensional behaviour, allowing first-order maps to slide along the shared environment

[1] we note that our category of combs is distinct from that developed there: the objects of their category being different than those studied in this document

connecting the two factors together:



In Subsection 2.3 we show that there is always a full and bijective on objects monoidal functor from optics to the extensional definition, $\mathsf{Optic}(\mathscr{C}) \to \mathsf{Comb}(\mathscr{C})$. We then give some sufficient conditions for this functor to exhibit an isomorphism of symmetric monoidal categories. In particular we show that when the category of first-order processes is cartesian and there exists a state for every type or when it is compact closed, the two definitions coincide. We also show that in the case of the category of unitaries between Hilbert spaces, the definitions again coincide. This case (alongside compact closed categories) is particularly important for quantum theory. We leave it as future work to fully characterise when $\mathsf{Optic}(\mathscr{C}) \cong \mathsf{Comb}(\mathscr{C})$ and note that there are important cases of combs not covered by the sufficient conditions proven in this work.

In section 3 we specialise to the case where the base category $\mathscr{C}$ is †-compact closed and restrict to the subcategory where the maps constituting the combs are the daggers of each other, that is where $g = f^\dagger$ in (1). This subcategory of the extensional category of combs collapses to the CPM-construction [30] which is used to generate a category of generalised completely positive maps from some underlying category.

In the final section 4 we turn our attention to n-combs, which map n first-order processes into first-order process. These combs naturally form a polycategory and we show that in the presence of compact closure, the extensional and intensional definitions once again coincide.

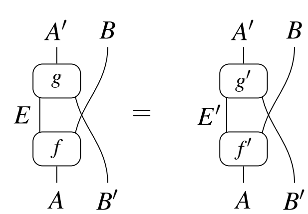
2 Combs

In this section we define two notions of factorisable 1st order single input combs. These categories are given by functors $\mathsf{SymMonCat} \to \mathsf{SymMonCat}$ whose morphisms are given by pairs of maps composable along an interface, as per the right hand side of (1). In Subsection 2.1 we establish the extensional definition; in Subsection 2.2 we review the intensional definition of optics; and in Subsection 2.3 we give sufficient conditions under which both definitions coincide.

2.1 Extensional combs

Let us begin by considering possible extensional definitions of combs. Firstly, one could ask that the combs are equal as morphisms in the original category when we extend their inputs:

$$(f,g) \sim_\sigma (f',g') \quad \iff$$

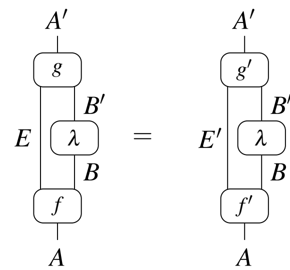


(2)

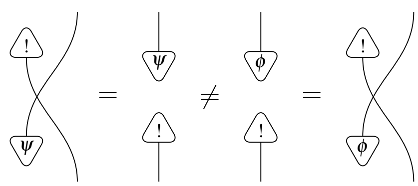
While this is an equivalence relation on pairs of morphisms, it is not a congruence with respect to composition. Suppose $(f,g) \sim_\sigma (f',g')$ and $(h,k) \sim_\sigma (h',k')$. Then $(h,k) \circ (f,g) = ((1 \otimes h)f, g(1 \otimes k)) \sim_\sigma ((1 \otimes h')f, g(1 \otimes k')) = (h',k') \circ (f,g)$ which is not in general equivalent to $(h',k') \circ (f',g')$.

We could instead ask that two combs are equivalent if they are equal on all inputs to the comb:

$$(f,g) \sim_\tau (f',g') \iff \forall \lambda : B \to B'$$



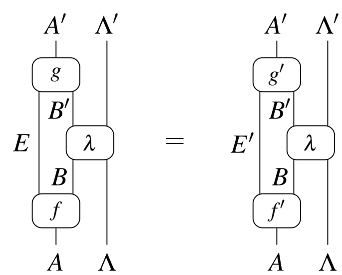
This also forms an equivalence relation on pairs of morphisms, although it is too coarse. Consider the free symmetric monoidal category generated by one object A, two states $\phi, \psi : I \to A$ and an effect $! : A \to I$ such that $! \circ \phi = ! \circ \psi = 1_I$. Then $(1_I \otimes \psi, 1_I \otimes !) \sim_\tau (1_I \otimes \phi, 1_I \otimes !)$; however evaluating these combs on the braid one finds,



So if we want comb to behave compatibly with the monoidal structure of the category, we need something stronger than equality on all inputs.

Definition 1 (Extensional Comb Equivalence)**.** We say that two combs are equivalent if they are equal on all extended inputs:

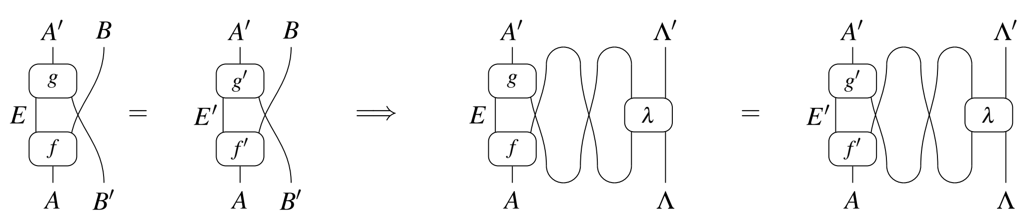
$$(f,g)_E \sim_{\text{comb}} (f',g')_{E'} \iff \begin{array}{c}\forall \Lambda, \Lambda' \\ \forall \lambda : B \otimes \Lambda \to B' \otimes \Lambda'\end{array} \tag{3}$$



This definition subsumes both of the previous definitions, but in the compact closed case (2) is sufficient to recover the full extensional equivalence.

Proposition 1. *When $\mathscr{C}$ is compact closed $(f,g) \sim_{comb} (f',g') \iff (f,g) \sim_\sigma (f',g')$.*

Proof. The forwards direction is immediate. The backwards direction follows by graphical manipulation:



□

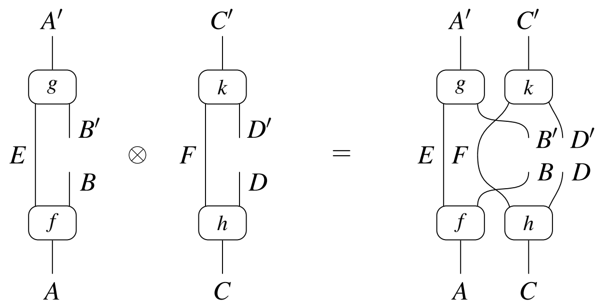
Definition 2. Given a symmetric monoidal category $\mathscr{C}$, the symmetric monoidal category of extensional combs $\mathsf{Comb}(\mathscr{C})$ has:

Objects: pairs (A,A') of objects of $\mathscr{C}$.

Morphisms: $(f,g) : (A,A') \to (B,B')$ are equivalence classes of pairs of morphisms $f : A \to E \otimes B$ and $g : E \otimes B' \to A'$ of $\mathscr{C}$ under the comb equivalence relation $\sim_{\text{comb}}$.

Composition of morphisms is given by $(f',g') \circ (f,g) = ((1 \otimes f')f, g(1 \otimes g'))$.

Monoidal structure: On objects $(A,A') \otimes (B,B') = (A \otimes B, A' \otimes B')$ and on morphisms:



The unit object is (I,I) with structural isomorphisms given by $(\lambda,\lambda^{-1}) : (A,A') \otimes (I,I) = (A \otimes I, A' \otimes I) \to (A,A')$ and (ρ,ρ^{-1}).

The symmetry is defined similarly.

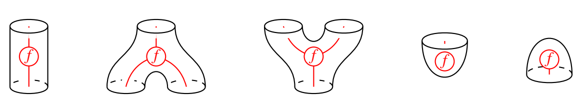
Lemma 1. Comb *defines a functor* $\mathsf{SymMonCat} \to \mathsf{SymMonCat}$.

2.2 Optics

Optics provide another potential definition of combs; albeit an intensional one, as opposed to the extensional one described in the previous subsection.

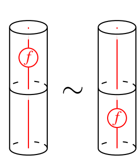
We will use the graphical calculus of internal string diagrams/pointed profunctors to work with optics. Internal string diagrams were first introduced in the Vect-enriched case in [3] and further explored in [23]. The same sort of graphical calculus was described in [29] where the author shows that they form a 2-category of pointed profunctors.

Internal string diagrams consist of usual string diagrams for monoidal categories bounded inside cobordisms. For example the identity, contravariant and covariant embeddings of the tensor product and tensor unit are drawn as follows:

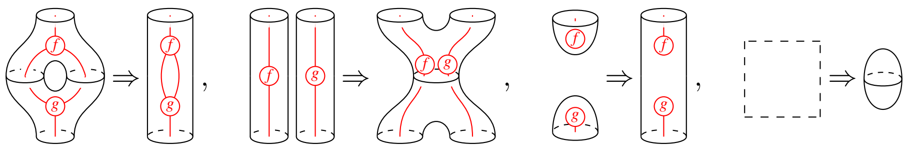


(4)

The internal diagrams can be manipulated and composed as usual, but they are constrained by the topology of the cobordisms. Moreover, when we compose these diagrams together, we are allowed to slide morphisms between them as follows:



The shapes in (4) are associative monoids and comonoids and there exist the following 2-cells which allow us to “pop bubbles” (as well as some extra coherence conditions):

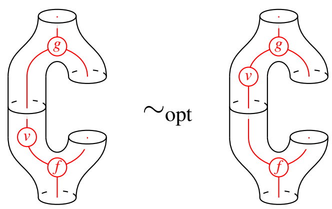


(5)

There is much more to say about pointed profunctors, but we will omit the technical discussion and refer the interested reader to [3] and [29] for a more in-depth discussion.

We are now in a position to give the definition of the category of optics.

Definition 3 (Category of optics [25, 13])**.** Given a symmetric monoidal category $\mathscr{C}$, the category of optics $\mathsf{Optic}(\mathscr{C})$, has the same objects as $\mathsf{Comb}(\mathscr{C})$. Morphisms are pairs $(f,g)_E$ like in $\mathsf{Comb}(\mathscr{C})$ however, instead of quotienting the morphisms by the equivalence relation $\sim_{\text{comb}}$, we quotient morphisms by the equivalence relation $\sim_{\text{opt}}$ imposed by embedding the combs inside the cobordisms:



(6)

The string diagrams can be freely moved around the interior of the cobordism, but can not pass through the surface: as a result we are able to slide maps on the environment wire between the two halves with the equivalence relation generated by $((v\otimes 1)f,g)_{E'} \sim (f,g(v\otimes 1))_E$.

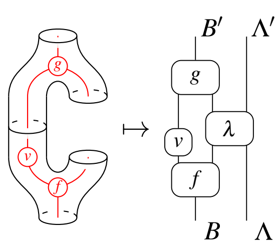
Composition, identities, and symmetric monoidal structure is as in $\mathsf{Comb}(\mathscr{C})$. That $\sim_{\text{opt}}$ is a congruence and that the composite of two optics is another optic (i.e. that the composite of the comb-shaped cobordisms in (6) can be manipulated to give another comb-shaped cobordism) follows by a composition of the 2-cells in (5), see e.g. [26] for more details.

2.3 Equivalence of the Definitions

In this section we consider the question of when $\mathsf{Optic}(\mathscr{C})$ and $\mathsf{Comb}(\mathscr{C})$ are equivalent. It is fairly straightforward to show that there is always a functor $\mathsf{Optic}(\mathscr{C}) \to \mathsf{Comb}(\mathscr{C})$ turning the intensional combs into extensional combs.

Proposition 2. *Given a symmetric monoidal category $\mathscr{C}$, there is a bijective on objects, full symmetric monoidal functor* $\mathsf{Optic}(\mathscr{C}) \to \mathsf{Comb}(\mathscr{C})$.

Proof. For each λ there is a mapping:



This preserves the sliding of morphisms v along the ancillary wire. □

Remark. Formally, the mapping above gives a cowedge for $\mathscr{C}(A, -\otimes B) \times \mathscr{C}(=\otimes B', A')$ and must therefore factor uniquely via the coend.

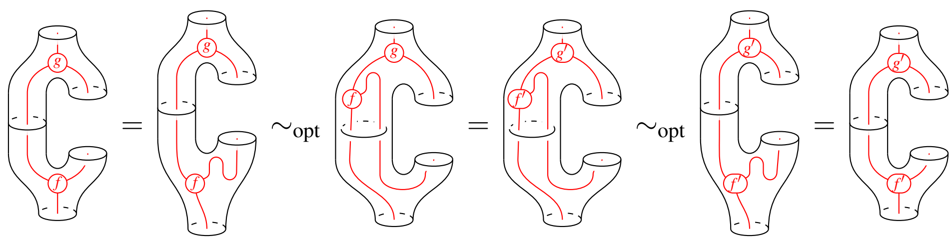
It is not immediately obvious whether the functor of the previous proposition is faithful and thus witnesses an equivalence of categories.

Counterexample 1. Consider the free commutative monoidal category generated by one object A and a single idempotent $f : A \to A$. Then $(1_A, f)_I \nsim_{\text{opt}} (f, 1_A)_I$ but $(1_A, f)_I \sim_{\text{comb}} (f, 1_A)_I$ and thus $\mathsf{Optic}(\mathscr{C}) \ncong \mathsf{Comb}(\mathscr{C})$ in this case.

We now explore some classes of categories where there is an equivalence $\mathsf{Optic}(\mathscr{C}) \cong \mathsf{Comb}(\mathscr{C})$.

Proposition 3. *Given a compact closed category $\mathscr{C}$, there is a symmetric monoidal isomorphism of categories* $\mathsf{Optic}(\mathscr{C}) \cong \mathsf{Comb}(\mathscr{C})$.

Proof.



So we have established that comb equivalence implies optic equivalence. This is sufficient to show that the functor of proposition 2 is also faithful. □

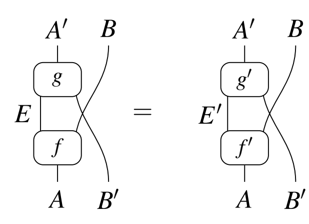
Remark. The previous result could also be established by Yoneda reduction (see e.g. [27, Sec. 4.2]) as follows:

$$\int^E \mathscr{C}(A, E\otimes B) \times \mathscr{C}(E \otimes B', A') \cong \int^E \mathscr{C}(A, E\otimes B) \times \mathscr{C}(E, B'^* \otimes A') \cong \mathscr{C}(A, B'^* \otimes A' \otimes B)$$
$$\cong \mathscr{C}(A \otimes B', A' \otimes B)$$

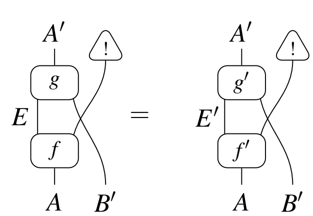
Note that $(f,g)_E \sim_{\text{comb}} (f',g')_{E'}$ implies $(f,g)_E \sim_\sigma (f',g')_{E'}$ which ensures they are the same element of the set $\mathscr{C}(A\otimes B', A' \otimes B)$.

Proposition 4. *Given a Cartesian category $\mathscr{C}$ where each type is inhabited, there is a symmetric monoidal isomorphism of categories* $\mathsf{Optic}(\mathscr{C}) \cong \mathsf{Comb}(\mathscr{C})$.

Proof. Suppose $(f,g)_E \sim_{\text{comb}} (f',g')_{E'}$. We know that these combs are equal on the braid:

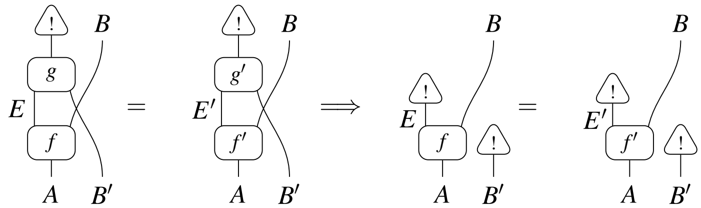


By the universal property of the product, this map is completely determined by its projections into A' and B. The former gives:

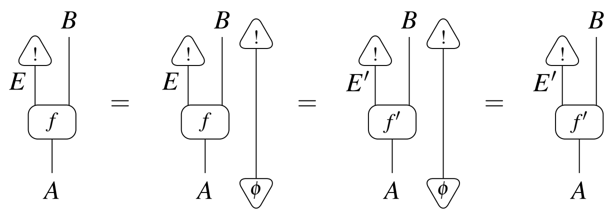


$\in \mathscr{C}(A \times B', A')$ (7)

while the latter gives

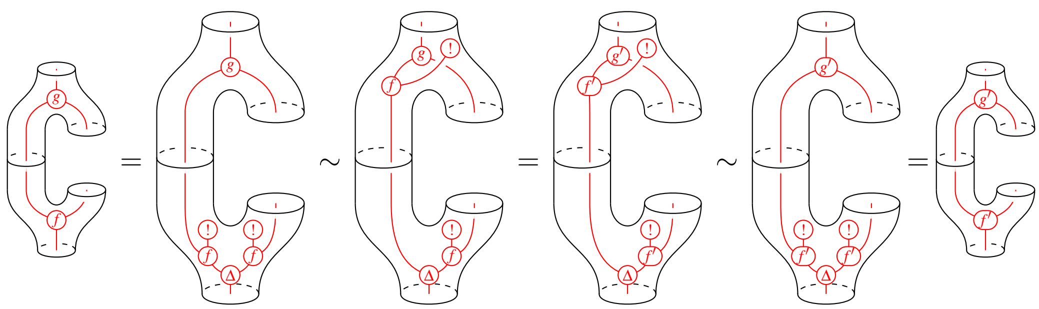


Pick a map $\phi : I \to A$, then



(8)

Thus:



□

Remark. The final part of the proof can also be derived by Yoneda reduction (see e.g. [13, Sec. 3.1]):

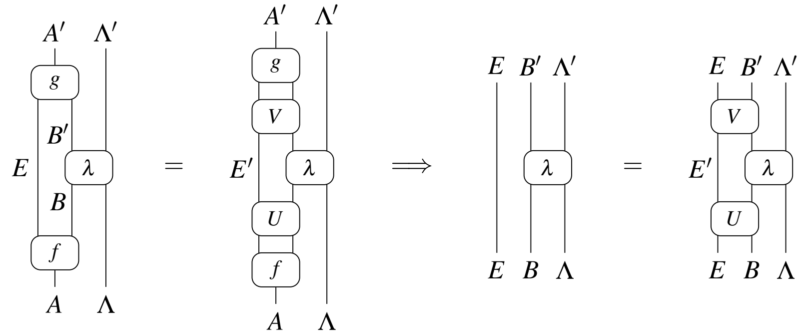
$$\int^{E} \mathscr{C}(A,E\times B)\times\mathscr{C}(E\times B',A') \cong \int^{E} \mathscr{C}(A,E)\times\mathscr{C}(A,B)\times\mathscr{C}(E\times B',A') \cong \mathscr{C}(A,B)\times\mathscr{C}(A\times B',A')$$

and then noting that the projections (7) and (8) precisely determine an element of $\mathscr{C}(A,B)\times\mathscr{C}(A\times B',A')$.

Proposition 5. *There is a symmetric monoidal isomorphism* $\mathsf{Optic}(\mathsf{Unitary}) \cong \mathsf{Comb}(\mathsf{Unitary})$*, where* $\mathsf{Unitary}$ *is the category of unitary maps between (not necessarily finite dimensional) Hilbert spaces.*

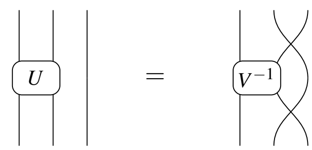
Proof. $f : A \to E \otimes B$ is a unitary and thus $A \cong E \otimes B$ are isomorphic as Hilbert spaces. Similarly from f' we see $A \cong E' \otimes B$ and from g and g', $A' \cong E \otimes B' \cong E' \otimes B'$. This means there must exist a unitary $U : E \otimes B \to E' \otimes B$ such that $f' = Uf$ and a unitary $V : E' \otimes B' \to E \otimes B'$ such that $g' = gV$.

Using the fact that $(f,g)_E \sim_{\text{comb}} (f',g')_{E'}$ and that f and g have two-sided inverses, we see that for all λ:

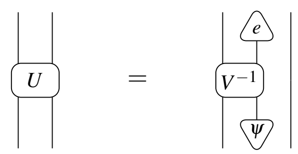


(9)

Taking $\lambda = \sigma$ we arrive at the following equality:

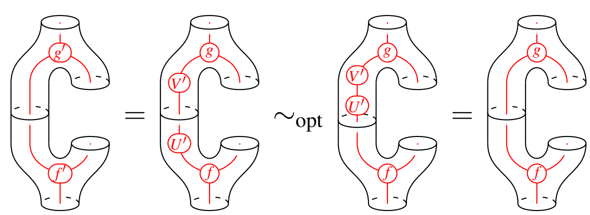


There exists a faithful embedding of Unitaries into Hilb where we can pick any state $|\psi\rangle$ and effect $\langle e|$ with $\langle e|\psi\rangle = 1$ to see that:



As a result U can be seen to $\otimes$-separate as $U = U' \otimes 1$ where $U' := (1 \otimes e)V^{-1}(1 \otimes \psi)$ must be a unitary else U could not be unitary and we would have a contradiction. Analagously one can show that V $\otimes$-separates as $V' \otimes 1$. Inserting these factorisations into the right hand side of (9) one can see that $V'U' = 1$.

Therefore:

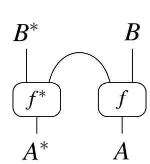


□

3 The CPM construction as optics

In this section we show that over a †-compact closed category, the CPM construction embeds within optics.

Definition 4 (CPM construction [30])**.** Given a †-compact closed category $\mathscr{C}$, the category CPM($\mathscr{C}$) of completely positive maps has the same objects as $\mathscr{C}$. A morphism $f : A \to B$ in CPM($\mathscr{C}$) is a morphism of type $A^* \otimes A \to B^* \otimes B$ in $\mathscr{C}$ of the form



(10)

where $(-)^* : \mathscr{C} \to \mathscr{C}$ is the conjugation functor. Composition and identities are inherited from $\mathscr{C}$.

Example 1. The †-symmetric monoidal category $\mathsf{CPM}(\mathsf{FHilb})$ is equivalent to the category of density operators between finite dimensional Hilbert spaces.

The Optic and Comb constructions provide another route to defining the category of completely positive maps. We write $\mathsf{Optic}^\dagger(\mathscr{C})$ and $\mathsf{Comb}^\dagger(\mathscr{C})$ for the subcategories of $\mathsf{Optic}(\mathscr{C})$ and $\mathsf{Comb}(\mathscr{C})$ respectively, generated by representatives of the form $(f, f^\dagger)_E$. The following proposition follows:

Proposition 6. *When $\mathscr{C}$ is †-compact closed, there is a symmetric monoidal isomorphism of categories* $\mathsf{Optic}^\dagger(\mathscr{C}) \cong \mathsf{Comb}^\dagger(\mathscr{C}) \cong \mathsf{CPM}(\mathscr{C})$.

Proof. (Sketch). The isomorphism $\mathsf{Optic}^\dagger(\mathscr{C}) \cong \mathsf{Comb}^\dagger(\mathscr{C})$ follows by proposition 3. The isomorphism $\mathsf{Comb}^\dagger(\mathscr{C}) \cong \mathsf{CPM}(\mathscr{C})$ is given by sending $(A,A) \mapsto A$ on objects and $(f, f^\dagger)_E$ to (10). Fullness is obvious and one can see it is faithful by inserting the braid into the comb equivalence relation (3). □

There have been attempts to generalise the CPM construction to infinite dimensional quantum systems where one does not have compact closure. For instance, in [17] the CP^∞ construction is developed which turns any monoidal †-category into a category of completely positive maps. The category $\mathsf{CP}^\infty(\mathscr{C})$ is very similar to the category $\mathsf{Comb}(\mathscr{C})$, the only difference being that CP^∞ only quantifies over the positive maps in the equivalence relation (3) (as opposed to all maps λ). In the case that $\mathscr{C}$ is †-compact closed it is known that $\mathsf{CPM}(\mathscr{C}) \cong \mathsf{CP}^\infty(\mathscr{C})$ and thus the CP^∞ construction produces an isomorphic category to $\mathsf{Optic}^\dagger$ and $\mathsf{Comb}^\dagger$. Dropping compact closure, but keeping the †-symmetric monoidal structure, $\mathsf{Optic}^\dagger$ and $\mathsf{Comb}^\dagger$ yield two potential candiates for generalised categories of completely positive maps.

4 *n*-Combs

In this section we consider generalisations of the Optic and Comb constructions to encompass *n*-combs. There are several categorical structures that could provide an adequate semantics for dealing with the many inputs and outputs that a generalised *n*-comb could have. Here we will use polycategories to handle *n*-combs.

A candidate definition of such an *n*-comb was suggested in [27] as a generalisation of the Optic construction. We generalise this even further, obtaining a polycategory. Our definition of the combs themselves is similar, but crucially our notion of composition is very different and coincides more closely with that of [16].

Definition 5. Given a symmetric monoidal category $\mathscr{C}$, the polycategory of *n*-combs $\mathsf{OPTIC}(\mathscr{C})$ has:

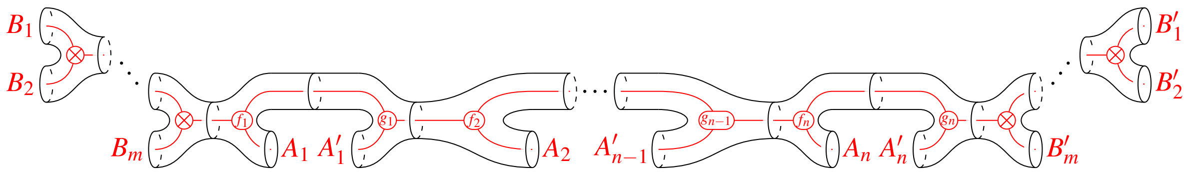
Objects: Pairs of objects in $\mathscr{C}$.

Morphisms: The polymorphisms of type $[(A_1, A_1'), \dots, (A_n, A_n')] \to [(B_1, B_1'), \dots, (B_m, B_m')]$ are elements of the set (where the zero-fold tensor in $\mathscr{C}$ is the tensor unit):

$$\int^{X_0,\dots,X_{n+1}} \mathscr{C}\left(\bigotimes_{i=1}^{n} B_i, X_0\right) \times \prod_{i=1}^{n} \mathscr{C}(X_{i-1}, X_i \otimes A_i) \times \mathscr{C}(X_i \otimes A_i', X_{i+1}) \times \mathscr{C}\left(X_{n+1}, \bigotimes_{i=1}^{n} B_i'\right)$$

For example consider the string diagram for a polymorphism of this type (drawn from left to right to conserve space):

$$(\langle f_1,\dots,f_n|g_1,\dots,g_n\rangle_{X_1,\dots,X_n} : [(A_1,A_1'),\dots,(A_n,A_n')] \to [(B_1,B_1'),\dots,(B_m,B_m')]) :=$$



The identities are the same as in optics. Given a map as above and another map

$$\langle h_0,\dots,h_\ell|k_0,\dots,k_n\rangle_{Y_1,\dots,Y_\ell} : [(C_1,C_1'),\dots,(C_\ell,C_\ell')] \to [(D_1,D_1'),\dots,(D_p,D_p')]$$

where $(C_q,C_q') = (B_j,B_j')$ for some $0 \le q \le \ell$, $0 \le j \le m$. Then the composite

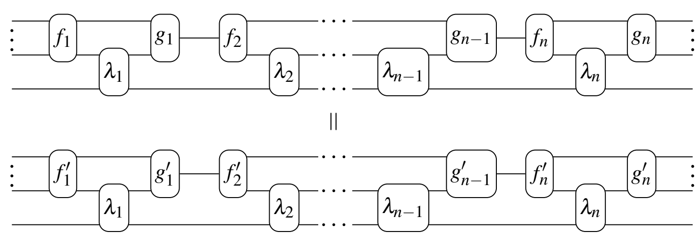
$$\langle f_1,\dots,f_n|g_1,\dots,g_n\rangle_{X_1,\dots,X_n} \circ_{(B_j,B_j')} \langle h_0,\dots,h_\ell|k_0,\dots,k_n\rangle_{Y_1,\dots,Y_\ell}$$

is given by plugging the first comb into the (B_j,B_j') hole and the collapsing the bubble. This can be verified to produce a diagram of the same shape via a lengthy, yet elementary application of the coend calculus, or equivalently a composition of the 2-cells (5) and associators.

There is also a polycategory of n-combs that generalises the Comb construction,

Definition 6. Given a symmetric monoidal category $\mathscr{C}$, the polycategory of n-combs $\mathsf{COMB}(\mathscr{C})$ has the same objects as $\mathsf{OPTIC}(\mathscr{C})$. The polymorphisms are given by tuples of maps under a generalisation of the comb equivalence relation where two combs are equivalent if they are equal on all extended inputs:

$\forall \lambda_1,\dots,\lambda_n$



Composition and identities are the same as in $\mathsf{COMB}(\mathscr{C})$.

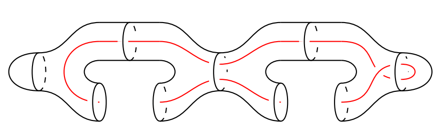
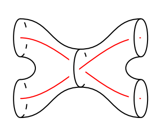
As in the case of 1-combs we can always quotient the intensional optics definition to get the extensional comb definition:

Proposition 7. *There is a full and bijective on objects polyfunctor* $\mathsf{OPTIC}(\mathscr{C}) \to \mathsf{COMB}(\mathscr{C})$.

Proof. (Sketch) The proof is similar to proposition 2: removing the cobordisms and evaluating the comb on the given $\lambda_1,\dots,\lambda_n$ gives a cowedge and thus factorises uniquely via the coend. □

Lemma 2. *When $\mathscr{C}$ is compact closed,* $\mathsf{OPTIC}(\mathscr{C})$ *and* $\mathsf{COMB}(\mathscr{C})$ *are $*$-polycategories.*

Proof. (Sketch) In $\mathsf{OPTIC}(\mathscr{C})$, the unit and counits which generate the *-polycategory structure are given by the following (rotated) internal string diagrams:

$\eta :=$  $\varepsilon :=$ 

The $*$-polycategory structure of $\mathsf{COMB}(\mathscr{C})$ is transported along the polyfunctor in Proposition 7. □

Proposition 8. *When $\mathscr{C}$ is compact closed there is an isomorphism of polycategories* $\mathsf{OPTIC}(\mathscr{C}) \cong \mathsf{COMB}(\mathscr{C})$.

Proof. (Sketch) The isomorphism is shown in a similar way to the proof of Proposition 3, by pulling all of the circuits into the same bubble. □

5 Conclusion and future work

In this article we have considered some categorical approaches to modelling combs with a particular focus on the operational motivations typically pursued in the quantum literature. There are several lines of future work we are actively investigating:

- It would be clarifying to pin down precisely when $\mathsf{Optic}(\mathscr{C}) \cong \mathsf{Comb}(\mathscr{C})$, or at least know whether this holds in cases beyond the few investigated here. Particularly for quantum theory we would like to know what happens in the case of $\mathsf{Isometry}$ and CPTP. The cases of $*$-autonomous categories and monoidally closed categories would also be interesting so we could better understand any connections with the Caus-construction [24].
- It may be possible to use profunctors to capture the causal structure of maps. Informally, one can replace causal graphs with profunctor tubes whose topology acts to restrict the families of maps that are compatible with the causal structure, for instance by enforcing one-way signalling constraints.
- We have reason to believe that the category of Tambara modules [31] (equivalently the presheaf category of optics [25]) is a good setting for modelling quantum supermaps more generally, possibly allowing for the modelling of maps like the quantum `switch` alongside combs. There are several operational principles one might ask of a quantum supermap to ensure that it is compatible with the monoidal structure of the category of first-order maps. These principles seem to translate pleasingly into the structure of Tambara module homomorphisms.
- Given a $\mathscr{V}$-enriched category $\mathscr{C}$, it is not clear to us whether the category $\mathsf{Comb}(\mathscr{C})$ inherits the enrichment. This is relevant to quantum theory because taking probabilistic mixures of quantum processes can be modelled by enrichment in the category CMon of commutative monoids [22, 21]. On the other hand, it is immediate that $\mathsf{Optic}(\mathscr{C})$ inherits the enrichment of $\mathscr{C}$ and thus might be a better setting for modelling quantum combs.
- In section 4 we provided a polycategorical semantics for n-combs. We are also persuing the possibility of a double categorical framework which might be cleaner and possibly more expressive. Indeed, as we were writing this article we became aware of [4] where such a framework is developed.

Acknowledgements

Many thanks to the authors of [3] for use of their TikZ package for producing internal string diagrams and to Guillaume Boisseau and Matt Wilson for helpful discussions and comments. JH is supported by University College London and the EPSRC [grant number EP/L015242/1].

References

[1] Samson Abramsky & Bob Coecke (2004): *A categorical semantics of quantum protocols*. In: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pp. 415–425, doi:10.1109/LICS.2004.1319636.

[2] Jonathan Barrett (2007): *Information processing in generalized probabilistic theories*. *Phys. Rev. A* 75, p. 032304, doi:10.1103/PhysRevA.75.032304.

[3] Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries & Jamie Vicary (2015): *Modular categories as representations of the 3-dimensional bordism 2-category*, doi:10.48550/ARXIV.1509.06811. arXiv:1509.06811.

[4] Guillaume Boisseau, Chad Nester & Mario Roman (2022): *Cornering Optics*, doi:10.48550/ARXIV.2205.00842. arXiv:2205.00842.

[5] Guillaume Boisseau & Paweł Sobociński (2021): *String Diagrammatic Electrical Circuit Theory*. doi:10.48550/arXiv.2106.07763.

[6] Filippo Bonchi, Robin Piedeleu, Pawel Sobociński & Fabio Zanasi (2019): *Graphical affine algebra*. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, pp. 1–12, doi:10.1109/LICS.2019.8785877.

[7] Giulio Chiribella (2012): *Perfect discrimination of no-signalling channels via quantum superposition of causal structures*. *Phys. Rev. A* 86, p. 040301, doi:10.1103/PhysRevA.86.040301.

[8] Giulio Chiribella, Giacomo Mauro D'Ariano & Paolo Perinotti (2008): *Quantum Circuit Architecture*. *Physical Review Letters* 101(6), doi:10.1103/physrevlett.101.060401.

[9] Giulio Chiribella, Giacomo Mauro D'Ariano & Paolo Perinotti (2008): *Transforming quantum operations: Quantum supermaps*. *EPL (Europhysics Letters)* 83(3), p. 30004, doi:10.1209/0295-5075/83/30004.

[10] Giulio Chiribella, Giacomo Mauro D'Ariano & Paolo Perinotti (2009): *Theoretical framework for quantum networks*. *Phys. Rev. A* 80, p. 022339, doi:10.1103/PhysRevA.80.022339.

[11] Giulio Chiribella, Giacomo Mauro D'Ariano & Paolo Perinotti (2010): *Probabilistic theories with purification*. *Phys. Rev. A* 81, p. 062348, doi:10.1103/PhysRevA.81.062348.

[12] Giulio Chiribella, Giacomo Mauro D'Ariano, Paolo Perinotti & Benoit Valiron (2013): *Quantum computations without definite causal structure*. *Phys. Rev. A* 88, p. 022318, doi:10.1103/PhysRevA.88.022318.

[13] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore & Mario Román (2020): *Profunctor optics, a categorical update*. arXiv:2001.07488.

[14] Bob Coecke (2010): *Quantum picturalism*. *Contemporary Physics* 51(1), pp. 59–83, doi:10.1080/00107510903257624.

[15] Bob Coecke & Ross Duncan (2011): *Interacting quantum observables: categorical algebra and diagrammatics*. *New Journal of Physics* 13(4), p. 043016, doi:10.1088/1367-2630/13/4/043016.

[16] Bob Coecke, Tobias Fritz & Robert W. Spekkens (2016): *A mathematical theory of resources*. *Information and Computation* 250, pp. 59–86, doi:10.1016/j.ic.2016.02.008.

[17] Bob Coecke & Chris Heunen (2016): *Pictures of complete positivity in arbitrary dimension*. Information and Computation 250, pp. 50–58, doi:10.1016/j.ic.2016.02.007. Quantum Physics and Logic.

[18] Cole Comfort & Aleks Kissinger (2021): *A Graphical Calculus for Lagrangian Relations*. doi:10.48550/arXiv.2105.06244.

[19] T Eggeling, D Schlingemann & R. F Werner (2002): *Semicausal operations are semilocalizable*. Europhysics Letters (EPL) 57(6), pp. 782–788, doi:10.1209/epl/i2002-00579-4.

[20] Tobias Fritz (2020): *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. Advances in Mathematics 370, p. 107239, doi:10.1016/j.aim.2020.107239.

[21] Stefano Gogioso & Carlo Maria Scandolo (2018): *Categorical Probabilistic Theories*. Electronic Proceedings in Theoretical Computer Science 266, pp. 367–385, doi:10.4204/eptcs.266.23.

[22] Chris Heunen & Jamie Vicary (2019): *Categories for Quantum Theory: An Introduction*. Oxford University Press, doi:10.1093/oso/9780198739623.001.0001.

[23] Nick Hu & Jamie Vicary (2021): *Traced Monoidal Categories as Algebraic Structures in Prof*. Electronic Proceedings in Theoretical Computer Science 351, pp. 84–97, doi:10.4204/eptcs.351.6.

[24] Aleks Kissinger & Sander Uijlen (2019): *A categorical semantics for causal structure*. Logical Methods in Computer Science Volume 15, Issue 3, doi:10.23638/LMCS-15(3:15)2019.

[25] Craig Pastro & Ross Street (2008): *Doubles for Monoidal Categories*. Theory and Applications of Categories 21(4), pp. 61–75.

[26] Mitchell Riley (2018): *Categories of Optics*, doi:10.48550/arXiv.1809.00738. arXiv:1809.00738.

[27] Mario Román (2020): *Comb Diagrams for Discrete-Time Feedback*, doi:10.48550/arXiv.2003.06214. arXiv:2003.06214.

[28] Mario Román (2020): *Profunctor optics and traversals*, doi:10.48550/arXiv.2001.08045. arXiv:2001.08045.

[29] Mario Román (2021): *Open Diagrams via Coend Calculus*. Electronic Proceedings in Theoretical Computer Science 333, p. 65–78, doi:10.4204/eptcs.333.5.

[30] Peter Selinger (2007): *Dagger Compact Closed Categories and Completely Positive Maps: (Extended Abstract)*. Electronic Notes in Theoretical Computer Science 170, pp. 139–163, doi:10.1016/j.entcs.2006.12.018. Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005).

[31] Daisuke Tambara (2006): *Distributors on a tensor category*. Hokkaido Mathematical Journal 35(2), pp. 379 – 425, doi:10.14492/hokmj/1285766362.

[32] John van de Wetering (2020): *ZX-calculus for the working quantum computer scientist*, doi:10.48550/arXiv.2012.13966. arXiv:2012.13966.

Compositional Modeling with Stock and Flow Diagrams

John Baez
Department of Mathematics
U. C. Riverside
California, USA
baez@math.ucr.edu

Xiaoyan Li
Department of Computer Science
University of Saskatchewan
Saskatoon, Canada
xiaoyan.li@usask.ca

Sophie Libkind
Department of Mathematics
Stanford University
University of California, USA
slibkind@stanford.edu

Nathaniel D. Osgood
Department of Computer Science
University of Saskatchewan
Saskatoon, Canada
nathaniel.osgood@usask.ca

Evan Patterson
Topos Institute
California, USA
evan@topos.institute

Stock and flow diagrams are widely used in epidemiology to model the dynamics of populations. Although tools already exist for building these diagrams and simulating the systems they describe, we have created a new package called StockFlow, part of the AlgebraicJulia ecosystem, which uses ideas from category theory to overcome notable limitations of existing software. Compositionality is provided by the theory of decorated cospans: stock and flow diagrams can be composed to form larger ones in an intuitive way formalized by the operad of undirected wiring diagrams. Our approach also cleanly separates the syntax of stock and flow diagrams from the semantics they can be assigned. We consider semantics in ordinary differential equations, although others are possible. As an example, we explain code in StockFlow that implements a simplified version of a COVID-19 model used in Canada.

1 Introduction

The theoretical advantages of compositionality and functorial semantics are widely recognized among applied category theorists. *Compositionality* means, at the very least, that systems can be described one piece at a time, with a clear formalism for composing these pieces. This formalism can appear in various styles: composing morphisms in a category, tensoring objects in a monoidal category, composing operations in an operad, etc. *Functorial semantics* then means that the map from system descriptions ("syntax") to their behavior ("semantics") preserves all the relevant forms of composition.

While these principles are elegant, in many fields it is still a challenge to produce useful software that takes advantage of them and is embraced by the intended users. This is one of the main challenges of applied category theory. Here we focus on developing software suited to one particular field: epidemiological modeling. At present this software is additionally capable of modeling a wide class of systems studied in the System Dynamics modeling discipline [12, 29].

The AlgebraicJulia ecosystem of software implements compositionality and functorial semantics in a thorough-going way [1]. Decorated and structured cospans are broad mathematical frameworks for turning "closed" system descriptions into "open" ones that can be composed along their boundaries [10, 3, 4]. One part of AlgebraicJulia, called Catlab [22], provides a generic interface for working with such cospans, among other categorical abstractions. With the help of Catlab, a tool called AlgebraicPetri was developed to work with one approach to epidemiological modeling based on Petri nets [18]. Here we explain a new tool, StockFlow, which handles a more flexible and more widely used formalism for epidemiological modeling: stock and flow diagrams.

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 77–96, doi:10.4204/EPTCS.380.5

In Section 2 we review how stock and flow diagrams are used in epidemiological modeling, and discuss some shortcomings of existing software for working with these diagrams. In Section 3 we first use decorated cospans to formalize a simple class of open stock and flow diagrams and their differential equation semantics, and then sketch how to extend this class to the full-fledged diagrams actually used in our software. In Section 4 we describe the software package, StockFlow, that we have developed to work with stock-flow diagrams compositionally and implement a functorial semantics for them. The reader can find the StockFlow repository on GitHub at `https://github.com/AlgebraicJulia/StockFlow.jl`.

2 Epidemiological modeling with stock and flow diagrams

Effective decision-making regarding prevention, control, and service delivery to address the health needs of the population involves reasoning about diverse complexities: policy resistance, feedbacks, heterogeneities, multi-condition interactions, and nonlinearities that collectively give rise to counterintuitive results [28, 30]. For over a century, researchers and practitioners have used epidemiology models to address such challenges. Since dynamic epidemiological modeling was first applied to communicable diseases [17, 24, 25], it has both deepened its reach in that area [2, 8] and spread to many other subdomains of epidemiology, including chronic, behavioural, environmental, occupational, and social epidemiology, as well as spheres such as mental health and addictions. Reflecting a world in which growing global interconnection is juxtaposed with increasing ecosystem encroachment and climate stresses, the rise of the "One Health" perspective [7, 21] has motivated such modeling to increasingly incorporate dynamics from domains such as ecology, veterinary and agricultural health, and social dynamics and inequities. Such efforts have come to define the field of mathematical and computational epidemiology.

The earliest and still most common epidemiological models are sets of ordinary differential equations, typically used to characterize epidemiological dynamics in an aggregate fashion [2]. Delay and partial differential equations have also been widely applied. Recent decades have witnessed a rapid growth in use of agent-based models. Although the techniques explored here may be more widely applicable, we focus on aggregate models described using differential equations.

Contemporary aggregate-level modeling involves widespread informal use of diagrams, with the most prevalent type of such diagrams being transition diagrams and their richer and more formal cousins, "stock and flow diagrams" [29], as depicted in Figure 1. Transition diagrams are a minimalist box-and-arrow formalism which draws state variables as boxes and transitions as arrows. Traditionally, most mathematical epidemiologists have focused directly on the underlying differential equations, regarding such diagrams only as an informal presentation of the equations. Thus, diagrams are commonly treated either as ephemeral artifacts useful for thinking out structures and then discarded, or as an expedient aid for communication.

Amongst the notable minority of health modelers who employ stock and flow diagrams (also called "Forrester diagrams", and termed here "stock-flow diagrams" for brevity), these diagrams play roles at different stages of the modeling process. Stock-flow diagrams depict state variables as stocks (rectangles), changes to those stocks as flows (thick arrows also termed "material connections"), constants and auxiliary variables (also called "dynamic variables"), and links (arrows sometimes called "informational connections" or simply "connections") characterizing instantaneous dependencies.

Stock-flow diagrams serve as the central formalism in the modeling tradition of System Dynamics, initiated by Forrester in the 1950s [12, 29]. Since the 1980s, System Dynamics software has provided refined, visually accessible, declarative user interfaces for interactively building, and browsing stock-flow

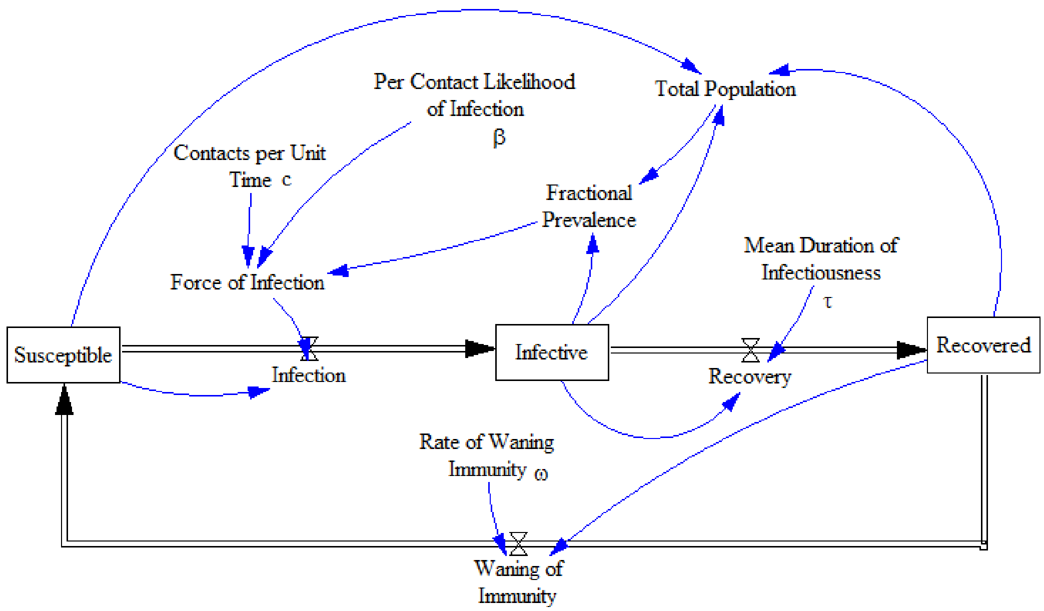


Figure 1: An example stock-flow diagram. This stock-flow diagram has three stocks labeled “Susceptible”, “Infective”, and “Recovered”; three flows labeled “Infection”, “Recovery”, and “Waning of Immunity”; many auxiliary variables including “Force of Infection” and “Total Population”; and links depicted by blue arrows.

diagrams [23]. While such packages are designed to ensure transparency of model structure to modelers and stakeholders [23], they also serve as simulation tools. For that purpose, the System Dynamics tradition universally interprets stock-flow diagrams as characterizing ordinary differential equations. Stocks represent the state variables; their formulation requires specifying an initial value. Flows represent the differentials associated with stocks, and are each associated with a modeler-specified mathematical expression specifying the flow rate (quantity per unit time) as a function of other variables. Each constant variable is associated with a real scalar. Auxiliary variables generally reflect quantities of domain significance that depend instantaneously on other model quantities. Each such auxiliary variable is associated with a modeler-specified expression characterizing the value of that auxiliary as a function of the current value of other variables (stocks, flows, constants and other auxiliary variables).

Reflecting the strong emphasis that System Dynamics practice places on stakeholder engagement and participatory model building, models built in the stock-flow paradigm are routinely shown to stakeholders without modeling background—be they domain experts from a modeling team, stakeholders, or community members—to elicit critiques and suggestions [15, 23, 31]. System Dynamics has also long sought to recognize, codify, and exploit widespread use of modeling idioms. Thus, researchers and practitioners have formalized dozens of simple stock-flow diagrams called “molecules” for reuse in modeling [14]. Some simulation packages provide molecules as pre-specified templates defined by the software, and mechanisms for for directly incorporating built-in templates for such molecules into models. When a molecule is added to a model, the elements of the molecule—such as stocks and flows—are simply added as elements of the surrounding diagram, rather than being reused as higher level abstractions. Moreover, because such libraries of molecules are fixed, such molecules cannot be created or packaged up by the user.

Software packages for stock-flow diagrams have as a central feature the simulation, via numerical integration, of the system of ordinary differential equations described by these diagrams. Many such packages also offer additional forms of model analysis, including identification of feedback loops, performing tests of dimensional homogeneity based on modeler unit annotations, sensitivity analysis and calibration. Some tools support more sophisticated forms of analysis, such as those involving Markov Chain Monte Carlo and extended Kalman filtering. But while existing stock-flow modeling tools offer refined interfaces for building, exploring and simulating models, their support for modern modeling practice is hampered by significant rigidity and several additional shortcomings. The present paper focuses on addressing two limitations of contemporary tools.

First, and most notable from a categorical perspective, existing tools *lack support for composition* of models, despite there being several natural ways in which models might be composed. Instead, each model is currently treated in isolation. If models are composed at all, it is by either outputting data files from one and importing such data into another, or by creating, via an ad hoc process, a third model that contains both of the original models.

Second, existing stock-flow modeling tools *privilege a single semantics* associated with stock-flow diagrams: the interpretation of these diagrams as ordinary differential equations. While alternative interpretations can sometimes be force-fit—for example, a difference equation interpretation by using Euler integration, or a stochastic differential equation interpretation using formulas for flows drawing from suitable probability distributions—they are commonly awkward, obscure, and error-prone. Although particular packages allow for select additional analyses—for example, identification of feedback loops—such features are hard-coded, and many analysis tools demonstrated as valuable by research [13, 16, 26] have not been incorporated in extant software packages.

We turn next to a mathematical framework that provides a remedy for these deficiencies: an explicitly compositional framework where "open" stock-flow diagrams become morphisms in a category and where semantics is described as a functor from this category to some other category. For reasons of space we only describe one choice of semantics, but the clear separation of syntax and semantics permits swapping out this choice for others.

3 The mathematics of stock-flow diagrams

Stock-flow diagrams come in many variants. To illustrate our methodology we begin with a very simple kind. In our code we have implemented a more sophisticated variant with additional features, but the ideas are easier to explain without those features. Our main goal is to study *open* stock-flow diagrams—that is, stock-flow diagrams in which various stocks are specified as "interfaces." We can treat open stock-flow diagrams with two interfaces as morphisms of a category. Composing these morphisms then lets us build larger diagrams from smaller ones. Alternatively, we can compose stock-flow diagrams with any number of interfaces using an operad. Both approaches let us describe the differential equation semantics for open stock-flow diagrams following a paradigm already explored for open Petri nets with rates [4, 5]. We describe that paradigm here.

3.1 A category of stock-flow diagrams

As a first step, we define "primitive" stock-flow diagrams with stocks, flows, and links but not the all-important functions that describe the rate of each flow. For this, we consider a category H freely generated by these objects and morphisms:

$$\begin{array}{ccc} \text{flow} & \underset{d}{\overset{u}{\rightrightarrows}} & \text{stock} \\ {\scriptstyle t}\nwarrow & & \nearrow{\scriptstyle s} \\ & \text{link} & \end{array}$$

We call a functor $F\colon \mathsf{H} \to \mathsf{FinSet}$ a **primitive stock-flow diagram**. It amounts to the following:

- a finite set of stocks $F(\text{stock})$,
- a finite set of flows $F(\text{flow})$,
- functions $F(u), F(d)\colon F(\text{flow}) \to F(\text{stock})$ assigning to each flow the stock **upstream** from it, and the stock **downstream** from it,
- a finite set of links $F(\text{link})$,
- functions $F(s)\colon F(\text{link}) \to F(\text{stock}), F(t)\colon F(\text{link}) \to F(\text{flow})$ assigning to each link its **source**, which is a stock, and its **target**, which is a flow.

Given $f \in F(\text{flow})$, we say f **flows from** the upstream stock $F(u)(f)$ and **flows to** the downstream stock $F(d)(f)$. We say that a link $\ell \in F(\text{link})$ **points from** its source $F(s)(\ell)$ and **points to** its target $F(t)(\ell)$. There is a category of primitive stock-flow diagrams, $\mathsf{FinSet}^{\mathsf{H}}$, where the objects are functors from H to FinSet and a morphism from $F\colon \mathsf{H} \to \mathsf{FinSet}$ to $G\colon \mathsf{H} \to \mathsf{FinSet}$ is a natural transformation.

Primitive stock-flow diagrams are useful for *qualitative* aspects of modeling, since they clearly show which flows depend on which stocks; as such, they can be seen as a restricted form of *system structure diagrams* [20, 9] used in System Dynamics practice. But they become useful for *quantitative* modeling and simulation only when we equip them with functions saying how the rate of each flow depends on the value of each stock. Thus, we define a **stock-flow diagram** to be a pair (F, ϕ) consisting of an object $F \in \mathsf{FinSet}^{\mathsf{H}}$ and a continuous function called a **flow function**

$$\phi_f\colon \mathbb{R}^{F(t)^{-1}(f)} \to \mathbb{R}$$

for each flow $f \in F(\text{flow})$, where $F(t)^{-1}(f)$ is the set of links with target f:

$$F(t)^{-1}(f) = \{\ell \in F(\text{link}) \mid F(t)(\ell) = f\}.$$

The idea is that the flow function ϕ_f says how the rate of the flow f depends on the values of all the stocks with links pointing to it. We make this precise in Section 3.4 when we introduce a semantics that maps each stock-flow diagram to a first-order differential equation. But rates and values play no formal role in this section.

To define a category of stock-flow diagrams, we need to define morphisms between them. What is a morphism from (F, ϕ) to (G, ψ)? It is a natural transformation $\alpha\colon F \Rightarrow G$ with an extra property. Because α is natural, we get a commutative square

$$\begin{array}{ccc} F(\text{link}) & \xrightarrow{\alpha(\text{link})} & G(\text{link}) \\ {\scriptstyle F(t)}\downarrow & & \downarrow{\scriptstyle G(t)} \\ F(\text{flow}) & \xrightarrow[\alpha(\text{flow})]{} & G(\text{flow}). \end{array}$$

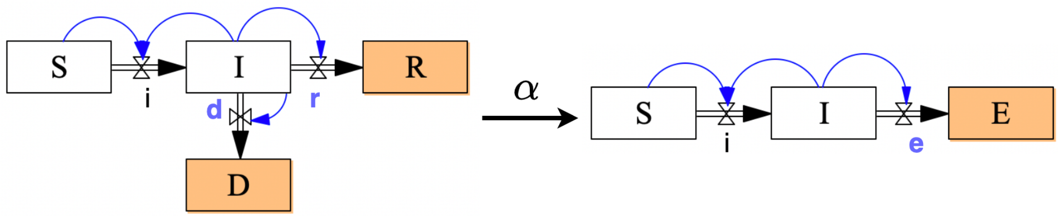


Figure 2: At left is a stock-flow diagram (F,ϕ) with stocks S,I,R,D corresponding to susceptible, infected, recovered and deceased populations and flows i,r,d corresponding to infection, recovery and death. Links are shown in blue. At right we see a simpler stock-flow diagram (G,ψ) where recovered and deceased populations are lumped into a single "removed" stock E, and recovery and death are lumped into a single "removal" flow e. There is an evident morphism $\alpha\colon (F,\phi)\to(G,\psi)$ sending R and D to E and sending r and d to e.

Thus, letting $g=\alpha(\text{flow})(f)$ for $f\in F(\text{flow})$, we get a map

$$\alpha(\text{link})\colon\, F(t)^{-1}(f)\to G(t)^{-1}(g)$$

and thus a linear map

$$\alpha(\text{link})^*\colon\, \mathbb{R}^{G(t)^{-1}(g)}\to\mathbb{R}^{F(t)^{-1}(f)}$$

given by precomposition:

$$\alpha(\text{link})^*(x)=x\circ\alpha(\text{link}).$$

We say that α is a **morphism of stock-flow diagrams** from (F,ϕ) to (G,ψ) if

$$\psi_g=\sum_{f\in\alpha(\text{flow})^{-1}(g)}\phi_f\circ\alpha(\text{link})^* \tag{1}$$

for every $g\in G(\text{flow})$. This equation expresses rates of flows in (G,ψ) as sums of rates of flows in (F,ϕ). For example, Figure 2 shows a morphism of stock-flow diagrams in which two flows, "recovery" r and "death" d, are mapped to a single "removal" flow e. The above equation implies that

$$\psi_e=\phi_r\circ\alpha(\text{link})^*+\phi_d\circ\alpha(\text{link})^*.$$

This equation says that the rate of the flow e is the sum of the rates of r and d.

Composition of morphisms between stock-flow diagrams is just composition of their underlying natural transformations; one can show that indeed the composite of two natural transformations obeying Eq. (1) again obeys this equation. We thus obtain a category of stock-flow diagrams, which we call $\mathsf{StockFlow}$.

3.2 Open stock-flow diagrams

We can build larger stock-flow diagrams by gluing together smaller ones. There are a number of choices of how to formalize this. Here we glue together two stock-flow diagrams by identifying two collections of stocks to serve as "interfaces." Thus, we define an **open stock-flow diagram** with finite sets X and Y

as interfaces to be a stock-flow diagram (F,ϕ) equipped with functions from X and Y to its set of stocks:

$$\begin{array}{ccccc} & & F(\text{stock}) & & \\ & \overset{i}{\nearrow} & & \overset{o}{\nwarrow} & \\ X & & & & Y \end{array}$$

We call this an open stock-flow diagram from X to Y and write it tersely as $(F,\phi)\colon X \to Y$, despite the maps i and o being a crucial part of the structure.

We can compose open stock-flow diagrams from X to Y and from Y to Z to obtain one from X to Z. To formalize this composition process we use Fong's theory of decorated cospans [10]. However, to make composition associative and get a category we need to use *isomorphism classes* of open stock-flow diagrams. Two open stock-flow diagrams (F,ϕ) and (F',ϕ') from X to Y are **isomorphic** if there is an isomorphism of stock-flow diagrams $\alpha\colon (F,\phi) \to (F',\phi')$ such that this diagram commutes:

$$\begin{array}{ccccc} & & F(\text{stock}) & & \\ & \overset{i}{\nearrow} & & \overset{o}{\nwarrow} & \\ X & & \Big\downarrow{\scriptstyle \alpha(\text{stock})} & & Y \\ & \underset{i'}{\searrow} & & \underset{o'}{\swarrow} & \\ & & F'(\text{stock}) & & \end{array}$$

Using the theory of decorated cospans, we obtain:

Theorem 3.1. *There is a category* $\mathsf{Open}(\mathsf{StockFlow})$ *such that:*

- *An object is a finite set X.*
- *A morphism from X to Y is an isomorphism class of open stock-flow diagrams from X to Y.*

This is a symmetric monoidal category, indeed a hypergraph category.

Proof Sketch. This follows from the theory of decorated cospans [10] once we check the following facts. For each finite set there is a category $\mathsf{C}(S)$ whose

- objects are stock-flow diagrams with S as their set of stocks, and
- morphisms are morphisms α of stock-flow diagrams where $\alpha(\text{stock})$ is the identity on S.

Let $C(S)$ be the set of isomorphism classes of objects in this category. A map of finite sets $f\colon S \to S'$ functorially determines a map from $C(S)$ to $C(S')$, and the resulting functor C is symmetric lax monoidal from $(\mathsf{FinSet},+)$ to $(\mathsf{Set},\times)$, where the laxator

$$\gamma\colon C(S)\times C(S') \to C(S+S')$$

maps a pair of stock-flow diagrams to their "disjoint union." It follows from [10, Prop. 3.2] that we get the desired symmetric monoidal category $\mathsf{Open}(\mathsf{StockFlow})$, and from [10, Thm. 3.4] that this is a hypergraph category. □

The point of making open stock-flow diagrams into the morphisms of a hypergraph category is that it gives ways of composing these diagrams that are more flexible than just composing them "end-to-end" (ordinary composition of morphisms) and "side-by-side" (a parallel arrangement expressed by tensoring). Indeed, hypergraph categories are algebras of an operad, sometimes called the operad of undirected wiring diagrams, that encapsulates a wide range of composition strategies [11]. We use this approach in our code, and instead of working with cospans we actually use multicospans [19, 27], a mild generalization that allows for open stock-flow diagrams with any number of interfaces, not just two.

We conclude with a technical remark on Theorem 3.1. In fact there is a symmetric lax monoidal functor $\mathsf{C}\colon (\mathsf{FinSet},+) \to (\mathsf{Cat},\times)$ that sends each finite set S to a *category* of stock-flow diagrams with S as their set of stocks. Theorem 2.2 of [4] thus gives a symmetric monoidal double category $\mathbb{O}\mathbf{pen}(\mathsf{StockFlow})$ where objects are finite sets and horizontal 1-cells are actual open stock-flow diagrams, not mere isomorphism classes of these.

This double category allows us to work with maps *between* open stock-flow diagrams. This should be useful for mapping several stocks to a single stock in a simplified stock-flow diagram, as in Figure 2, or embedding a stock-flow diagram in a more complicated one. However, StockFlow currently does not attempt to support maps between open stock-flow diagrams, so Theorem 3.1 suffices for us. Indeed, when working with a mere *category* of open stock-flow diagrams, as opposed to a double category, we can define an isomorphic category using structured rather than decorated cospans: for open stock-flow diagrams, the difference only becomes visible at the double category level. Thus, our treatment using decorated cospans looks forward to a future where we work with maps between open stock-flow diagrams.

3.3 Open dynamical systems

Our next goal is to define a semantics for stock-flow diagrams mapping each such diagram to a dynamical system: a system of differential equations that describes the continuous-time evolution of the value of each stock. This semantics is implicit in the usual applications of stock-flow diagrams; indeed, the stock-flow diagram is sometimes regarded merely as a convenient notation for a dynamical system. While we illustrate the choice of a semantics for stock-flow diagrams using the continuous dynamical system interpretation, this semantics holds no privileged status, and there are several other semantics of practical value that could be employed instead.

In fact, our semantics is more general than suggested above: we describe a map from *open* stock-flow diagrams to *open* dynamical systems. Our strategy for defining this semantics closely follows the strategy already used for open Petri nets with rates [3, 4, 5] and implemented for epidemiological models using AlgebraicJulia [18].

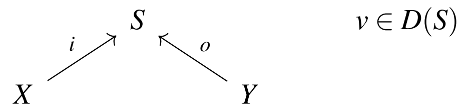
For Petri nets with rates, the dynamics is typically described by the "law of mass action," which only produces dynamical systems that are polynomial-coefficient vector fields on $\mathbb{R}^n$. In stock-flow diagrams this restriction is dropped, but the rate of any flow out of its upstream stock equals the rate of flow into its downstream stock, so the total value of all stocks is conserved. However, the more general stock-flow diagrams of Section 3.5 no longer obey this conservation law, since they allow "inflows" and "outflows" to the diagram as a whole. With these generalizations, stock-flow diagrams become strictly more general than Petri nets with rates—at least in terms of the dynamical systems they can describe.

We begin by defining a **dynamical system** on a finite set S to be a continuous vector field $v\colon \mathbb{R}^S \to \mathbb{R}^S$. In our applications, S will be the set of stocks of some stock-flow model, and the vector field v is used to write down a differential equation describing the dynamics:

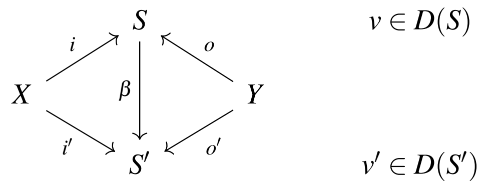
$$\frac{dx(t)}{dt} = v(x(t))$$

where at each time t, the vector $x(t) \in \mathbb{R}^S$ describes the value of each stock at time t. Since the vector field is continuous, the Peano existence theorem implies that, for any initial value $x(0) \in \mathbb{R}^S$, the above equation has a solution for all t in some interval $(-\varepsilon, \varepsilon)$. However, the solution may not be unique unless we require that v be nicer. The theory we develop now can be modified to add extra restrictions, simply by replacing continuous functions with functions of a suitably nicer sort.

We define an **open** dynamical system from the finite set X to the finite set Y to be a pair (S, v), consisting of a finite set S and a dynamical system v on S, together with functions from X and Y into S. We depict this as follows:



where $D(S)$ is the set of all dynamical systems on S. Two open dynamical systems (S, v) and (S', v') from X to Y are **isomorphic** if there is a bijection $\beta \colon S \to S'$ such that the following diagram commutes:



and $\beta_* \circ v \circ \beta^* = v'$, where $\beta_* : \mathbb{R}^S \to \mathbb{R}^{S'}$ is the pushforward map defined by

$$\beta_*(x)(\sigma') = \sum_{\sigma \in \beta^{-1}(\sigma')} x(\sigma) \qquad \forall x \in \mathbb{R}^S,\ \sigma' \in S'.$$

We can then construct a category where objects are finite sets and morphisms from X to Y are isomorphism classes of open dynamical systems from X to Y. This was done in [5, Theorem 17] by applying Fong's theory of decorated cospans to a functor $D \colon \mathsf{FinSet} \to \mathsf{Set}$ sending any finite set S to the set $D(S)$ of dynamical systems on S:

Theorem 3.2 (Baez–Pollard). *There is a category* $\mathsf{Open}(\mathsf{Dynam})$ *such that:*

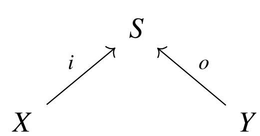
- *An object is a finite set X.*
- *A morphism from X to Y is an isomorphism class of open dynamical systems from X to Y.*

This is a symmetric monoidal category, indeed a hypergraph category.

In fact, there is a symmetric lax monoidal functor $\mathsf{D} \colon (\mathsf{FinSet}, +) \to (\mathsf{Cat}, \times)$ that maps any set S to the discrete category on the set $D(S)$ described above. The theory of decorated cospans then gives a symmetric monoidal *double* category $\mathbb{O}\mathbf{pen}(\mathsf{Dynam})$ where objects are finite sets and horizontal 1-cells are open dynamical systems. This is discussed in [4, Sec. 6.4].

3.4 Open dynamical systems from open stock-flow diagrams

Next we describe a functor sending any open stock-flow diagram to an open dynamical system. Suppose we have an open stock-flow diagram $(F, \phi) \colon X \to Y$, equipped with the cospan



where $S = F(\text{stock})$. Then there is an open dynamical system $v(F,\phi)$ on S given by

$$v(F,\phi)(x)(\sigma) = \sum_{f \in F(d)^{-1}(\sigma)} \phi_f(x \circ F(s)) \; - \sum_{f \in F(u)^{-1}(\sigma)} \phi_f(x \circ F(s)) \qquad \forall x \in \mathbb{R}^S, \sigma \in S. \tag{2}$$

This formula looks a bit cryptic, so let us explain it. Taking the expression $\mathbb{R}^S$ seriously, we can think of $x \in \mathbb{R}^S$ as a real-valued function on the set S of stocks. Each flow $f \in F(\text{flow})$ has a set $F(t)^{-1}(f)$ of links with f as target, so there is an inclusion of sets $F(t)^{-1}(f) \hookrightarrow F(\text{link})$, and we can thus form the composite

$$F(t)^{-1}(f) \hookrightarrow F(\text{link}) \xrightarrow{F(s)} F(\text{stock}) \xrightarrow{x} \mathbb{R}$$

which for short we call simply

$$x \circ F(s) \in \mathbb{R}^{F(t)^{-1}(f)}.$$

For each link ℓ with f as its target, this composite gives the value of the stock that is ℓ's source. Applying the function $\phi_f \colon \mathbb{R}^{F(t)^{-1}(f)} \to \mathbb{R}$, we obtain the rate of the flow f:

$$\phi_f(x \circ F(s)) \in \mathbb{R}.$$

This quantity has the effect of increasing the stock $d(f)$ and also decreasing the stock $u(f)$. Thus the rate of change of any stock $\sigma \in S$ is

$$\sum_{f \in F(d)^{-1}(\sigma)} \phi_f(x \circ F(s)) - \sum_{f \in F(u)^{-1}(\sigma)} \phi_f(x \circ F(s)).$$

This gives our formula for $v(F,\phi)$ in Equation (2).

Now, recall that in Theorem 3.1 the category of open stock-flow diagrams was defined as a decorated cospan category using the functor $C \colon \mathsf{FinSet} \to \mathsf{Set}$, while in Theorem 3.2 the category of open dynamical systems was defined in a similar way using the functor $D \colon \mathsf{FinSet} \to \mathsf{Set}$. According to the theory [10], to obtain a semantics mapping open stock-flow diagrams to open dynamical systems, we need to define a natural transformation $\theta \colon C \Rightarrow D$. We do this as follows: for each finite set S, define $\theta(S)$ to map the isomorphism class (F,ϕ) in $C(S)$ to the isomorphism class of $v(F,\phi)$ in $D(S)$.

Theorem 3.3. *There is a functor*

$$v \colon \mathsf{Open}(\mathsf{StockFlow}) \to \mathsf{Open}(\mathsf{Dynam})$$

sending

- *any finite set to itself,*
- *the isomorphism class of the stock-flow diagram (F,ϕ) made open as follows:*

$$\begin{array}{ccccc} & & F(\text{stock}) & & \\ & \overset{i}{\nearrow} & & \overset{o}{\nwarrow} & \\ X & & & & Y \end{array}$$

to the isomorphism class of the open dynamical system

$$\begin{array}{ccccc} & & F(\text{stock}) & & \\ & \overset{i}{\nearrow} & & \overset{o}{\nwarrow} & \\ X & & & & Y \end{array} \qquad v(F,\phi) \in D(S).$$

This is a symmetric monoidal functor, indeed a hypergraph functor.

Proof Sketch. By [10, Thm. 4.1] it suffices to check that $\theta\colon C \Rightarrow D$ is indeed a natural transformation and furthermore a *monoidal* natural transformation. □

With more work one can extend the natural transformation v to a monoidal natural transformation between the 2-functors $\mathsf{C}\colon \mathsf{FinSet} \to \mathsf{Cat}$ and $\mathsf{D}\colon \mathsf{FinSet} \to \mathsf{Cat}$. By [4, Thm. 2.5], this gives a symmetric monoidal double functor from $\mathbb{O}\mathbf{pen}(\mathsf{StockFlow})$ to $\mathbb{O}\mathbf{pen}(\mathsf{Dynam})$. However, we do not need this yet in our code.

3.5 Full-fledged stock-flow diagrams

In Section 3.1 we defined a simple category of stock-flow diagrams, called StockFlow. Stock-flow diagrams of this type capture two main features of the diagrams used by practitioners: (1) flows between stocks and (2) links that represent the dependency of flow rates on the values of particular stocks. However, the stock-flow diagrams used in epidemiological modeling have additional useful features. Our "full-fledged" stock-flow diagrams include auxiliary variables, sum variables, and partial flows.

Auxiliary variables are quantities on which flow functions can depend. An auxiliary variable is linked to stocks and other auxiliary variables and is equipped with an arbitrary continuous function of the values of stocks and variables to which it is linked. In Figure 1, "Fractional Prevalence", "Force of Infection", and "Infection" are all examples of auxiliary variables. Auxiliary variables are important to practitioners for several reasons. First, they simplify model specification because they are reusable: instead of computing each flow rate directly as a function of stocks, we can often compute them more simply with the help of auxiliary variables. Many flow rates can depend on a single auxiliary variable. Second, they often represent quantities that are of interest to stakeholders; representing these quantities explicitly make them easier to track throughout a simulation, such as for comparison with empirical data. Third, they are practical for the communication and revision of models. Auxiliary variables give a meaningful decomposition of the flow functions, and changing a single auxiliary variable automatically revises all the flow functions that depend on it, which eliminates the need to revise all these flow functions separately.

While not explicitly distinguished in current stock-flow modeling packages, flow functions often rely on a special case of auxiliary variables called *sum variables*. Such a variable equals the sum of the values of some subset of the stocks. In epidemiology, this frequently corresponds to the size of a population or sub-population. For example, "Total Population" in Figure 1 is a sum variable. In general, a sum variable may link to only a subset of stocks. Sum variables can be seen as a particular type of auxiliary variable in which the function merely sums the values of the stocks to which it is linked—and thanks to this fact, we do not need to label sum variables with functions.

Finally, in the simple stock-flow diagrams described earlier, each flow must have an upstream stock and a downstream stock. However, practitioners often use diagrams including *partial flows*, which may have only an upstream stock or only a downstream stock. These represent the creation or the destruction of some resource, and are commonly used to represent open populations.

Figure 3 presents the category H_f used to define full-fledged stock-flow diagrams. A **full-fledged stock-flow diagram** is a pair (F, ϕ) consisting of:

- a functor $F\colon \mathsf{H}_f \to \mathsf{FinSet}$ such that the functions $F(\mathrm{if})$ and $F(\mathrm{of})$ are injective;
- a continuous function $\phi_v\colon \mathbb{R}^{F(\mathrm{lv})^{-1}(v)} \times \mathbb{R}^{F(\mathrm{slv})^{-1}(v)} \to \mathbb{R}$ for each $v \in F(\mathrm{variable})$, called an **auxiliary function**.

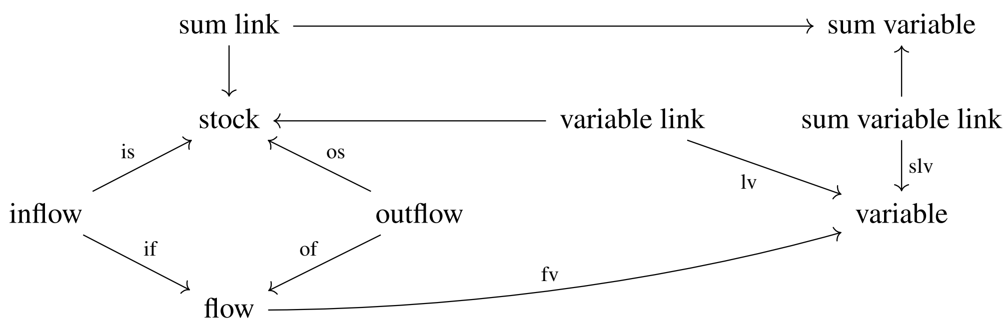


Figure 3: The free category on this diagram, called H_f, is used to define full-fledged stock-flow diagrams. We have named only some of the arrows here.

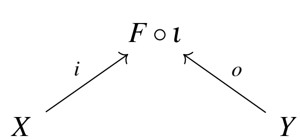
The elements of $F(\text{variable})$ are called **auxiliary variables**. The idea is that in a full-fledged stock-flow diagram each flow has its rate equal to some auxiliary variable. Each auxiliary variable can depend on any finite multiset of sum variables and stocks, and each sum variable can depend on any finite multiset of stocks.

Given an inflow $f \in F(\text{inflow})$, we say that the stock $F(\text{is})(f)$ is the **upstream** stock of the flow $F(\text{if})(f)$. Similarly, given an outflow $f \in F(\text{outflow})$, the stock $F(\text{os})(f)$ is the **downstream** stock of the flow $F(\text{of})(f)$. The injectivity of $F(\text{if})$ and $F(\text{of})$ ensure that each flow has at most one upstream stock and at most one downstream stock. Flows having an upstream stock but not a downstream stock or vice versa are called **partial flows**.

Following the ideas of Section 3.1, we can define a category $\mathsf{StockFlow}_f$ of full-fledged stock-flow diagrams. It is useful to glue together such diagrams not only along stocks but also along sum variables and sum links, for example to keep track of the total population in an epidemiological model. We can still do this using decorated cospans if we introduce the category $\mathsf{FinSet}^{\mathsf{G}}$, where G is the free category on this diagram:

$$\text{stock} \longleftarrow \text{sum link} \longrightarrow \text{sum variable}.$$

There is an evident inclusion functor $\iota\colon \mathsf{G} \to \mathsf{H}_f$, so any functor $F \in \mathsf{FinSet}^{\mathsf{H}_f}$ restricts to a functor $F \circ \iota \in \mathsf{FinSet}^{\mathsf{G}}$, and we define an **open** full-fledged stock-flow diagram to be a full-fledged stock-flow diagram (F, ϕ) equipped with a cospan



where $X, Y \in \mathsf{FinSet}^{\mathsf{G}}$. With this adjustment we can define a category $\mathsf{Open}(\mathsf{StockFlow}_f)$ of open full-fledged stock-flow diagrams following the ideas in Section 3.2. Most importantly, in analogy to Theorem 3.3, there is a functor

$$v\colon \mathsf{Open}(\mathsf{StockFlow}_f) \to \mathsf{Open}(\mathsf{Dynam})$$

providing a semantics for open full-fledged stock-flow diagrams. We have implemented full-fledged stock-flow diagrams and this semantics in our Julia package StockFlow (below)—but for simplicity, Section 4 only discusses the simpler stock-flow diagrams treated in Sections 3.1–3.3.

4 Implementing stock-flow diagrams in AlgebraicJulia

Existing tools for building stock-flow diagrams and simulating the systems they represent suffer from several limitations. In Section 2, we singled out two: an absence of compositionality, which it makes it difficult to build complex models in an intelligible manner, and a blurring of the distinction between syntax and semantics, which inhibits the reusability and interoperability of stock-flow diagrams in different contexts. In Section 3, we addressed both these problems at the mathematical level, the first by constructing a category of open stock-flow diagrams, and the second by constructing a functor from this category into a category of open dynamical systems, whose morphisms describe systems of differential equations. We now describe our implementation of these mathematical structures as new software for System Dynamics modeling.

Our software, available at `https://github.com/AlgebraicJulia/StockFlow.jl` as the open source package StockFlow, is implemented using AlgebraicJulia [1], a family of packages for applied category theory written in the Julia programming language [6]. The AlgebraicJulia ecosystem consists of Catlab, which implements many standard abstractions in category theory, and a collection of packages which apply these abstractions to specific domains of science and engineering. Most relevant to this article are AlgebraicDynamics [19], implementing open dynamical systems based on ordinary and delay differential equations, and AlgebraicPetri [18], implementing Petri nets with rates and their ODE semantics.

Existing capabilities within AlgebraicJulia, based on general category-theoretic abstractions, enable us to give an economical implementation of stock-flow diagrams. The combinatorial essence of stock-flow diagrams—what we called primitive stock-flow diagrams in Section 3.1—are set-valued functors on a certain category H, or H-sets. Such structures are encompassed by the paradigm of categorical databases, for which Catlab has extensive support [22]. Subject to one caveat, stock-flow diagrams—including the flow functions—can also be implemented as categorical databases. In this way, stock-flow diagrams become combinatorial data structures that can be manipulated algorithmically through high-level operations such as limits and colimits. Currently, Catlab has better support for structured cospans than decorated cospans. The latter have some theoretical advantages, as mentioned in Section 3.2, but luckily the two formalisms are equivalent for the tasks carried out here [4]. Thus, at present we use structured cospans to implement open stock-flow diagrams in StockFlow. We elaborate on this in the following subsections.

4.1 Stock-flow diagrams as categorical databases

In Section 3.1 we defined a primitive stock-flow diagram to be a finite H-set for a certain category H, called the **schema** for these diagrams. In Catlab, we present this schema as:

```
@present SchPrimitiveStockFlow(FreeSchema) begin
  (Stock, Flow, Link)::Ob
  (up, down)::Hom(Flow, Stock)
  src::Hom(Link, Stock)
  tgt::Hom(Link, Flow)
end
```

To define stock-flow diagrams, we need to add a data attribute for the flow functions. In general, data attributes [22] are a practical necessity and the main feature that distinguishes categorical databases from the standard mathematical notion of a C-set, meaning a functor from C to Set. In this case, we extend the schema with an attribute type and a data attribute:

```
@present SchStockFlow <: SchPrimitiveStockFlow begin
  FlowFunc::AttrType
  flow::Attr(Flow, FlowFunc)
end
```

Having defined the schema, we can generate a Julia data type for stock-flow diagrams with this single line of code:

```
@acset_type StockFlow(SchStockFlow, index=[:up, :down, :src, :tgt])
```

where the indices are generated for morphisms in the schema to enable fast traversal of stock-flow diagrams. A stock-flow diagram will then have the Julia type `StockFlow{Function}`, where `Function` is the built-in type for functions in Julia. We see that there is a gap between the mathematical definition of stock-flow diagrams and the present implementation: the domains of the flow functions should be constrained by the links, but this constraint is not yet expressible in the data model supported by Catlab. In practice this is not a major obstacle to using stock-flow diagrams, but it could motivate future work toward increasing the expressivity of database schemas and instances in Catlab.

The full-fledged stock flow diagrams described in Section 3.5 are implemented similarly.

4.2 Composition using structured cospans

While the mathematics described in Section 3 uses decorated cospans, we can also describe open stock-flow diagrams using structured cospans [3, 4]. A structured cospan is a diagram of the form

$$L(A) \xrightarrow{\;i\;} X \xleftarrow{\;o\;} L(B)$$

in some category X, where A, B are objects in some other category A, and $L\colon \mathsf{A} \to \mathsf{X}$ is a functor. When L is a left adjoint, we can equivalently think of a structured cospan as a diagram

$$A \xrightarrow{\;i\;} R(X) \xleftarrow{\;o\;} B$$

where R is the right adjoint of L.

For example, we can take $\mathsf{A} = \mathsf{FinSet}$, take $\mathsf{X} = \mathsf{FinSet}^{\mathsf{H}}$, and take $R\colon \mathsf{FinSet}^{\mathsf{H}} \to \mathsf{FinSet}$ to be the functor sending any primitive stock-flow diagram to its set of stocks. In this case, a structured cospan amounts to an **open primitive stock-flow diagram**, that is a primitive stock-flow diagram $F\colon \mathsf{H} \to \mathsf{FinSet}$ together with functions

$$A \xrightarrow{\;i\;} F(\text{stock}) \xleftarrow{\;o\;} B.$$

With more work we can define a structured cospan category equivalent to $\mathsf{Open}(\mathsf{StockFlow})$. The advantage of this change in viewpoint is that Catlab already provides a generic framework for working with structured cospans and multicospans—and it implements the composition operations available in both the hypergraph category of structured cospans and the operad algebra of structured multicospans. In addition, Catlab includes a concrete instantiation of structured cospans for systems defined by attributed C-sets. This makes it possible, for a broad class of systems, to use structured cospans in just a few lines of code. This is the approach taken in StockFlow and also its companion package AlgebraicPetri. In order to support the implementation of full-fledged stock-flow diagrams, we expanded the implementation of structured multicospans in Catlab so that the feet in a multicospan can be arbitrary C-sets as opposed to merely finite sets.

4.3 Composing epidemiological models: an example

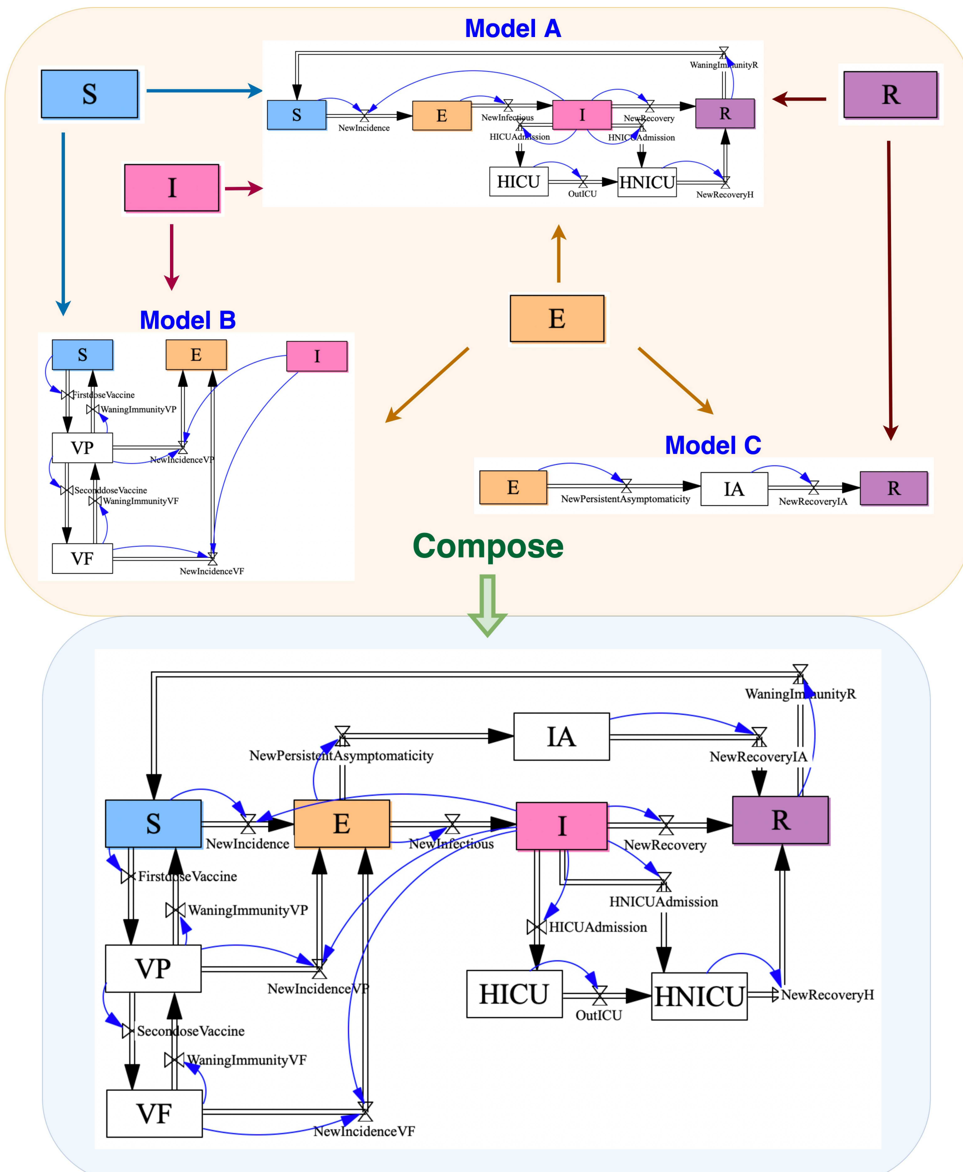


Figure 4: Example of composing a COVID-19 model from three smaller models

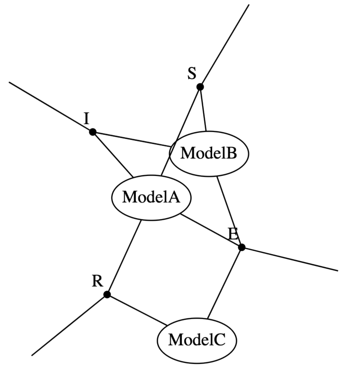


Figure 5: The undirected wiring diagram representing composing structured multicospans

The Julia package StockFlow implements both the open stock-flow diagrams of Section 3.2 and the full-fledged open stock-flow diagrams of Section 3.5. We now illustrate the use of this package by constructing a simplified version of a COVID-19 model that has has been employed during the pandemic for daily reporting and planning throughout the Province of Saskatchewan, and for weekly reporting by the Public Health Agency of Canada for each of Canada's ten provinces, as well as by First Nations and Inuit Health for reporting to provincial groupings of First Nations Reserves.

We build this simplified model as the composite of three component models: (A) a model of the natural history of infection, pathogen transmission, and hospitalization, (B) a model of vaccination, and (C) a model of the natural history of infection among asymptomatic or oligosymptomatic individuals. To exhibit the ideas with a minimum of complexity, we use the simpler open stock-flow diagrams discussed in Sections 3.1-3.4, not the full-fledged ones.

The top of Figure 4 shows the open stock-flow diagrams for three models. Although the flow functions are omitted from the figure, they are defined in the Jupyter notebook implementing this example.[1] Model (A) is the SEIRH (Susceptible-Exposed-Symptomatic Infectious-Recovered-Hospitalized) model, which simulates the disease transmission from, course of infection amongst, and hospitalization of symptomatically infected individuals. The stocks labelled "HICU" and "HNICU" represent the populations of hospitalized ICU and non-ICU patients, respectively. Model (B) characterizes vaccination-related dynamics. The stock "VP" represents individuals who are partially protected via vaccination, due to having been administered only a first dose or to waning of previously full vaccine-induced immunity. In contrast, the stock "VF" represents individuals who are fully vaccinated by virtue of having received two or more doses of the vaccine. Notably, neither partially or fully vaccinated individuals are considered fully protected from infection. Thus, there are flows from stock "VP" and "VF" to "E" that represent new infection of vaccinated individuals. Model (C) characterizes the natural history of infection in individuals who are persistently asymptomatic. The stock "IA" indicates the infected individuals without any symptoms.

[1]Readers interested in the code for this example can refer to `https://github.com/AlgebraicJulia/StockFlow.jl/blob/master/examples/primitive_schema_examples/Covid19_composition_model_in_paper.ipynb` on the GitHub repository for StockFlow.

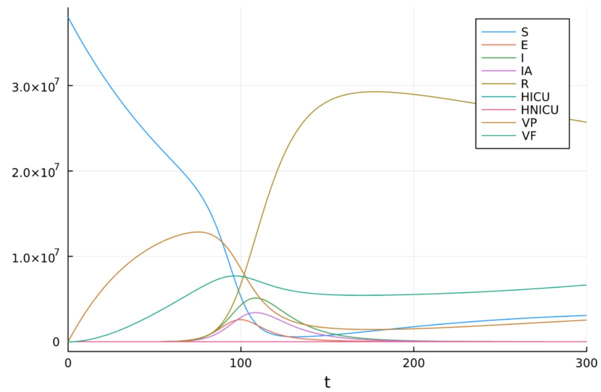


Figure 6: A simulation of the composite COVID-19 model

The ODEs finally generated from the composite stock-flow diagram are as follows:

$$\dot{S} = \frac{R}{t_w} + \frac{V_P}{t_w} - \frac{\beta SI}{N} - r_v S \qquad \dot{E} = \frac{\beta SI}{N} + \frac{\beta(1-e_p)IV_P}{N} + \frac{\beta(1-e_f)IV_F}{N} - r_{ia}E - r_i E$$

$$\dot{I} = r_i E - \frac{I}{t_r} \qquad \dot{R} = \frac{(1-f_H)I}{t_r} + \frac{I_A}{t_r} + \frac{H_{\text{NICU}}}{t_H} - \frac{R}{t_w}$$

$$\dot{I_A} = r_{ia}E - \frac{I_A}{t_r} \qquad \dot{V_F} = r_v V_P - \frac{V_F}{t_w} - \frac{\beta(1-e_f)IV_F}{N}$$

$$\dot{H}_{\text{ICU}} = \frac{f_H f_{\text{ICU}} I}{t_r} - \frac{H_{\text{ICU}}}{t_{\text{ICU}}} \qquad \dot{V_P} = r_v S + \frac{V_F}{t_w} - \frac{V_P}{t_w} - r_v V_P - \frac{\beta(1-e_p)IV_P}{N}$$

$$\dot{H}_{\text{NICU}} = \frac{H_{\text{ICU}}}{t_{\text{ICU}}} + \frac{f_H(1-f_{\text{ICU}})I}{t_r} - \frac{H_{\text{NICU}}}{t_H}$$

where for simplicity we use $1/t_r$ to stand for the rate at which infected individuals proceed to the next stage (stocks R, HICU or HNICU), and assume this is also the rate at which asymptomatic infected individuals go to the next stage (stock R). A plot of a solution of these equations is shown in Figure 6. In our software, the initial values and values of parameters are defined separately from the stock-flow diagram. This design enables the users to flexibly define and explore multiple scenarios involving the same dynamical system, in a manner similar to some existing stock-flow modeling packages. For example, the parameter values used in Figure 6 are from Canada's population. We can efficiently run this model on other populations (e.g., the United States) by changing these parameter values.[2]

This particular COVID-19 model simplifies the structure and assumptions of the model used in practice. Our example omits features such as characterization of active case-finding, diagnosis and reporting, mortality, and transmission by asymptomatic/oligosymptomatic individuals, because the simplified stock-flow diagrams do not support auxiliary variables, sum variables, and partial flows. However, our StockFlow package implements the full-fledged stock-flow diagrams defined in Section 3.5, and hence enables the application of these additional features.

[2]Readers interested in the code for this example can refer to `https://github.com/AlgebraicJulia/StockFlow.jl/blob/master/examples/primitive_schema_examples/Covid19_composition_model_in_paper.ipynb` on the GitHub repository for StockFlow.

4.4 Future work

Three lines of work are underway to extend the work described here: extending the Julia application programming interface (API), constructing a graphical user interface, and training modelers to use StockFlow.

For the first, key priorities include supporting within-diagram constants in the diagrams and allowing auxiliary variables to depend on other auxiliary variables in an acyclic fashion. Approaches are also being explored to allow hierarchical composition of diagrams and ensure consistency of the functions governing flows, in the sense of dimensional analysis.

Second, we aim to help modelers use the software without needing to know category theory. Thus, building atop the API, we are currently constructing a declarative, real-time graphical user interface (GUI) for collaboratively constructing, manipulating, composing and packaging stock-flow diagrams.

Third, we are training both students and professional modelers in the use of Stockflow, and will ramp up these efforts soon, once the GUI achieves sufficient functionality to offer practical utility. This will both build a user base and provide useful feedback as to how epidemiological modelers interact with the software.

Acknowledgments Author Osgood wishes to express his appreciation to SHA, PHAC and FNIH for support of varying elements of this work, to SYK, and to NSERC for support via the Discovery Grants program (RGPIN 2017-04647). Patterson acknowledges support from AFOSR (Award FA9550-20-1-0348). Baez and Libkind thank the Topos Institute for their support.

References

[1] AlgebraicJulia team: *AlgebraicJulia: Bringing compositionality to technical computing*. Available at https://www.algebraicjulia.org.

[2] R. M. Anderson & R. M. May (1992): *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, doi:10.1017/s0950268800059896.

[3] J. C. Baez & K. Courser (2020): *Structured cospans*. Theor. Appl. Categ. 35, pp. 1771–1822, doi:10.48550/arXiv.1911.04630. Available at arXiv:1911.04630.

[4] J. C. Baez, K. Courser & C. Vasilakopoulou (2022): *Structured versus decorated cospans*. Compositionality 4(3), doi:10.32408/compositionality-4-3. Available at arXiv:2101.09363.

[5] J. C. Baez & B. S. Pollard (2017): *A compositional framework for reaction networks*. Reviews in Mathematical Physics 29(09), p. 1750028, doi:10.1142/s0129055x17500283. Available at arXiv:1704.02051.

[6] J. Bezanson, A. Edelman, S. Karpinski & V. B. Shah (2017): *Julia: A fresh approach to numerical computing*. SIAM Review 59(1), pp. 65–98, doi:10.1137/141000671.

[7] D. Destoumieux-Garzón, P. Mavingui, G. Boetsch, J. Boissier, F. Darriet et al. (2018): *The One Health concept: 10 years old and a long road ahead*. Frontiers in Veterinary Science, p. 14, doi:10.3389/fvets.2018.00014.

[8] O. Diekmann & J. A. P. Heesterbeek (2000): *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*. 5, John Wiley & Sons, doi:10.1093/ije/30.1.186.

[9] V. Faghihi, A. R. Hessami & D. N. Ford (2015): *Sustainable campus improvement program design using energy efficiency and conservation*. Journal of Cleaner Production 107, pp. 400–409, doi:10.1016/j.jclepro.2014.12.040.

[10] B. Fong (2015): *Decorated cospans*. Theor. Appl. Categ. 30(33), pp. 1096–1120, doi:10.48550/arXiv.1502.00872. Available at arXiv:1502.00872.

[11] B. Fong & D. I. Spivak (2018): *Hypergraph categories*, doi:10.48550/arXiv.1806.08304. Available at arXiv:1305.0297.

[12] J. W. Forrester (1961): *Industrial Dynamics*. Pegasus Communications.

[13] B. Güneralp, J. D. Sterman, N. P. Repenning, R.S. Langer, J. I. Rowe & J.M. Yanni (2005): *Progress in eigenvalue elasticity analysis as a coherent loop dominance analysis tool*. In: The 23rd International Conference of The System Dynamics Society, System Dynamics Society.

[14] J. Hines: *Molecules of structure: building blocks for System Dynamics models, Version 2.02*. Available at https://vensim.com/modeling-with-molecules-2-02/.

[15] P. S. Hovmand (2014): *Community Based System Dynamics*. Springer, doi:10.1007/978-1-4614-8763-0.

[16] C. E. Kampmann (2012): *Feedback loop gains and system behavior (1996)*. System Dynamics Review 28(4), pp. 370–395, doi:10.1002/sdr.1483.

[17] W. O. Kermack & A. G. McKendrick (1927): *A contribution to the mathematical theory of epidemics*. Proceedings of the Royal Society of London. Series A 115(772), pp. 700–721, doi:10.1098/rspa.1927.0118.

[18] S. Libkind, A. Baas, M. Halter, E. Patterson & J. Fairbanks (2022): *An algebraic framework for structured epidemic modeling*. Phil. Trans. R. Soc. A. 380, p. 20210309, doi:10.1098/rsta.2021.0309. Available at arXiv:2203.16345.

[19] S. Libkind, A. Baas, E. Patterson & J. Fairbanks (2022): *Operadic modeling of dynamical systems: mathematics and computation*. EPTCS 372, pp. 192–206, doi:10.4204/eptcs.372.14. Available at arXiv:2105.12282.

[20] K. H. Lich & J. Kuhlberg (2020): *Engaging Stakeholders in Mapping and Modeling Complex Systems Structure to Inform Population Health Research and Action*. In Y. Apostolopoulos, K. H. Lich & M. K. Lemke, editors: Complex Systems and Population Health, Oxford University Press, p. 119, doi:10.1093/oso/9780190880743.003.0009.

[21] J. S. Mackenzie & M. Jeggo (2019): *The One Health approach—why is it so important?* Tropical Medicine and Infectious Disease 4(2), p. 88, doi:10.3390/tropicalmed4020088.

[22] E. Patterson, O. Lynch & J. Fairbanks (2022): *Categorical data structures for technical computing*. Compositionality 4(2), doi:10.32408/compositionality-4-5. Available at arXiv:2106.04703.

[23] B. Richmond (1985): *STELLA: Software for bringing System Dynamics to the other 98%*. In: Proceedings of the 1985 International Conference of the System Dynamics Society: 1985 International System Dynamics Conference, System Dynamics Society, pp. 706–718.

[24] R. Ross (1916): *An application of the theory of probabilities to the study of a priori pathometry—Part I*. Proceedings of the Royal Society of London. Series A 92(638), pp. 204–230, doi:10.1098/rspa.1916.0007.

[25] R. Ross & H. P. Hudson (1917): *An application of the theory of probabilities to the study of a priori pathometry—Part II*. Proceedings of the Royal Society of London. Series A 93(650), pp. 212–225, doi:10.1098/rspa.1917.0014.

[26] M. Saleh, P. Davidsen & K. Bayoumi (2005): *A comprehensive eigenvalue analysis of system dynamics models*. In: Proceedings of the International System Dynamics Conference, The System Dynamics Society, Boston, System Dynamics Society, p. 130.

[27] D. I. Spivak (2013): *The operad of wiring diagrams: formalizing a graphical language for databases, recursion, and plug-and-play circuits*, doi:10.48550/arXiv.1305.0297. Available at arXiv:1305.0297.

[28] J. D. Sterman (1994): *Learning in and about complex systems*. System Dynamics Review 10(2-3), pp. 291–330, doi:10.1002/sdr.4260100214.

[29] J. D. Sterman (2000): *Business Dynamics*. McGraw-Hill, Inc.

[30] M. A. Trecker, D. J. Hogan, C. L. Waldner, J. R. Dillon & N. D. Osgood (2015): *Revised simulation model does not predict rebound in gonorrhoea prevalence where core groups are treated in the presence of antimicrobial resistance*. Sexually Transmitted Infections 91(4), pp. 300–302, doi:10.1136/sextrans-2014-051792.

[31] J. A. M. Vennix (1996): *Group Model Building: Facilitating Team Learning Using System Dynamics*. Chichester.

Cornering Optics

Guillaume Boisseau
University of Oxford *

Chad Nester
Tallinn University of Technology †

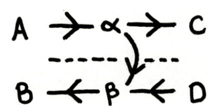
Mario Román
Tallinn University of Technology †

We show that the category of optics in a monoidal category arises naturally from the free cornering of that category. Further, we show that the free cornering of a monoidal category is a natural setting in which to work with comb diagrams over that category. The free cornering admits an intuitive graphical calculus, which in light of our work may be used to reason about optics and comb diagrams.

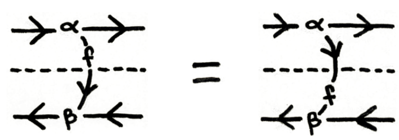
Introduction

Optics in a monoidal category are a notion of bidirectional transformation, and have been something of a hot topic in recent years. In particular *lenses*, which are optics in a cartesian monoidal category, play an important role in the theory of open games [8], compositional machine learning [5], dialectica categories [16], functional programming [17, 3], the theory of polynomial functors [21], and of course in the study of bidirectional transformations [15, 6].

We recall the elementary presentation of the category $\mathsf{Optic}_{\mathbb{A}}$ of optics in a monoidal category $\mathbb{A}$. Objects (A,B) are pairs of objects of $\mathbb{A}$. Arrows $\langle \alpha \mid \beta \rangle_M : (A,B) \to (C,D)$ consist of arrows $\alpha : A \to M \otimes C$ and $\beta : M \otimes D \to B$ of $\mathbb{A}$. It is helpful to visualize this as follows:



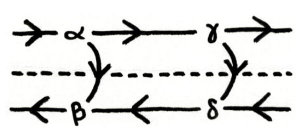
Arrows are subject to equations of the form $\langle \alpha(f \otimes 1_C) \mid \beta \rangle_N = \langle \alpha \mid (f \otimes 1_D)\beta \rangle_M$ for $f : M \to N$ in $\mathbb{A}$. This is often visualized as a sort of sliding between components, as in:



Equivalently, the hom-sets of $\mathsf{Optic}_{\mathbb{A}}$ can be given as a coend of hom-functors of $\mathbb{A}$:

$$\mathsf{Optic}_{\mathbb{A}}((A,B),(C,D)) \cong \int^M \mathbb{A}(A, M \otimes C) \times \mathbb{A}(M \otimes D, B)$$

Composition is given by $\langle \alpha \mid \beta \rangle_M \langle \gamma \mid \delta \rangle_N = \langle \alpha(1_M \otimes \gamma) \mid (1_M \otimes \delta)\beta \rangle_{M \otimes N}$. Visually:



*This research is supported by the EPSRC.

†This research was supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001).

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 97–110, doi:10.4204/EPTCS.380.6

Identity arrows are given by $1_{(A,B)} = \langle 1_A \mid 1_B \rangle_I$.

Originally studied as an approach to concurrency by Nester [14], the *free cornering* of a monoidal category is the double category obtained by freely adding companion and conjoint structure to it. The usual string diagrams for monoidal categories extend to an intuitive graphical calculus for the free cornering. The free cornering is the main piece of mathematical machinery in our development, and we give a detailed introduction to it in Section 1.

Our main contribution is a characterisation of optics in a monoidal category in terms of its free cornering. More exactly, in Theorem 1 we show that the category of optics is a full subcategory of the horizontal cells of the free cornering. In addition to shedding some light on the nature of optics, this allows us to reason about them using the graphical calculus of the free cornering. We demonstrate this by using the graphical calculus to prove Lemmas 3, 4, 5, and 6, which are a series of results originally due to Riley [18] concerning the lens laws. This occupies Section 3.

Optics in a monoidal category can be seen as a special case of *comb diagrams* in that category. Comb diagrams arose in the theory of quantum circuits [2], and have since appeared in algebraic investigations of causal structure [11, 10]. We suspect comb diagrams to be widely applicable, but there is not yet a commonly accepted algebra of comb diagrams. In Section 4 we give a notion of (single-sided) comb diagram in terms of the free cornering that coincides with the notion of comb diagram present in the work of Román [19]. We demonstrate that the free cornering is a natural setting in which to work with comb diagrams, and consider this a further contribution of the present work.

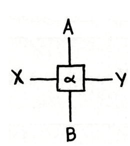
Our results are consequences of Lemma 2, which characterises cells of the free cornering with a certain boundary shape in terms of coends. In particular, we make use of the soundness result for the graphical calculus of the free cornering due to Myers [13]. The relevant definitions and the lemma itself are presented in Section 2. The reader need not be familiar with coends to follow our development. While coends connect the free cornering to the wider literature through Lemma 2, our work offers an alternate perspective that is conceptually simpler.

In summary, we give a novel characterisation of optics and comb diagrams in a monoidal category in terms of the free cornering of that category. The graphical calculus of the free cornering allows one to work with these structures more easily. In addition to telling us something about the nature of optics and comb diagrams, our results suggest that the free cornering is worthy of further study in its own right.

1 Double Categories and the Free Cornering

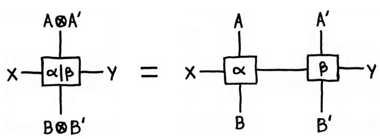
In this section we set up the rest of our development by presenting the theory of single object double categories and the free cornering of a monoidal category. In this paper we consider only *strict* monoidal categories, and in our development the term "monoidal category" should be read as "strict monoidal category". That said, we imagine that our results will hold in some form for arbitrary monoidal categories via the coherence theorem for monoidal categories [12].

A *single object double category* is a double category $\mathbb{D}$ with exactly one object. In this case $\mathbb{D}$ consists of a *horizontal edge monoid* $\mathbb{D}_H = (\mathbb{D}_H, \otimes, I)$, a *vertical edge monoid* $\mathbb{D}_V = (\mathbb{D}_V, \otimes, I)$, and a collection of *cells*

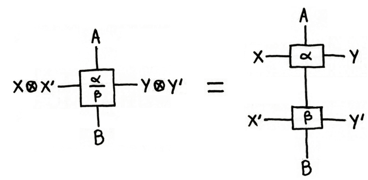


where $A, B \in \mathbb{D}_H$ and $X, Y \in \mathbb{D}_V$. We write $\mathbb{D}(X\,^{A}_{B}\,Y)$ for the *cell-set* of all such cells in $\mathbb{D}$. Given cells α, β

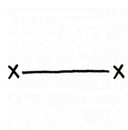
where the right boundary of α matches the left boundary of β we may form a cell $\alpha|\beta$ – their *horizontal composite* – and similarly if the bottom boundary of α matches the top boundary of β we may form $\frac{\alpha}{\beta}$ – their *vertical composite* – with the boundaries of the composite cell formed from those of the component cells using $\otimes$. We depict horizontal and vertical composition, respectively, as in:



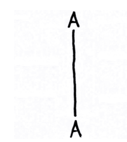
and



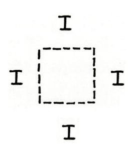
Horizontal and vertical composition of cells are required to be associative and unital. We omit wires of sort I in our depictions of cells, allowing us to draw horizontal and vertical identity cells, respectively, as in:

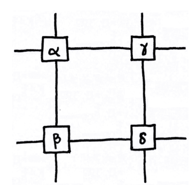


and

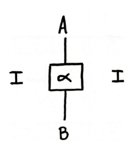


Finally, the horizontal and vertical identity cells of type I must coincide – we write this cell as $\square_I$ and depict it as empty space, see below on the left – and vertical and horizontal composition must satisfy the interchange law. That is, $\frac{\alpha}{\beta}|\frac{\gamma}{\delta} = \frac{\alpha|\gamma}{\beta|\delta}$, allowing us to unambiguously interpret the diagram below on the right:

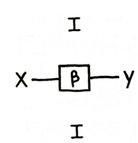




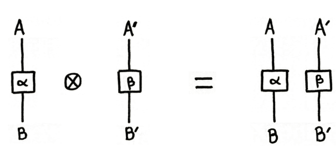
Every single object double category $\mathbb{D}$ defines strict monoidal categories $\mathbf{V}\mathbb{D}$ and $\mathbf{H}\mathbb{D}$, consisting of the cells for which the $\mathbb{D}_H$ and $\mathbb{D}_V$ valued boundaries respectively are all I, as in:



and



That is, the collection of objects of $\mathbf{V}\mathbb{D}$ is $\mathbb{D}_H$, composition in $\mathbf{V}\mathbb{D}$ is vertical composition of cells, and the tensor product in $\mathbf{V}\mathbb{D}$ is given by horizontal composition:

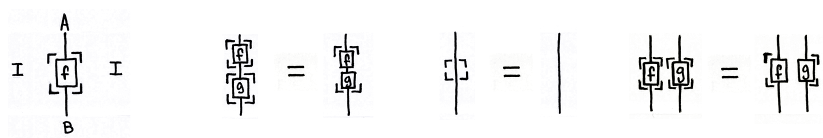


In this way, $\mathbf{V}\mathbb{D}$ forms a strict monoidal category, which we call the category of *vertical cells* of $\mathbb{D}$. Similarly, $\mathbf{H}\mathbb{D}$ is also a strict monoidal category (with collection of objects $\mathbb{D}_V$) which we call the *horizontal cells* of $\mathbb{D}$.

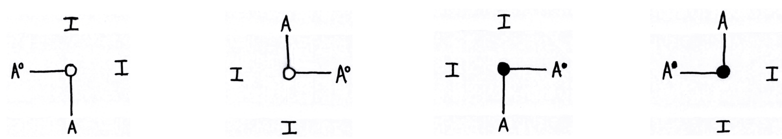
Next, we introduce the free cornering of a monoidal category.

Definition 1 ([14])**.** Let $\mathbb{A}$ be a monoidal category. We define the *free cornering* of $\mathbb{A}$, written $\ulcorner\mathbb{A}\lrcorner$, to be the free single object double category on the following data:

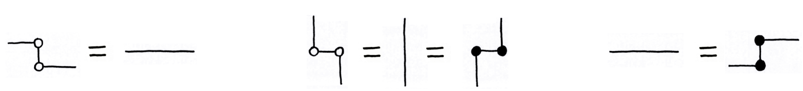
- The horizontal edge monoid $\ulcorner\mathbb{A}\lrcorner_H = (\mathbb{A}_0, \otimes, I)$ is given by the objects of $\mathbb{A}$.
- The vertical edge monoid $\ulcorner\mathbb{A}\lrcorner_V = (\mathbb{A}_0 \times \{\circ, \bullet\})^*$ is the free monoid on the set $\mathbb{A}_0 \times \{\circ, \bullet\}$ of polarized objects of $\mathbb{A}$ – whose elements we write A° and $A^\bullet$.
- The generating cells consist of vertical cells $\ulcorner f \lrcorner$ for each morphism $f : A \to B$ of $\mathbb{A}$ subject to equations as in:



along with the following *corner cells* for each object A of $\mathbb{A}$:

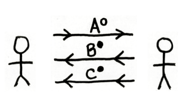


which are subject to the *yanking equations*:



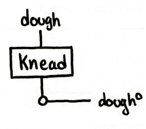
For a precise development of free double categories see [4]. Briefly, cells are formed from the generating cells by horizontal and vertical composition, subject to the axioms of a double category in addition to any generating equations. The corner structure has been heavily studied under various names including *proarrow equipment*, *connection structure*, and *companion and conjoint structure*. A good resource is the appendix of [20].

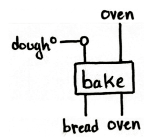
We understand elements of $\ulcorner\mathbb{A}\lrcorner_V$ as $\mathbb{A}$-*valued exchanges*. Each exchange $X_1 \otimes \cdots \otimes X_n$ involves a left participant and a right participant giving each other resources in sequence, with A° indicating that the left participant should give the right participant an instance of A, and $A^\bullet$ indicating the opposite. For example say the left participant is `Alice` and the right participant is `Bob`. Then we can picture the exchange $A^\circ \otimes B^\bullet \otimes C^\bullet$ as:

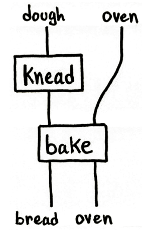
`Alice` ⇝  ⇜ `Bob`

Think of these exchanges as happening *in order*. For example the exchange pictured above demands that first `Alice` gives `Bob` an instance of A, then `Bob` gives `Alice` an instance of B, and then finally `Bob` gives `Alice` an instance of C.

Cells of $\ulcorner\mathbb{A}\lrcorner$ can be understood as *interacting* morphisms of $\mathbb{A}$. Each cell is a method of obtaining the bottom boundary from the top boundary by participating in $\mathbb{A}$-valued exchanges along the left and right boundaries in addition to using the arrows of $\mathbb{A}$. For example, if the morphisms of $\mathbb{A}$ describe processes involved in baking bread, we might have the following cells of $\ulcorner\mathbb{A}\lrcorner$:







The cell on the left describes a procedure for transforming `dough` into nothing by `kneading` it and sending the result away along the right boundary, and the cell in the middle describes a procedure for transforming an `oven` into `bread` and an `oven` by receiving `dough` along the left boundary and then using the `oven` to `bake` it. Composing these cells horizontally results in the cell on the right via the yanking equations. In this way the free cornering models concurrent interaction, with the corner cells capturing the flow of information across different components.

The vertical cells of the free cornering involve no exchanges, and as such are the cells of the original monoidal category:

Lemma 1 ([14])**.** There is an isomorphism of categories $\mathbf{V}\ulcorner\mathbb{A}\lrcorner \cong \mathbb{A}$.

In comparison, the horizontal cells of the free cornering are not well understood. In the sequel we will see that $\mathbf{H}\ulcorner\mathbb{A}\lrcorner$ contains $\mathsf{Optic}_{\mathbb{A}}$ as a full subcategory.

2 Alternation and Coends

In this section we prove a technical lemma characterizing certain cell-sets of $\ulcorner\mathbb{A}\lrcorner$ as coends.

Definition 2. An element of $\ulcorner\mathbb{A}\lrcorner_V$ is said to be $\bullet\circ$-*alternating* in case it is of the form $A_1^\bullet \otimes B_1^\circ \otimes \cdots \otimes A_n^\bullet \otimes B_n^\circ$ for some $n \in \mathbb{N}$ such that $n > 0$. The *alternation length* of a $\bullet\circ$-alternating element is defined to be the evident $n \in \mathbb{N}$. For example:

- $B^\bullet \otimes A^\circ$ is $\bullet\circ$-alternating with alternation length 1.
- $A^\bullet \otimes B^\circ \otimes C^\bullet \otimes A^\circ$ is $\bullet\circ$-alternating with alternation length 2.
- $(A \otimes B)^\bullet \otimes I^\circ$ is $\bullet\circ$-alternating with alternation length 1.
- None of the following are $\bullet\circ$-alternating:

$$I \qquad A^\bullet \otimes B^\circ \otimes C^\circ \qquad A^\bullet \otimes B^\bullet \qquad A^\bullet \qquad (A \otimes B)^\circ \otimes B^\bullet \qquad A^\bullet \otimes B^\circ \otimes C^\bullet$$

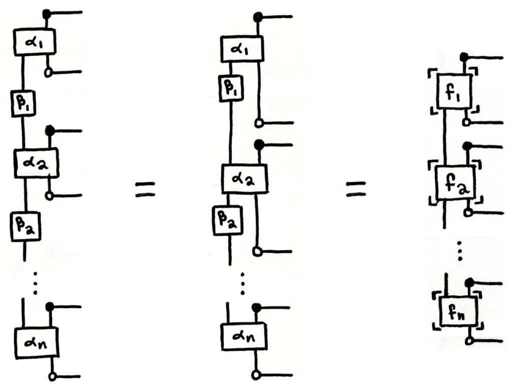
Definition 3. A cell-set of the form $\ulcorner\mathbb{A}\lrcorner({}_I{}^I_I X)$ is said to be *right-$\bullet\circ$-alternating* in case X is $\bullet\circ$-alternating. The *alternation depth* of a right-$\bullet\circ$-alternating cell-set is the alternation length of its right boundary.

Lemma 2. If $\ulcorner\mathbb{A}\lrcorner({}_I{}^I_I X)$ is right-$\bullet\circ$-alternating with alternation depth n and $X = A_1^\bullet \otimes B_1^\circ \otimes \cdots \otimes A_n^\bullet \otimes B_n^\circ$ then

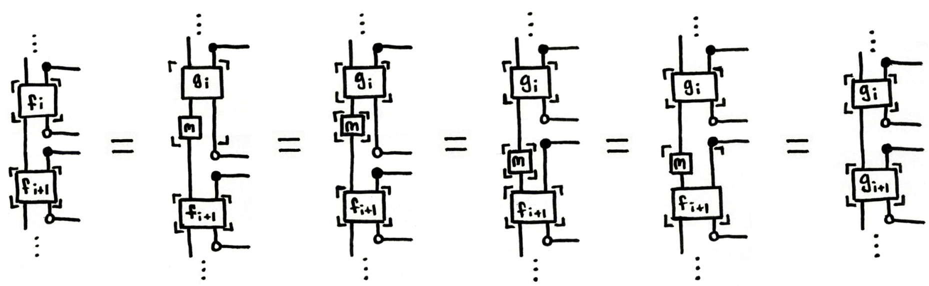
$$\ulcorner\mathbb{A}\lrcorner\left({}_I{}^I_I X\right) \cong \int^{M_1,\ldots,M_{n-1}} \prod_{i=1}^{n} \mathbb{A}(M_{i-1} \otimes A_i, M_i \otimes B_i)$$

where $M_0 = M_n = I$.

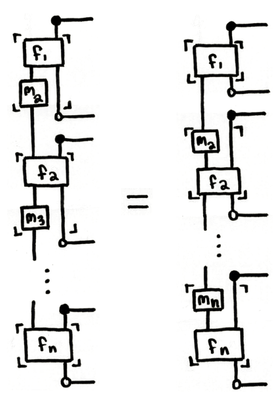
Proof. By inspecting the generating cells of $\ulcorner\mathbb{A}\urcorner$ and making use of Lemma 1 we find that any cell of $\ulcorner\mathbb{A}\urcorner(I{}^{I}_{I}X)$ is necessarily of the form:



Thus cells of $\ulcorner\mathbb{A}\urcorner(I{}^{I}_{I}X)$ may be written as n-tuples $\langle f_1 \mid \cdots \mid f_n \rangle$. As a consequence of Myers' soundness result for the graphical calculus [13], we know that two cells $\langle f_1 \mid \cdots \mid f_n \rangle$ and $\langle g_1 \mid \cdots \mid g_n \rangle$ of $\ulcorner\mathbb{A}\urcorner(I{}^{I}_{I}X)$ are equal iff they are deformable into each other modulo the equations of $\mathbb{A}$. Consider that all *local* deformations $\langle \cdots \mid f_i \mid f_{i+1} \mid \cdots \rangle = \langle \cdots \mid g_i \mid g_{i+1} \mid \cdots \rangle$ are of the form:



where $f_i = g_i(m \otimes 1)$ and $g_{i+1} = (m \otimes 1)f_{i+1}$. Now, the only way $\langle f_1 \mid \cdots \mid f_n \rangle$ and $\langle g_1 \mid \cdots \mid g_n \rangle$ can be equal is by (repeated) parallel local deformation of the associated diagrams, as in:



Thus, $\ulcorner\mathbb{A}\urcorner(I{}^{I}_{I}X)$ is the set of (appropriately typed) n-tuples $\langle f_1 \mid \cdots \mid f_n \rangle$ of morphisms of $\mathbb{A}$, quotiented by equations of the form:

$$\langle f_1(m_2 \otimes 1) \mid f_2(m_3 \otimes 1) \mid \cdots \mid f_n \rangle = \langle f_1 \mid (m_2 \otimes 1)f_2 \mid \cdots \mid (m_n \otimes 1)f_n \rangle$$

which is precisely to say that the claim holds. □

Remark 1. There is an obvious dual notion of *left-○●-alternating* cell-set for which a version of Lemma 2 holds.

3 Optics and the Free Cornering

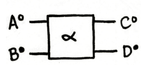
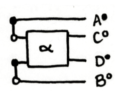
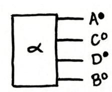
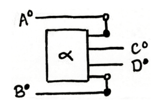
In this section we use Lemma 2 to show that $\mathsf{Optic}_{\mathbb{A}}$ is a full subcategory of $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ for any monoidal category $\mathbb{A}$. We then briefly discuss lenses, and illustrate the power of the graphical calculus for $\ulcorner\mathbb{A}\urcorner$ by reproving a correspondence between lenses satisfying the the lens laws and lenses that are comonoid homomorphisms with respect to a certain comonoid structure. These results about lenses are originally due to Riley [18], and were also used to demonstrate Boisseau's approach to string diagrams for optics [1]. We end with Observation 1, which discusses the relation of teleological categories [9] to the free cornering.

Theorem 1. Let $\mathbb{A}$ be a monoidal category. Then $\mathsf{Optic}_{\mathbb{A}}$ is the full subcategory of $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ on objects of the form $A^\circ \otimes B^\bullet$ for $A, B \in \mathbb{A}_0$.

Proof. We begin by noticing that

$$\mathbf{H}\ulcorner\mathbb{A}\urcorner(A^\circ \otimes B^\bullet, C^\circ \otimes D^\bullet) \cong \ulcorner\mathbb{A}\urcorner\big(I \overset{I}{\underset{I}{\,}} {\scriptstyle A^\bullet \otimes C^\circ \otimes D^\bullet \otimes B^\circ}\big)$$

via:


$\mapsto$

and

$\mapsto$


This cell-set is right-●○-alternating of depth 2, and so we have:

$$\ulcorner\mathbb{A}\urcorner\left(I \overset{I}{\underset{I}{\,}} {\scriptstyle A^\bullet \otimes C^\circ \otimes D^\bullet \otimes B^\circ}\right) \cong \int^{M\in\mathbb{A}} \mathbb{A}(A, M \otimes C) \times \mathbb{A}(M \otimes D, B)$$

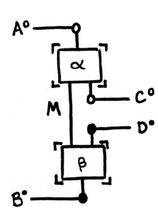
Now we already know that

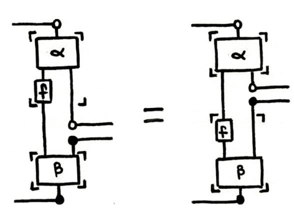
$$\int^{M\in\mathbb{A}} \mathbb{A}(A, M \otimes C) \times \mathbb{A}(M \otimes D, B) \cong \mathsf{Optic}_{\mathbb{A}}((A,B),(C,D))$$

and so we have a correspondence between arrows of $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ and arrows of $\mathsf{Optic}_{\mathbb{A}}$:

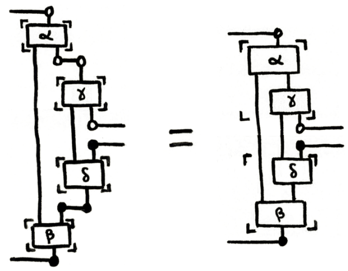
$$\mathbf{H}\ulcorner\mathbb{A}\urcorner(A^\circ \otimes B^\bullet, C^\circ \otimes D^\bullet) \cong \mathsf{Optic}_{\mathbb{A}}((A,B),(C,D))$$

In particular, we know that arrows in $\mathbf{H}\ulcorner\mathbb{A}\urcorner(A^\circ \otimes B^\bullet, C^\circ \otimes D^\bullet)$ are equivalently optics $\langle \alpha \mid \beta \rangle_M$ as below left, and that the equations between optics – below right – capture all equations in $\mathbf{H}\ulcorner\mathbb{A}\urcorner(A^\circ \otimes B^\bullet, C^\circ \otimes D^\bullet)$:

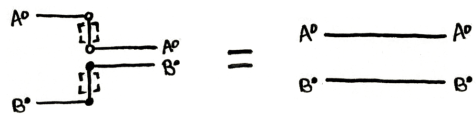




Next, given arrows $\langle\alpha \mid \beta\rangle_M : (A,B) \to (C,D)$ and $\langle\gamma \mid \delta\rangle_N : (C,D) \to (E,F)$ of $\mathsf{Optic}_{\mathbb{A}}$, we find that composing the corresponding arrows of $\mathbf{H}{\ulcorner \atop \llcorner}\mathbb{A}{\urcorner \atop \lrcorner}$ yields the arrow corresponding to $\langle\alpha(1_M \otimes \gamma) \mid (1_M \otimes \delta)\beta\rangle_{M\otimes N} = \langle\alpha \mid \beta\rangle_M \langle\gamma \mid \delta\rangle_N$ as in:



Further, the identity on $A^\circ \otimes B^\bullet$ in $\mathbf{H}{\ulcorner \atop \llcorner}\mathbb{A}{\urcorner \atop \lrcorner}$ corresponds to the $1_{(A,B)} = \langle 1_A \mid 1_B\rangle_I$ in $\mathsf{Optic}_{\mathbb{A}}$ as in:

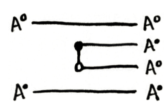


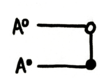
The result is thus proven. □

Remark 2. Following Remark 1, a similar argument gives that if A is symmetric monoidal then $\mathbf{H}{\ulcorner \atop \llcorner}\mathbb{A}{\urcorner \atop \lrcorner}^{op}$ also contains $\mathsf{Optic}_{\mathbb{A}}$ as the full subcategory on those objects of the form $A^\bullet \otimes B^\circ$.

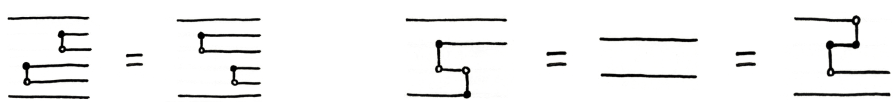
Remark 3. If $\mathbb{A}$ is a *symmetric* monoidal category then $\mathsf{Optic}_{\mathbb{A}}$ is itself monoidal [18]. We remark that while $\mathsf{Optic}_{\mathbb{A}}$ remains a subcategory of $\mathbf{H}{\ulcorner \atop \llcorner}\mathbb{A}{\urcorner \atop \lrcorner}$ in this case, it is not a *monoidal* subcategory. That is, the tensor product of optics is *not* given by the tensor product in $\mathbf{H}{\ulcorner \atop \llcorner}\mathbb{A}{\urcorner \atop \lrcorner}$.

As an illustration of our approach, we consider the characterisation of the lens laws given in [18]. Say that an optic is *homogeneous* in case it is contained in the full subcategory of $\mathsf{Optic}_{\mathbb{A}}$ on objects (A,A) for some $A \in \mathbb{A}_0$. Notice that every object of this subcategory is a comonoid in $\mathbf{H}{\ulcorner \atop \llcorner}\mathbb{A}{\urcorner \atop \lrcorner}$, with the comultiplication and counit given as in:





where the comonoid axioms hold as in:

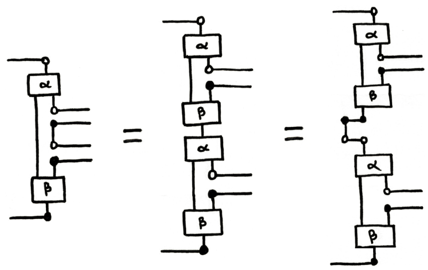


Definition 4 ([18])**.** A homogeneous optic $h : (A,A) \to (B,B)$ of $\mathsf{Optic}_{\mathbb{A}}$ is called *lawful* in case the following equations hold in $\mathbf{H}{\ulcorner \atop \llcorner}\mathbb{A}{\urcorner \atop \lrcorner}$:

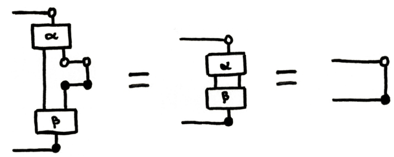
That is, in case h is a comonoid homomorphism with respect to the comonoid structure given above.

Lemma 3 ([18])**.** If $h = \langle\alpha \mid \beta\rangle_M : (A,A) \to (B,B)$ in $\mathsf{Optic}_{\mathbb{A}}$ with α and β mutually inverse, then h is lawful.

Proof.



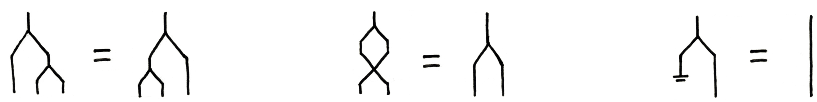
and



□

Recalling the algebraic characterisation of cartesian monoidal categories [7], we denote the commutative comonoid structure in a cartesian monoidal category as follows:

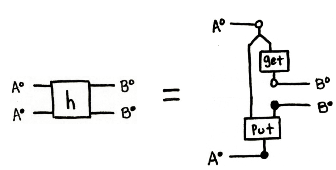
This structure must satisfy the commutative comonoid axioms:



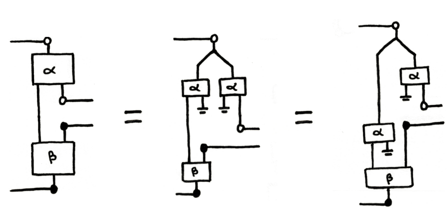
Must further be coherent with respect to the monoidal structure:

And every morphism f of the category in question must be a comonoid homomorphism:

Lemma 4 ([18])**.** Let $\mathbb{A}$ be a cartesian monoidal category, and let $h = \langle \alpha \mid \beta \rangle_M : (A, A) \to (B, B)$ be a homogeneous optic in $\mathbb{A}$. Then there exist arrows get : $A \to B$ and put : $A \otimes B \to A$ of $\mathbb{A}$ such that:



Proof. We have:

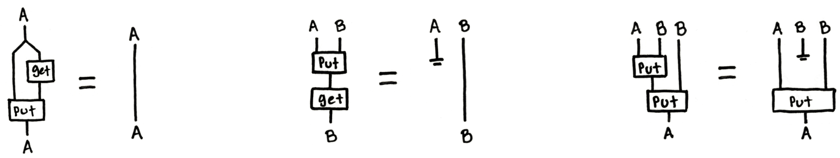


and so the claim follows via:

□

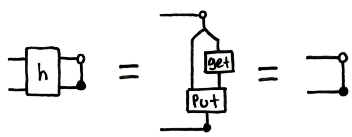
Homogeneous optics in cartesian monoidal categories are called *lenses*. We write [put | get] : $(A,A) \to (B,B)$ for the lens specified by appropriate put and get arrows in the above manner.

Definition 5 ([6])**.** A lens [put | get] : $(A,A) \to (B,B)$ is is said satisfy the *lens laws* in case:

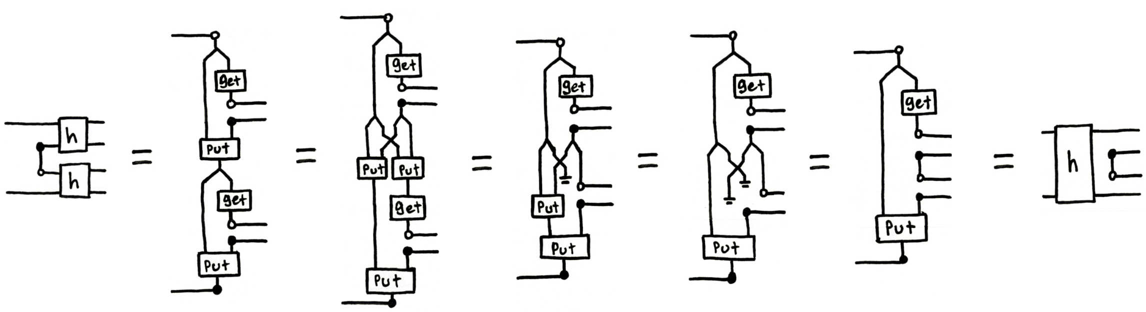


Lemma 5 ([18])**.** If a lens $h =$ [put | get] : $(A,A) \to (B,B)$ satisfies the lens laws then it is lawful.

Proof. For the counit we have:



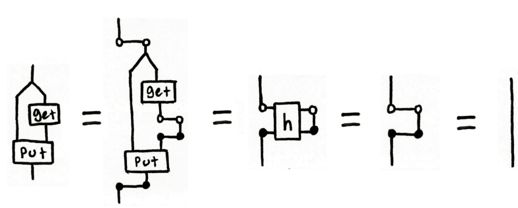
And for the comultiplication:



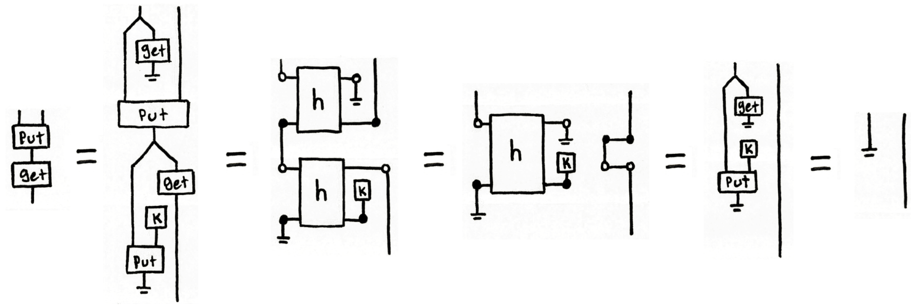
□

Lemma 6 ([18])**.** If a lens $h =$ [put | get] : $(A,A) \to (B,B)$ is lawful and B is inhabited in the sense that there is an arrow $k : 1 \to B$ in $\mathbb{A}$, then it satisfies the lens laws.

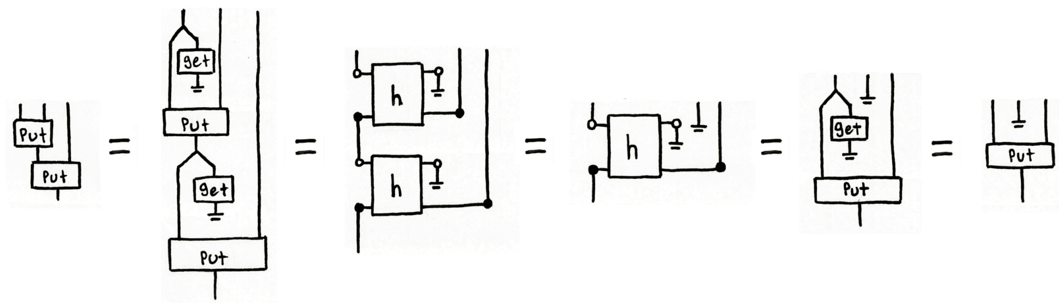
Proof. The first lens law holds as in:



The second lens law holds as in:

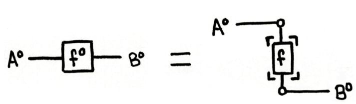


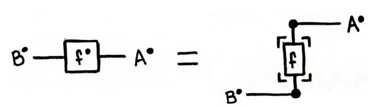
and the third lens law holds as in:



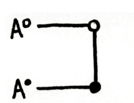
□

Observation 1 (Teleological Categories). $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ contains structure reminiscent of *teleological categories* [9], which were introduced to allow well-founded diagrammatic reasoning about lenses. Analogous to the *dualizable* morphisms of a teleological category are those of the form f°, defined as below left, with duals $f^\bullet$, defined as below right:

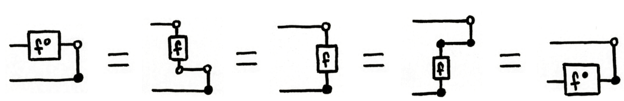




Standing in for the *counits* of a teleological category we have the following cell for each $A \in \mathbb{A}$:



We then obtain an analogue of the condition that the counits be *extranatural* as in:



Notice that all arrows $A^\circ \to B^\circ$ of $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ are of the form f° for some $f : A \to B$ in $\mathbb{A}$ and that dually all arrows $B^\bullet \to A^\bullet$ are of the form $f^\bullet$, further characterising our analogue of the dualizable morphisms.

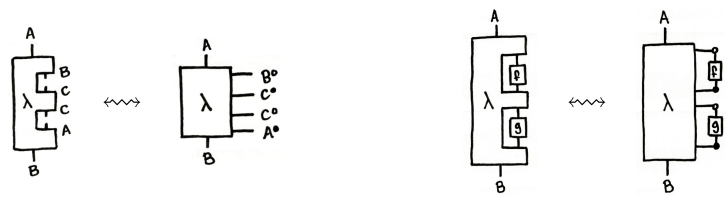
In light of this, we suggest that teleological categories are a shadow of the fact that A° is formally left adjoint to $A^\bullet$ in $\mathbf{H}\ulcorner\mathbb{A}\urcorner$. We also point out that teleological categories do not contain enough of the relevant structure to prove Lemmas 5 and 6, which require the unit of the formal adjunction between A° and $A^\bullet$ as well as the counit.

4 Comb Diagrams

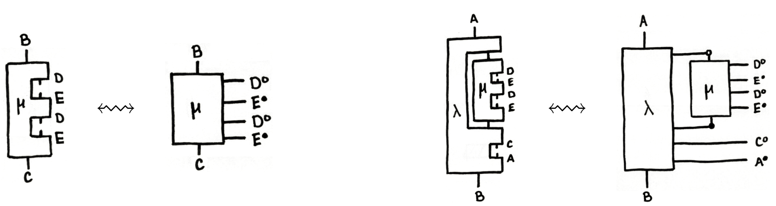
In this section we discuss comb diagrams in the free cornering. The basic idea is that we would like to have *higher-order* diagrams for our monoidal categories, pictured below on the left. Supplying the appropriate first-order string diagrams to a higher-order diagram results in a first-order diagram, pictured below on the right:

These higher-order diagrams have been called *(right) comb diagrams* due to their appearance.

In the free cornering of a monoidal category $\mathbb{A}$, elements of right-•○-alternating cell-sets are a good notion of right comb diagram, with the alternation depth corresponding to the number of gaps between the teeth:



Lemma 2 tells us that this notion of comb diagram coincides with the notion of comb diagram developed by Román in the more general framework of open diagrams [19]. The free cornering admits common comb diagram operations beyond inserting morphisms into the gaps. First, we may insert a comb diagram into one of the gaps to form another comb diagram:



Next, following Remarks 1 and 2 there is an dual notion of *left comb diagrams* in the free cornering corresponding to the left-○•-alternating cell-sets. In certain cases it makes sense to compose a right comb diagram with a left comb diagram by interleaving their teeth. The free cornering supports this as well:

Thus, the free cornering is a natural setting in which to work with comb diagrams in a monoidal category.

References

[1] G. Boisseau (2020): *String Diagrams for Optics*. In Zena M. Ariola, editor: *5th International Conference on Formal Structures for Computation and Deduction, FSCD 2020, June 29-July 6, 2020, Paris, France (Virtual Conference)*, *LIPIcs* 167, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 17:1–17:18, doi:10.4230/LIPIcs.FSCD.2020.17.

[2] G. Chiribella, G. M. D'Ariano & P. Perinotti (2008): *Quantum Circuit Architecture*. *Physical Review Letters* 101(6), doi:10.1103/physrevlett.101.060401.

[3] B. Clarke, D. Elkins, J. Gibbons, F. Loregiàn, B. Milewski, E. Pillmore & M. Román (2020): *Profunctor optics, a categorical update*. Available at `https://arxiv.org/abs/2001.07488`.

[4] M. Fiore, S. Paoli & D. Pronk (2008): *Model Structures on the Category of Small Double Categories*. *Algebraic and Geometric Topology* 8(4), pp. 1855–1959, doi:10.2140/agt.2008.8.1855.

[5] B. Fong, D. Spivak & R. Tuyéras (2019): *Backprop as functor: A compositional perspective on supervised learning*. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, pp. 1–13, doi:10.1109/LICS.2019.8785665.

[6] J.N. Foster, M.B. Greenwald, J.T. Moore, B.C. Pierce & A. Schmitt (2005): *Combinators for Bi-Directional Tree Transformations: A Linguistic Approach to the View Update Problem*. In Jens Palsberg & Martín Abadi, editors: *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2005, Long Beach, California, USA, January 12-14, 2005*, ACM, pp. 233–246, doi:10.1145/1040305.1040325.

[7] T. Fox (1976): *Coalgebras and cartesian categories*. *Communications in Algebra* 4(7), pp. 665–667, doi:10.1080/00927877608822127.

[8] N. Ghani, J. Hedges, V. Winschel & P. Zahn (2018): *Compositional Game Theory*. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, Association for Computing Machinery, New York, NY, USA, p. 472–481, doi:10.1145/3209108.3209165.

[9] J. Hedges (2017): *Coherence for lenses and open games*. Available at `https://arxiv.org/abs/1704.02230`.

[10] B. Jacobs, A. Kissinger & F. Zanasi (2021): *Causal inference via string diagram surgery: A diagrammatic approach to interventions and counterfactuals*. *Math. Struct. Comput. Sci.* 31(5), pp. 553–574, doi:10.1017/S096012952100027X.

[11] A. Kissinger & S. Uijlen (2017): *A categorical semantics for causal structure*. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–12, doi:10.1109/LICS.2017.8005095.

[12] S. Mac Lane (1971): *Categories for the Working Mathematician*. Springer, doi:10.1007/978-1-4612-9839-7.

[13] D.J. Myers (2016): *String Diagrams For Double Categories and Equipments*. Available at `https://arxiv.org/abs/1612.02762`.

[14] C. Nester (2021): *The Structure of Concurrent Process Histories*. In: *International Conference on Coordination Models and Languages*, pp. 209–224, doi:10.1007/978-3-030-78142-2_13.

[15] F.J. Oles (1982): *A Category-Theoretic Approach to the Semantics of Programming Languages*. Ph.D. thesis, Syracuse University, USA.

[16] V. de Paiva & J.W. Gray (1989): *The dialectica categories*. *Categories in Computer Science and Logic* 92, pp. 47–62.

[17] M. Pickering, J. Gibbons & N. Wu (2017): *Profunctor Optics: Modular Data Accessors*. *Art Sci. Eng. Program.* 1(2), p. 7, doi:10.22152/programming-journal.org/2017/1/7.

[18] M. Riley (2018): *Categories of Optics*. Available at `https://arxiv.org/abs/1809.00738`.

[19] M. Román (2021): *Open Diagrams via Coend Calculus*. *Electronic Proceedings in Theoretical Computer Science* 333, p. 65–78, doi:10.4204/eptcs.333.5.

[20] M. Shulman (2008): *Framed Bicategories and Monoidal Fibrations*. Theory and Applications of Categories 20(18), pp. 650–738. Available at `http://www.tac.mta.ca/tac/volumes/20/18/20-18abs.html`.

[21] D. Spivak & N. Niu (2022): *Polynomial Functors: A General Theory of Interaction*.

Data Structures for Topologically Sound Higher-Dimensional Diagram Rewriting

Amar Hadzihasanovic
[1] Tallinn University of Technology
[2] Quantinuum, 17 Beaumont Street, Oxford, UK
amar@ioc.ee

Diana Kessler
Tallinn University of Technology
diana-maria.kessler@taltech.ee

We present a computational implementation of diagrammatic sets, a model of higher-dimensional diagram rewriting that is "topologically sound": diagrams admit a functorial interpretation as homotopies in cell complexes. This has potential applications both in the formalisation of higher algebra and category theory and in computational algebraic topology. We describe data structures for well-formed shapes of diagrams of arbitrary dimensions and provide a solution to their isomorphism problem in time $O(n^3 \log n)$. On top of this, we define a type theory for rewriting in diagrammatic sets and provide a semantic characterisation of its syntactic category. All data structures and algorithms are implemented in the Python library `rewalt`, which also supports various visualisations of diagrams.

Introduction

This article concerns the computational implementation of higher-dimensional diagrams in the sense of higher category theory, and contains some first steps in the computational complexity theory of diagrammatic rewriting in arbitrary dimensions.

Higher-dimensional rewriting, as emergent from the theory of polygraphs [5] – see [12] for a survey – is founded on an interpretation of *rewrites as directed homotopies*. A particular aim of our work is provable *topological soundness*, namely, the existence of a functorial interpretation of rewrite systems as cell complexes, and of rewrites as homotopies. This ensures that our implementation of higher-dimensional rewriting can act as a formal system for homotopical algebra and higher category theory in all generality.

With this aim, we turn to the *diagrammatic set* model [13] developed by the first author as a combinatorial alternative to polygraphs. Diagrammatic sets have a dual nature as higher-dimensional rewrite systems and "combinatorial directed cell complexes". They support a model of weak higher categories and, unlike polygraphs, are topologically sound.

Beside the formalisation of higher algebra and category theory, potential applications are manifold. *String diagram rewriting*, which is a form of 3-dimensional rewriting, is arguably the characteristic computational mechanism of applied category theory. It has been suggested [4] that even "classical" forms of rewriting are more faithfully represented as diagram rewriting: for example, term rewriting implemented as rewriting in monoidal categories with cartesian structure explicitates the "hidden costs" of copying and deleting terms. In these contexts, it is important to have a grasp on the computational complexity of the basic operations of diagram rewriting, to ensure that one's cost model for a machine operating by diagram rewriting is reasonable.

Via topological soundness, we also envisage applications to computational algebraic topology. Directedness of cells gives an *algebraic* grip on their pasting, which lends itself better to computation. Directed cell complexes are also equipped with an orientation on their cells, which makes them naturally suited to the computation of cellular homology.

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 111–127, doi:10.4204/EPTCS.380.7

Structure of the paper

In Section 1, we present some basic data structures from the theory of diagrammatic sets, together with their formal encoding: in particular, *oriented graded posets* which are used to encode shapes of diagrams.

In Section 2, we focus on the implementation of *regular molecules*, the inductive subclass of oriented graded posets corresponding to well-formed shapes of diagrams. To construct regular molecules, we need to decide their isomorphism problem; for general oriented graded posets, this is equivalent to the graph isomorphism problem (Proposition 2.11), not known to be in P. Our main result is a solution to the isomorphism problem for regular molecules in time $O(n^3 \log n)$ (Theorem 2.19), which also gives us a canonical form, hence a unique representation of shapes of diagrams.

In Section 3, we move on to the formalisation of diagrams and diagrammatic sets. We present this in the form of a type theory DiagSet living "on top" of our implementation of shapes of diagrams: the terms, corresponding to diagrams, are "filtered by regular molecules". This allows us to define formal semantics and give a semantic characterisation of our formal system (Theorem 3.10).

Related work

A number of type theories for higher-categorical structures of arbitrary dimension have been defined in recent years: most notably, Finster and Mimram's CaTT [8], implementing the Maltsiniotis model of weak higher categories [3], together with its "strictly associative" [10] and "strictly unital" [9] variants; and the *opetopic* type theories by Ho Thanh, Curien, and Mimram [15, 6].

The former are not particularly concerned with diagram rewriting, and focus instead on the implementation of coherent globular composition; the link to our work is tenuous. The latter have some commonality, albeit with a focus on a more restrictive class of shapes. In fact, DiagSet takes some inspiration not from one of the published opetopic type theories, but from a privately communicated variant due to Curien, which similarly rests on a "black-boxed" implementation of opetopic shapes.

Most closely related is the work by Vicary, Bar, Dorn, and others on quasistrict [2] and later associative [7, 17] n-categories, serving as the foundation of the homotopy.io proof assistant. While the aim is nearly the same, we believe that our framework has a number of advantages over associative n-categories.

From a theoretical perspective, it is only conjectural that associative n-categories, in general, are topologically sound or satisfy the homotopy hypothesis. They also currently lack connections with other models of higher categories and a clear functorial viewpoint. On the other hand, diagrammatic sets are topologically sound, satisfy a version of the homotopy hypothesis, and support a model of weak higher categories with concrete functorial ties to well-established models.

From a user perspective, the main point of divergence is that diagrams in associative n-categories have "strict units" but "weak interchange", while our diagrams have "strict interchange" but need weak units to model "nullary" inputs or outputs. For rewrite systems with many "nullary" generators, associative n-categories may have a practical advantage, while diagrammatic sets are otherwise favoured.

Finally, in associative n-categories, diagram shapes are essentially descriptions of cubical tilings, and by lack of strict interchange, each rewrite gets by default its own "layer" in the tiling. This makes it so a "local" rewrite on a portion of a diagram leads to an inefficient "global" duplication of information. Our "face poset" representation of diagrams, on the other hand, allows local rewrites to stay local, which is more efficient and will be beneficial to the parallelisability of diagram rewriting.

Implementation

All data structures, algorithms, and systems discussed in this article were implemented by the authors as part of a Python library for higher-dimensional rewriting and algebra, called `rewalt`.[1] An example of `rewalt` code is included in Example 3.13. The library also supports various kinds of visualisation for diagrams, optionally in the form of TikZ output. All the Hasse and string diagrams in this article were generated by `rewalt` and included here with no subsequent retouching.

Acknowledgements

This work was supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001) and by the Estonian Research Council grant PSG764.

1 Basic data structures

1.1. In the theory of diagrammatic sets, the shape of a pasting diagram is encoded by its *face poset*, recording whether a cell is located in the boundary of another cell, together with *orientation* data which specifies whether an $(n-1)$-dimensional cell is in the *input* or *output* half of the boundary of an n-dimensional cell. We call the mathematical structure containing these data an *oriented graded poset*. This is essentially the same as what Steiner calls a *directed precomplex* [18] and Forest an *ω-hypergraph* [11].

1.2 (Graded poset)**.** Let P be a finite poset with order relation $\leq$ and let $P_\perp$ be P extended with a least element $\perp$. We say that P is *graded* if, for all $x \in P$, all directed paths from x to $\perp$ in the Hasse diagram $\mathscr{H}P_\perp$, with edges going from covering to covered elements, have the same length. If this length is $n+1$, we let $\dim(x) := n$ be the *dimension* of x. We write P_n for the subset of n-dimensional elements of P.

1.3 (Oriented graded poset)**.** An *orientation* on a finite poset P is an edge-labelling of its Hasse diagram with values in $\{+,-\}$. An *oriented graded poset* is a finite graded poset with an orientation.

Implementation 1.4. If we linearly order the elements of an oriented graded poset in each dimension, each element x is uniquely identified by a pair of integers (n,k), where n is the *dimension* of x, and k is the *position* of x in the linear ordering of n-dimensional elements.
We then represent an oriented graded poset as a pair $(\mathsf{face\_data}, \mathsf{coface\_data})$ of *arrays of arrays of pairs of sets of integers*, where

1. $j \in \mathsf{face\_data}[n][k][i]$ if and only if $(n-1, j)$ is covered by (n,k), and
2. $j \in \mathsf{coface\_data}[n][k][i]$ if and only if $(n+1, j)$ covers (n,k)

with orientation $-$ $(i=0)$ or $+$ $(i=1)$. We may implement the sets of integers as sorted arrays, or another data type which supports binary search in logarithmic time. This defines a data type OgPoset.

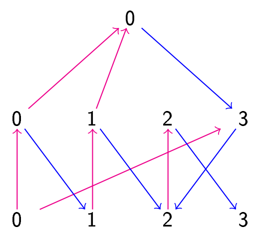
This representation is essentially an adjacency list representation of the poset's Hasse diagram, with vertices separated according to their dimension, and incoming and outgoing edges separated according to their label. If E_P is the set of edges of the Hasse diagram of P, the OgPoset representation of P takes space $O(|P| + |E_P|)$.

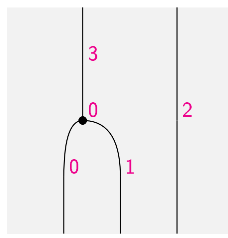
Storing both face_data and coface_data is redundant since these are uniquely determined by each other. However, most of the computations we need to perform on oriented graded posets require regular access both to faces (covered elements) and cofaces (covering elements) of a given element, so it is advantageous to be able to access them in constant time.

[1] Code: `https://github.com/ahadziha/rewalt`. Documentation: `https://rewalt.readthedocs.io`.

Example 1.5. Consider a diagram formed of one 2-cell with two input 1-cells and a single output 1-cell, whiskered to the right with a single 1-cell. The following are representations of its shape as

- an oriented face poset, pictured as a Hasse diagram with input faces pointing upwards (in magenta) and output faces downwards (in blue);
- a string diagram (0-cells are unlabelled, but correspond to bounded regions of the plane);
- the pair of face_data and coface_data (rows are outer array indices and columns inner array indices).





face_data:

([],[])	([],[])	([],[])	([],[])
([0],[1])	([1],[2])	([2],[3])	([0],[2])
([0,1],[3])			

coface_data:

([0,3],[])	([1],[0])	([2],[1,3])	([],[2])
([0],[])	([0],[])	([],[])	([],[0])
([],[])			

Remark 1.6. The representation of an oriented graded poset (up to isomorphism) is not unique: any permutation of the linear order on elements in each dimension leads to an equivalent representation.

1.7. Many important computations are performed on *(downwards) closed subsets*, rather than the whole of an oriented graded poset. In particular, the structure of an oriented graded poset supports a purely combinatorial definition of the input and output boundary of a closed subset.

1.8 (Closed subsets)**.** Let P be an oriented graded poset and $U \subseteq P$. We say that U is *closed* if, for all $y \in U$ and $x \in P$, if $x \leq y$ then $x \in U$. The *closure* of U is the subset $\mathrm{cl}U := \{x \in P \mid \exists y \in U \; x \leq y\}$.

We let $\dim(U)$ be the maximum of $\dim(x)$ for $x \in U$, or -1 if U is empty.

1.9 (Input and output boundaries)**.** Let P be an oriented graded poset and $U \subseteq P$ a closed subset. For all $\alpha \in \{+,-\}$ and $n \in \mathbb{N}$, let

- $\Delta_n^\alpha U \subseteq U$ be the subset of elements x such that $\dim(x) = n$ and, if $y \in U$ covers x, then it covers it with orientation α;
- $\mathscr{M}_n U \subseteq U$ be the subset of elements x such that $\dim(x) = n$ and x is maximal in U (not covered by any other element of U).

The *input* ($\alpha := -$) or *output* ($\alpha := +$) *n-boundary* of U is the closed subset

$$\partial_n^\alpha U := \mathrm{cl}\Big(\Delta_n^\alpha U \cup \bigcup_{k<n} \mathscr{M}_k U\Big).$$

We let $\partial_n U := \partial_n^+ U \cup \partial_n^- U$ and omit n when $n = \dim(U) - 1$. For all $x \in P$, we let $\partial_n^\alpha x := \partial_n^\alpha \mathrm{cl}\{x\}$.

Remark 1.10. It is convenient to also let $\partial_{-1}^\alpha U = \partial_{-2}^\alpha U := \emptyset$, so that $\partial^\alpha U$ is defined for all $U \subseteq P$.

Example 1.11. Let U be the oriented face poset of Example 1.5. Then

$$\partial_1^- U = \{(0,0),(0,1),(0,2),(0,3),(1,0),(1,1),(1,2)\},$$
$$\partial_1^+ U = \{(0,0),(0,2),(0,3),(1,2),(1,3)\},$$
$$\partial_0^- U = \{(0,0)\}, \qquad \partial_0^+ U = \{(0,3)\}.$$

Implementation 1.12. We represent a set of elements of an OgPoset as an *array of sets of positions, indexed by dimensions*. This allows us to access the subset of elements of a given dimension in constant time. The size of arrays can be fixed to be equal to the dimension of a specific OgPoset, or dynamically adjusted to the dimension of each set of elements. Sets of positions can again be implemented as sorted arrays. This defines a data type GrSet (for *graded set*).

1.13 (Map of oriented graded posets)**.** A *map* $f\colon P \to Q$ of oriented graded posets is a function of their underlying sets that satisfies $\partial_n^\alpha f(x) = f(\partial_n^\alpha x)$ for all $x \in P$, $n \in \mathbb{N}$, and $\alpha \in \{+,-\}$. We call an injective map an *inclusion*. Oriented graded posets and their maps form a category **ogPos**.

Example 1.14. A closed subset of an oriented graded poset inherits the structure of an oriented graded poset by restriction. Its subset inclusion is an inclusion of oriented graded posets.

Implementation 1.15. We represent a map $f\colon P \to Q$ as an *array of arrays of pairs of integers* mapping, together with pointers source, target to OgPoset representations of P and Q. This defines a data type OgMap. As an array of arrays, mapping has the same size of P's face_data, and is defined by

$$\mathsf{mapping}[n][k] = (m, j) \text{ if and only if } f((n,k)) = (m, j).$$

This representation takes space $O(|P|)$.

2 Unique representation of shapes of diagrams

2.1. In the theory of diagrammatic sets, shapes of diagrams form an inductively generated class of oriented graded posets, called regular *molecules* after Steiner [18].

2.2 (Round subset)**.** Let U be a closed subset of an oriented graded poset, $n := \dim(U)$. We say that U is *round* if, for all $k < n$,

$$\partial_k^+ U \cap \partial_k^- U = \partial_{k-1} U.$$

Remark 2.3. Roundness is called "spherical boundary" in [13].

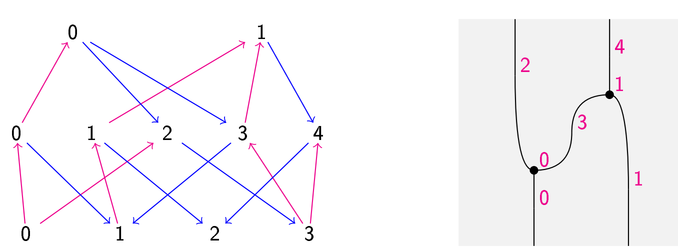
Example 2.4. Shapes of 2-dimensional diagrams, as oriented face posets, are round precisely when

1. their string diagram representation is *connected*, and
2. all nodes of the string diagram have at least one input and one output wire.

For example, the oriented graded poset of Example 1.5 is not round: we have

$$\partial_0 U = \{(0,0),(0,3)\} \subsetneq \partial_1^+ U \cap \partial_1^- U = \{(0,0),(0,2),(0,3)\}.$$

On the other hand, the following oriented graded poset is round:



2.5 (Regular molecules)**.** The class of regular molecules is generated by the following clauses.

- (Point). The terminal oriented graded poset $\bullet$ is a regular molecule.
- (Atom). Let U, V be *round* regular molecules such that $\dim(U) = \dim(V)$ and, for all $\alpha \in \{+,-\}$, $\partial^\alpha U$ is isomorphic to $\partial^\alpha V$. Then $U \Rightarrow V$ is a regular molecule, where $U \Rightarrow V$ is the essentially unique oriented graded poset $U \Rightarrow V$ with the property that
  1. $U \Rightarrow V$ has a greatest element, and
  2. $\partial^-(U \Rightarrow V)$ is isomorphic to U, while $\partial^+(U \Rightarrow V)$ is isomorphic to V.
- (Paste). Let U, V be regular molecules and $k < \min(\dim(U), \dim(V))$, such that $\partial_k^+ U$ is isomorphic to $\partial_k^- V$. Then the pushout $U \#_k V$ of the span $\partial_k^+ U \hookrightarrow U$, $\partial_k^+ U \overset{\sim}{\hookrightarrow} \partial_k^- V \hookrightarrow V$ is a regular molecule.

A regular molecule is an *atom* if it has a greatest element; these are precisely the molecules whose final generating clause is (Point) or (Atom).

The *submolecule* relation $U \sqsubseteq V$ is the preorder generated by $U, V \sqsubseteq U \Rightarrow V$ and $U, V \sqsubseteq U \#_k V$.

Comment 2.6. The properties of regular molecules are explored in [13, Sections 1, 2]. Importantly, the following results ensure that §2.5 is a valid definition:

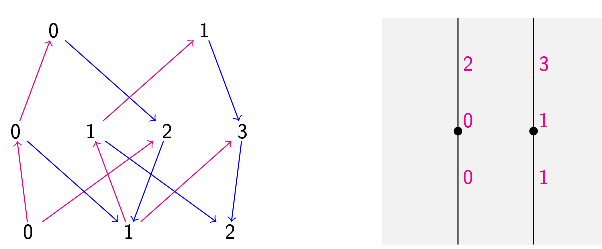
1. the category **ogPos** has pushouts of inclusions;
2. if U and V are isomorphic regular molecules, they are isomorphic *in a unique way*;
3. input and output boundaries of regular molecules are regular molecules;
4. if U and V are round, then a pair of isomorphisms between $\partial^\alpha U$ and $\partial^\alpha V$ for $\alpha \in \{+,-\}$ extends uniquely to an isomorphism between ∂U and ∂V.

The first three imply that $U \#_k V$ is well-defined and does not depend on a choice of isomorphism between $\partial_k^+ U$ and $\partial_k^- V$. The fourth implies that $U \Rightarrow V$ can be uniquely constructed by extending the isomorphisms $\partial^\alpha U \overset{\sim}{\hookrightarrow} \partial^\alpha V$ to an isomorphism $\partial U \overset{\sim}{\hookrightarrow} \partial V$, then gluing U and V along this isomorphism, and finally adding a greatest element with the appropriate orientation.

Example 2.7. Let $\mathsf{arrow} := (\bullet \Rightarrow \bullet)$ and $\mathsf{binary} := ((\mathsf{arrow} \#_0 \mathsf{arrow}) \Rightarrow \mathsf{arrow})$. The shape of the diagram of Example 1.5 is generated as $\mathsf{binary} \#_0 \mathsf{arrow}$, while the oriented graded poset of Example 2.4 is generated as $(\mathsf{cobinary} \#_0 \mathsf{arrow}) \#_1 (\mathsf{arrow} \#_0 \mathsf{binary})$, where $\mathsf{cobinary} := (\mathsf{arrow} \Rightarrow (\mathsf{arrow} \#_0 \mathsf{arrow}))$.

Remark 2.8. As discussed in [13, §2.1], the pasting constructions $- \#_k -$ satisfy the equations of composition in strict ω-categories *up to unique isomorphism*. It follows that the "same" regular molecule may be constructed in different ways. For example, letting $\mathsf{globe} := (\mathsf{arrow} \Rightarrow \mathsf{arrow})$, we have

$$(\mathsf{globe} \#_0 \mathsf{arrow}) \#_1 (\mathsf{arrow} \#_0 \mathsf{globe}) \;\simeq\; \mathsf{globe} \#_0 \mathsf{globe} \;\simeq\; (\mathsf{arrow} \#_0 \mathsf{globe}) \#_1 (\mathsf{globe} \#_0 \mathsf{arrow}).$$



Implementation 2.9. We want to implement regular molecules as a subtype Shape of OgPoset with a nullary constructor point and partial binary constructors $\mathsf{atom}(-,-)$ and $\mathsf{paste}_k(-,-)$ for $k \in \mathbb{N}$. In order to implement the constructors, we need to be able to perform the following operations:

1. compute input and output k-boundaries;

2. check if a closed subset is round;
3. determine if two regular molecules are isomorphic;
4. compute the pushout of a span of inclusions.

The first, second, and fourth of these admit straightforward algorithms of low-degree polynomial time complexity, that do not rely on any special properties of regular molecules. The third problem, however, is non-trivial. Indeed, the isomorphism problem generalised to all oriented graded posets is equivalent to the *graph isomorphism* (GI) problem, which is not known to be in P; the best known algorithm, due to Babai, runs in quasipolynomial time [1].

Remark 2.10. As customary in this context, a graph is a *simple* graph (no loops or multiple edges).

Proposition 2.11 — *The isomorphism problem for oriented graded posets is* GI*-complete.*

Proof. Deciding isomorphism of oriented graded posets is equivalent to deciding isomorphism of their Hasse diagrams with $\{+,-\}$-labelled edges. The isomorphism problem for edge-labelled finite graphs is an instance of the isomorphism problem for finite relational structures, which is GI-complete [16].

Conversely, a directed graph can be represented by its "oriented incidence poset": the 0-dimensional elements are the vertices, the 1-dimensional elements are the edges, the only input face of an edge is its source, and the only output face of an edge is its target. Two directed graphs are isomorphic if and only if their oriented incidence posets are isomorphic. Since GI reduces to the isomorphism problem for directed graphs, it reduces to the isomorphism problem for 1-dimensional oriented graded posets. ■

Nevertheless, in the special case of regular molecules, we can do much better. Our strategy is to describe a deterministic *traversal algorithm*, where the traversal order depends only on the intrinsic structure of a regular molecule as an oriented graded poset and not on its representation.

Given U, V : OgPoset representing regular molecules, we traverse both U and V, and then *reorder* their elements in each dimension according to their traversal order. If U', V' : OgPoset are the reordered versions of U, V, we then have

$$U \simeq V \text{ if and only if } U' \equiv V'.$$

We will show that, with this strategy, we can solve the isomorphism problem for regular molecules in time $O(n^3 \log n)$. A more precise upper bound is given in Theorem 2.19 below.

In addition to solving the isomorphism problem for regular molecules, the traversal order gives us a canonical form for regular molecules in OgPoset form. If we implement the constructors of Shape in such a way that they always produce an OgPoset in traversal order, we obtain that

$$\text{for all } U, V : \mathsf{Shape},\ U \simeq V \text{ if and only if } U \equiv V,$$

that is, we have a *unique representation* for shapes of diagrams.

The algorithm is described in Figure 1. At each iteration of the main loop (line 4), the current state is fully described by the *stack* – including its top element, the *focus* – and by the list of *marked* elements.

Lemma 2.12 — *Let V be an item on the stack. Then V is a regular molecule. If W is below V on the stack, then V is a proper subset of W.*

Proof. Initially, the stack only contains U, which is a regular molecule by assumption. Assume, inductively, that the statement is true at the beginning of the current iteration with focus V, and that a set V' is pushed onto the stack at the end. Then either

1. $V' = \partial^\alpha V$ for some $\alpha \in \{+,-\}$, or

```
procedure TRAVERSE(U : regular molecule)
    marked ← []
    stack ← [U]
    while stack is not empty do
        focus ← top of stack
        dim ← dim(focus)
        if focus ⊆ marked then
            pop focus from top of stack
        else
            if ∂⁻focus ⊈ marked then
                push ∂⁻focus to top of stack
            else
                if focus = cl{x} for some x then
                    append x to marked
                    pop focus from top of stack
                    if ∂⁺focus ⊈ marked then
                        push ∂⁺focus on top of stack
                else
                    y ← first item of dimension dim − 1 in marked such that
                          y has an unmarked input coface in focus
                    x ← unique input coface of y in focus
                    push cl{x} on top of stack
    return marked
```

Figure 1: The traversal algorithm.

2. $V' = \mathrm{cl}\{x\}$ for some $x \in V$.

In both cases, V' is a regular molecule and a proper subset of V (hence also of each item below V), under the assumption that V is a regular molecule. ∎

Remark 2.13. In fact, any V that appears on the stack is either $\partial_k^- U$, which we call "U-linked", or it is $\mathrm{cl}\{x\}$ or $\partial_k^\alpha x$, which we call "x-linked", for some $x \in U$. In the latter case, V is *round*, which implies that it is also *pure* [13, Lemma 1.35]: its maximal elements all have the same dimension.

Lemma 2.14 — *Suppose V is on the stack. Then all elements of V must be marked before any item below V is accessed, or before any proper superset of V becomes the focus.*

Proof. By Lemma 2.12, as long as V is on the stack, only V and its proper subsets can be on top. It follows that, for a proper superset of V to be the focus, V must be popped from the stack at the end of an iteration where V is the focus. There are only two ways this can happen:

- V was already fully marked before the current loop iteration, or
- $\partial^- V$ was fully marked and $V = \mathrm{cl}\{x\}$ for some x which is marked at the current loop iteration.

In both cases, $\partial^- V$ was already fully marked before the current loop iteration. In the latter case, if $\partial^+ V$ is already fully marked, then $V = \{x\} \cup \partial^- V \cup \partial^+ V$ is also fully marked. Otherwise, $\partial^+ V \subsetneq V$ gets pushed onto the stack to replace V, and must be popped before any superset of V becomes the focus. By the same case distinction, whenever $\partial^+ V$ is popped, either

- it was fully marked, in which case V was fully marked, or

- it is of the form $\text{cl}\{y\}$ for some y which is marked at the current loop iteration.

Either way, since all regular molecules satisfy the *globularity* property $\partial^{\alpha}(\partial^{+}V) = \partial^{\alpha}(\partial^{-}V) \subseteq \partial^{-}V$, we know that $\partial^{+}V$, hence V, is fully marked at the end of the iteration, and nothing is added to the stack. ■

Lemma 2.15 — *Any subset V of U can be pushed onto the stack at most once.*

Proof. Suppose V is pushed onto the stack. As long as V is on the stack, any subsequent addition to the stack must be a proper subset of V, so it cannot be equal to V.

If V is popped from the stack, by Lemma 2.14, it must be fully marked before any item below it is accessed. Since the algorithm checks if a set is fully marked before pushing it onto the stack, V can never appear again. ■

Lemma 2.16 — *Let V be the focus, $n := \dim(V)$. Then either V is fully marked, or there exists an n-dimensional element of V which is unmarked.*

Proof. First, we prove a weaker result: either V is fully marked, or there exists a *maximal* element of V which is unmarked.

Let $x \in V$ be marked. At some prior iteration, $\text{cl}\{x\}$ must have been the focus, and by Lemma 2.14, in order for V to become the focus, $\text{cl}\{x\}$ must have been fully marked as well. Because

$$V = \bigcup_{k \leq n} \text{cl}\mathscr{M}_k V = \bigcup_{k \leq n} \bigcup_{x \in \mathscr{M}_k V} \text{cl}\{x\},$$

it follows that V is fully marked if and only if its maximal elements are all marked.

Now, V has one of the two forms in Remark 2.13. If V is of the second form, its maximal elements all have the top dimension, so we only need to consider the case $V = \partial_k^{-}U$.

At the start of the algorithm, $U, \ldots, \partial_0^{-}U$ are all consecutively added to the stack. So $\partial_k^{-}U$ becomes the focus either at this stage, in which case *all* its elements are unmarked, or after $\partial_{k-1}^{-}U$ is fully marked. In the latter case, any maximal element of $\partial_k^{-}U$ of dimension strictly smaller than k also belongs to $\partial_{k-1}^{-}U$. ■

Theorem 2.17 — *The traversal algorithm is correct: given a regular molecule U, it terminates returning a unique linear ordering of the elements of U.*

Proof. As a particular case of Lemma 2.14, U must be fully marked before the stack is emptied. Therefore, the algorithm either terminates after all elements have been traversed, or it does not terminate.

To prove that the algorithm does always terminate, it suffices to show that, unless all elements are already marked, it always finds an element to mark. First of all, observe that, from any state, the algorithm first goes through the following sequence of steps:

1. popping all fully marked subsets from the top of the stack;
2. once it reaches a subset which is not fully marked, successively pushing its lower-dimensional input boundaries that are not fully marked onto the stack.

At the end of this sequence, we always reach a state in which the focus V is not fully marked, but $\partial^{-}V$ is fully marked. Let us call such a V a *proper* focus.

We proceed by induction on dimension and proper subsets of a proper focus. If $\dim(V) = 0$, since a 0-molecule always consists of a single element, $V = \{x\}$, and x gets marked at the current iteration.

Let $n := \dim(V)$. By Lemma 2.16, there is an unmarked $x \in V_n$. If $V = \mathrm{cl}\{x\}$, then x is marked at the current iteration, and we are done. Otherwise, we prove that there always exists a pair (y,x) where $x \in V_n$ is unmarked, and y is a marked input face of x. By [13, Lemma 1.16] applied to V, the coface x is unique given y, so among such pairs we can pick the one where y comes *earliest* in the list of marked elements, and this selects a unique x.

Let $x \in V_n$ be unmarked. By a dual version of [*ibid.*, Lemma 1.37], there exists a sequence

$$y_0 \to x_0 \to \ldots \to y_m \to x_m = x$$

where $y_0 \in \Delta^-_{n-1} V$, $x_i \in V_n$, y_i is an input face of x_i, and y_{i+1} is an output face of x_i. Since V is a proper focus, y_0 is marked. Let k be the smallest index such that x_k is unmarked; because x_m is unmarked, such a k exists. Then x_i is marked for all $i < k$, hence $\mathrm{cl}\{x_i\}$ is also marked. It follows that $y_k \in \partial^+ x_{k-1}$ is marked, and the pair (y_k, x_k) satisfies our requirement.

Thus, the algorithm will find a unique $x \in V_n$ and push $\mathrm{cl}\{x\}$ onto the stack. The next proper focus will necessarily be a proper subset of V, and we conclude by the inductive hypothesis. ∎

2.18. In what follows, for a fixed regular molecule U, we let $|E_n|$ be the number of edges between n and $(n-1)$-dimensional elements in the Hasse diagram of U, and we let

$$|U_{\max}| := \max_n |U_n|, \qquad |E_{\max}| := \max_n |E_n|.$$

Theorem 2.19 — *The traversal algorithm admits an implementation running in time*

$$O\big(|U|^2(|E_{\max}| \cdot \log|E_{\max}| + |U_{\max}| \cdot \log|U_{\max}|)\big).$$

Proof. First of all, we represent any closed set on the stack with its graded set of maximal elements. To initialise the algorithm, we only need to compute the maximal elements of U. This can be done in time $O(|U|)$ by going through the elements of U and checking if their set of cofaces is empty.

Next, let us find an upper bound for the number of iterations of the main loop (line 4). Let V be a set on the stack, $n := \dim(V)$. Then V can become the focus

- at most once before pushing $\partial^- V$ onto the stack (line 11),
- at most once before pushing $\mathrm{cl}\{y\}$ onto the stack for each $y \in V_n$ (line 22), and
- at most once to be popped from the stack (line 8),

after which, by Lemma 2.15, it can never appear again. Thus, the number of loop iterations with V as focus is bounded by $|V_n| + 2$.

By Remark 2.13, every set V on the stack is either "U-linked" or "x-linked" for some $x \in U$. There are $(\dim(U)+1)$ many U-linked focusses and $(2\dim(x)+1)$ many x-linked focusses. Then

- the number of loop iterations with U-linked focusses is bounded by $|U| + 2\dim(U) + 2$, and
- for each x, the number of iterations with x-linked focusses is bounded by $|\mathrm{cl}\{x\}| + 4\dim(x) + 2$.

Since there are $|U|$ elements, $|\mathrm{cl}\{x\}| \leq |U|$, and $\dim(x) \leq \dim(U)$, we have a coarse upper bound of $(|U|+1)(|U| + 4\dim(U) + 2)$ on the total number of iterations, which is $O(|U|^2)$.

Next, in our implementation, we split the list of marked elements into three objects: a list `order` (for the total traversal order), an array of lists `grorder` (for the traversal order split by dimension), and a graded set `marked` (for the set of marked elements).

Consider a single loop iteration with focus V, $n := \dim(V)$.

(Line 7). By Lemma 2.16, to check if V is fully marked, it suffices to check whether $V_n \subseteq \mathsf{marked}_n$. Since both are sorted arrays of integers, they can be compared in time linear in $|V_n| + |\mathsf{marked}_n|$, which is $O(|U_n|)$. At this stage, we may also record the unmarked n-dimensional elements of V in a sorted array unmarked without affecting the complexity.
(Line 10). To compute the maximal elements of $\partial^- V$ and $\partial^+ V$, we may use different strategies depending on whether V is "U-linked" or not.

If $V = \partial_n^- U$, we compute the $(n-1)$-dimensional elements of $\partial^- V = \partial_{n-1}^- U$ simply by going through the elements of U_{n-1} and checking which ones have empty sets of output cofaces, in time $O(|U_{n-1}|)$. Lower-dimensional maximal elements are shared between V and $\partial^- V$, so we may then point from the latter to the former, at no extra cost.

If V is not U-linked, V and its boundaries are pure, so the set of maximal elements of $\partial^\alpha V$ is equal to $\Delta^\alpha V$, and each of its elements is covered by an element of V_n. To compute it, we add all the input and output faces of all $x \in V_n$ to sets in_faces and out_faces, respectively, then use the relations $\Delta^- V = \mathsf{in\_faces} \setminus \mathsf{out\_faces}$ and $\Delta^+ V = \mathsf{out\_faces} \setminus \mathsf{in\_faces}$.

There are $O(|E_n|)$ faces of elements of V_n, and we can sort in_faces and out_faces, remove duplicates, and compute their difference in time $O(|E_n| \cdot \log |E_n|)$.

At this stage, we also create an associative array candidates as follows: whenever $x \in V_n$ is in unmarked, and y is an input face of x, we add the position of x as a value to candidates, indexed by the position of y. We then sort the indices of candidates. This also takes time $O(|E_n| \cdot \log |E_n|)$ so it does not affect the overall complexity.
(Lines 10, 16). By the same reasoning applied to line 7, checking if $\partial^- V$ and $\partial^+ V$ are fully marked takes time $O(|U_{n-1}|)$.
(Line 14). If V_n has a single element that we mark, adding it to order and grorder takes constant time with an appropriate implementation of lists. Adding it to marked takes $O(|U_n|)$.
(Lines 19—21). To select the next focus we traverse $\mathsf{grorder}_{n-1}$ starting from the first item and search for each item in the indices of candidates until we find a hit y. This takes time $O(|U_{n-1}| \cdot \log |U_{n-1}|)$ in the worst case. The next focus will be $\mathrm{cl}\{x\}$, where x is the value corresponding to index y.

Overall, the worst-case complexity is $O(|U_n| + |E_n| \cdot \log |E_n| + |U_{n-1}| \cdot \log |U_{n-1}|)$. Using the bounds $|U_n|, |U_{n-1}| \leq |U_{\max}|$ and $|E_n| \leq |E_{\max}|$, and multiplying by our bound on the number of iterations, we conclude. ■

3 A type theory for higher-dimensional rewriting

3.1. We rapidly go through the definitions of diagrammatic sets and some related notions. For a thorough treatment, we refer to [13, Section 4 and onwards], and to [14, Section V] for diagrammatic complexes as presentations of higher-dimensional theories.

3.2 (Diagrammatic set)**.** Let $\odot$ (to be read *atom*) be a skeleton of the full subcategory of **ogPos** on the atoms of every dimension. A *diagrammatic set* is a presheaf on $\odot$. Diagrammatic sets and their morphisms of presheaves form a category $\odot$**Set**.

3.3. We identify $\odot$ with a full subcategory $\odot \hookrightarrow \odot\mathbf{Set}$ via the Yoneda embedding. With this identification, we use morphisms in $\odot$**Set** as our notation for both elements and structural operations of a diagrammatic set X:

- $x \in X(U)$ becomes $x \colon U \to X$, and
- for each map $f \colon V \to U$ in $\odot$, $X(f)(x) \in X(V)$ becomes $f;x \colon V \to X$.

The embedding $\odot \hookrightarrow \odot\mathbf{Set}$ extends along pushouts of inclusions to the full subcategory of **ogPos** on the regular molecules.

3.4 (Diagrams and cells)**.** Let X be a diagrammatic set and U a regular molecule. A *diagram of shape U in X* is a morphism $x\colon U \to X$. A diagram is a *cell* if U is an atom. For all $n \in \mathbb{N}$, we say that x is an *n-diagram* or an *n-cell* when $\dim(U) = n$.

If U decomposes as $U_1 \#_k U_2$, we write $x = x_1 \#_k x_2$ for $x_i := \iota_i;x$, where ι_i is the inclusion $U_i \hookrightarrow U$ for $i \in \{1,2\}$. Let $\iota_k^\alpha\colon \partial_k^\alpha U \hookrightarrow U$ be the inclusions of the k-boundaries of U. The *input k-boundary* of x is the diagram $\partial_k^- x := \iota_k^-;x$ and the *output k-boundary* of x is the diagram $\partial_k^+ x := \iota_k^+;x$. We write $x\colon y^- \Rightarrow y^+$ to express that $\partial_k^\alpha x = y^\alpha$ for each $\alpha \in \{+,-\}$.

3.5 (Diagrammatic complex)**.** For each $n \in \mathbb{N}$, let $\odot_n$ be the full subcategory of $\odot$ on the atoms of dimension $\leq n$, and let $\odot_{-1}$ be the empty subcategory. The restriction functor $\odot\mathbf{Set} \to \mathrm{PSh}(\odot_n)$ has a left adjoint; let $\sigma_{\leq n}$ be the comonad induced by this adjunction. The *n-skeleton* of a diagrammatic set X is the counit $\sigma_{\leq n}X \to X$. For all $k \leq n$, the k-skeleton factors uniquely through the n-skeleton of X.

A *diagrammatic complex* is a diagrammatic set X together with a set $\mathscr{X} = \sum_{n\in\mathbb{N}} \mathscr{X}_n$ of *generating* cells such that, for all $n \in \mathbb{N}$,

$$\begin{array}{ccc} \coprod_{x\in\mathscr{X}_n} \partial U(x) & \hookrightarrow & \coprod_{x\in\mathscr{X}_n} U(x) \\ \downarrow (\partial x)_{x\in\mathscr{X}_n} & & \downarrow (x)_{x\in\mathscr{X}_n} \\ \sigma_{\leq n-1}X & \hookrightarrow & \sigma_{\leq n}X \end{array}$$

is a pushout in $\odot\mathbf{Set}$, where $U(x)$ denotes the shape of x. A diagrammatic complex is *finite* if $\mathscr{X}$ is finite.

3.6 (Support-based diagrammatic complex)**.** Each cell in a diagrammatic complex $(X, \mathscr{X})$ is uniquely of the form $(p\colon U \twoheadrightarrow V, x\colon V \to X)$, where p is a surjective map of atoms and $x \in \mathscr{X}$. We let $\mathsf{supp}(p,x) := x$, the *support* of (p,x).

A *support-based diagrammatic complex* is the quotient of a diagrammatic complex by the relations

$$x \sim y \text{ if and only if } \mathsf{supp}(\iota;x) = \mathsf{supp}(\iota;y) \text{ for all inclusions of atoms } \iota\colon V \hookrightarrow U, \tag{1}$$

for all atoms U and cells $x,y\colon U \to X$. We let $\odot\mathbf{Cpx}_{fsb}$ denote the category of finite, support-based diagrammatic complexes with morphisms of their underlying diagrammatic sets.

3.7. We define a dependent type theory for diagrammatic sets – more precisely, for finite, support-based diagrammatic complexes – that relies on an underlying unique representation of regular molecules and their maps, treated as a "black box". Of course, in the previous section we have provided such an implementation and proved that it is computationally feasible. Nevertheless, it is useful to separate its abstract properties from the implementation details.

3.8 ($\mathsf{DiagSet}$)**.** Let $\mathbb{V}$ be an infinite set of variables. We define a type theory $\mathsf{DiagSet}$ as follows.
Terms. A term t is a pair of a regular molecule U, the *shape* of t, and a function $t\colon U \to \mathbb{V}$. We write $t/_U$ to express that t is a term of shape U. Maps $p\colon U \to V$ act on terms by precomposition: if $t/_V$ is a term, then $p^*t := (p;t)/_U$. In particular, we let $\partial_k^\alpha t := (\iota_k^\alpha;t)/_{\partial_k^\alpha V}$ for all $k \in \mathbb{N}$ and $\alpha \in \{+,-\}$.
Types. A type A is either $\varnothing$ or an expression $t \Rightarrow s$ where t,s are terms. We may annotate a term t of shape U with the type $A := \varnothing$ if $U \equiv \bullet$, and $A := \partial^- t \Rightarrow \partial^+ t$ otherwise.
Contexts. A context Γ is a list $x_1 : A_1, \ldots, x_n : A_n$ of typed variables. We consider two contexts to be equal if they are equal up to a permutation. If $x : A$ is a typed variable, we say that x has *shape* $\bullet$ if $A \equiv \varnothing$, and $U \Rightarrow V$ if $A \equiv t/_U \Rightarrow s/_V$. We write $x/_U : A$ to express that $x : A$ has shape U.

Substitutions. A substitution σ is a list $x_1 \mapsto t_1, \ldots, x_n \mapsto t_n$ of assignments of terms to variables. We consider two substitutions to be equal if they are equal up to a permutation.

Judgments. We consider three kinds of judgments:

- $\Gamma \vdash$ meaning that Γ is a well-formed context,
- $\Gamma \vdash t$ meaning that t is a well-formed term in context Γ, and
- $\Delta \vdash \sigma : \Gamma$ meaning that σ is a well-formed substitution from context Δ to context Γ.

The inference rules of $\mathsf{DiagSet}$ are the following. We use $\langle\rangle$ to indicate the empty list.

Rules for contexts.

$$\frac{}{\langle\rangle \vdash}\,\mathsf{init} \qquad \frac{\Gamma \vdash}{\Gamma, x : \varnothing \vdash}\,\mathsf{pt} \qquad \frac{\Gamma \vdash t/_U : r^- \Rightarrow r^+ \quad \Gamma \vdash s/_V : r^- \Rightarrow r^+ \quad U, V \text{ round}}{\Gamma, x : t \Rightarrow s \vdash}\,\mathsf{gen}$$

(where $x \in \mathbb{V}$ is fresh)

Rules for terms.

$$\frac{\Gamma \vdash \qquad (x/_V : A) \in \Gamma \qquad U \text{ atom} \qquad p \colon U \twoheadrightarrow V \text{ surjective}}{\Gamma \vdash p^*\widehat{x}/_U}\,\mathsf{cell}$$

$$\frac{\Gamma \vdash t/_U \qquad \Gamma \vdash s/_V \qquad \partial_k^+ t \equiv \partial_k^- s}{\Gamma \vdash (t \#_k s)/_{(U \#_k V)}}\,\mathsf{paste}_k, \quad k < \min(\dim(U), \dim(V))$$

Rules for substitutions.

$$\frac{\Gamma \vdash}{\Gamma \vdash \langle\rangle : \Gamma}\,\mathsf{id} \qquad \frac{\Delta \vdash \sigma : \Gamma \qquad \Gamma, x : s/_U \Rightarrow r/_V \vdash \qquad \Delta \vdash t/_{U \Rightarrow V} : s[\sigma] \Rightarrow r[\sigma]}{\Delta \vdash \langle \sigma, x \mapsto t\rangle : (\Gamma, x : s \Rightarrow r)}\,\mathsf{ext}$$

In the rules cell and paste, the terms $\widehat{x}$ and $t \#_k s$ are defined as follows:

- $\widehat{x}$ is the unique term of shape V which sends the greatest element of V to x, and, if $A \equiv t \Rightarrow s$, is equal to t on $\partial^- V$ and to s on $\partial^+ V$;
- $t \#_k s$ is the unique term of shape $U \#_k V$ that is equal to t on $U \hookrightarrow (U \#_k V)$ and to s on $V \hookrightarrow (U \#_k V)$.

The side conditions for gen and paste ensure that this is well-defined.

To define the action $t[\sigma]$ of a well-formed substitution σ on a term t, we extend σ to a function $\mathbb{V} \to \mathbb{V}$ as follows: for all $x \in \mathbb{V}$, if $(x \mapsto t/_U) \in \sigma$, we let $\sigma(x) := t(\top)$, where $\top$ is the greatest element of U; otherwise, $\sigma(x) := x$. Then $t[\sigma]$ is the composite of $t \colon U \to \mathbb{V}$ and $\sigma \colon \mathbb{V} \to \mathbb{V}$. Note that this is well-defined because a well-formed substitution assigns to each variable a term whose shape is an atom.

3.9 (Syntactic category)**.** The syntactic category $\mathbf{Ctx}[\mathsf{DiagSet}]$ has

- well-formed contexts Γ as objects, and
- well-formed substitutions as morphisms from Δ to Γ,

with the obvious composition of substitutions, and empty substitutions as identities.

Theorem 3.10 — *The category* $\mathbf{Ctx}[\mathsf{DiagSet}]^{\mathrm{op}}$ *is equivalent to* $\odot\mathbf{Cpx}_{fsb}$.

Sketch of proof. We define an encoding enc of finite support-based diagrammatic complexes, diagrams, and morphisms as contexts, terms, and substitutions. Given $(X,\mathscr{X})$, we pick an injective function $\mathsf{name}\colon \mathscr{X} \to \mathbb{V}$, assigning unique variable names to the generating cells of X.

For all diagrams $d\colon U \to X$, we define a term $\mathsf{enc}(d)$ as follows: for all $x \in U$, we let $\mathsf{enc}(d)(x)$ be equal to $\mathsf{name}(\mathsf{supp}(d|_{\mathrm{cl}\{x\}}))$. Since $(X,\mathscr{X})$ is support-based, $\mathsf{enc}(d) \equiv \mathsf{enc}(d')$ implies $d = d'$.

Let n be the greatest dimension in which $\mathscr{X}_n$ is non-empty, and pick a linear ordering $x_1,\ldots,x_{m_k}$ of $\mathscr{X}_k$ for all $k \le n$. We let $\mathsf{enc}(X,\mathscr{X}) := \Gamma_0,\ldots,\Gamma_n$, where

$$\Gamma_k := \mathsf{name}(x_1) : \mathsf{enc}(\partial^- x_1) \Rightarrow \mathsf{enc}(\partial^+ x_1), \ldots, \mathsf{name}(x_{m_k}) : \mathsf{enc}(\partial^- x_{m_k}) \Rightarrow \mathsf{enc}(\partial^+ x_{m_k}).$$

By the construction of X as a colimit of its generating cells, any map $X \to Y$ is uniquely determined by what it does on $\mathscr{X}$. Given a map $f\colon (X,\mathscr{X}) \to (Y,\mathscr{Y})$ in $\odot\mathbf{Cpx}_{fsb}$, we let $\mathsf{enc}(f)$ be the substitution

$$\langle \mathsf{name}_X(x) \mapsto \mathsf{enc}_Y(f(x))\rangle_{x\in\mathscr{X}}.$$

Conversely, we define an interpretation $[\![-]\!]$ of well-formed contexts, terms, and substitutions by induction on inference rules of $\mathsf{DiagSet}$. At each step the interpretation $[\![\Gamma]\!]$ of a well-formed context is a support-based diagrammatic complex with one generator $[\![\widehat{x}]\!]$ of shape U for each variable $x/_U$ in Γ.

- (init) The interpretation of the empty context is the initial diagrammatic set.
- (pt) Suppose $[\![\Gamma]\!]$ is defined. The interpretation of $\Gamma, x : \varnothing$ is the coproduct $[\![\Gamma]\!] + \bullet$. The interpretation of $\widehat{x}$ is the inclusion $\bullet \hookrightarrow [\![\Gamma]\!] + \bullet$.
- (gen) Suppose $[\![\Gamma]\!]$ and $[\![t/_U]\!], [\![s/_V]\!]$ are defined. The interpretation of $\Gamma, x : t \Rightarrow s$ is the pushout of $\partial[\![\widehat{x}]\!]\colon \partial(U \Rightarrow V) \to [\![\Gamma]\!]$ and $\partial(U \Rightarrow V) \hookrightarrow (U \Rightarrow V)$, quotiented by the equations (1), where $\partial[\![\widehat{x}]\!]$ is equal to $[\![t]\!]$ on $\partial^-(U \Rightarrow V)$ and to $[\![s]\!]$ on $\partial^+(U \Rightarrow V)$.
- (cell) Suppose $[\![\Gamma]\!]$ is defined and has a generating cell $[\![\widehat{x}]\!]$. The interpretation of $p^*\widehat{x}$ is $p;[\![\widehat{x}]\!]$.
- (paste_k) Suppose $[\![\Gamma]\!]$ and $[\![t]\!],[\![s]\!]$ are defined with $\partial_k^+[\![t]\!] = [\![\partial_k^+ t]\!] = [\![\partial_k^- s]\!] = \partial_k^-[\![s]\!]$. The interpretation of $t \#_k s$ is the diagram $[\![t]\!] \#_k [\![s]\!]$.
- (id) The interpretation of the empty substitution in context Γ is the identity of $[\![\Gamma]\!]$.
- (ext) Suppose $[\![\sigma]\!]$ and $[\![t]\!]$ are defined, where $[\![\widehat{x}]\!]$ and $[\![t]\!]$ both have the same shape U. By the construction of $[\![\Gamma,x]\!]$ as a colimit of $[\![\Gamma]\!]$ and U, the pair of $[\![\sigma]\!]\colon [\![\Gamma]\!] \to [\![\Delta]\!]$ and $[\![t]\!]\colon U \to [\![\Gamma]\!]$ induces a unique morphism $[\![\sigma, x \mapsto t]\!]\colon [\![\Gamma,x]\!] \to [\![\Delta]\!]$.

It is routine to check that enc and $[\![-]\!]$ define contravariant functors between $\odot\mathbf{Cpx}_{fsb}$ and $\mathbf{Ctx}[\mathsf{DiagSet}]$, and that they are each other's inverse up to natural isomorphism. ∎

Remark 3.11. The proof of Theorem 3.10 gives a semantic characterisation of well-formed terms as diagrams in a diagrammatic set. An immediate consequence is that the following rule is admissible:

$$\frac{\Gamma \vdash t/_V \qquad p\colon U \to V \ \mathsf{map}}{\Gamma \vdash p^* t/_U}\ \mathsf{pb}$$

where p is an arbitrary map of regular molecules.

Comment 3.12. A sticking point in our type theory is the fact that cell is parametrised by an arbitrary surjective map of atoms p. This is necessary to access the "weak units" and degenerate cells which in our framework are needed, among other things, to model nullary operations in an algebraic theory.

In practice, however, this is the one point in which the underlying implementation of regular molecules and their maps has to be explicitly accessed in order to define p and its domain. To avoid this, in a practical implementation, we want to include explicitly some extra admissible rules, corresponding to the application of useful maps that are parametric in their codomain.

In particular, we want to explicitly include

- the trivial case $p \equiv \mathrm{id}_U$:

$$\frac{\Gamma \vdash \qquad (x/_U : A) \in \Gamma}{\Gamma \vdash \widehat{x}/_U}\ \mathsf{cell}',$$

- *unit* rules, modelling [13, §4.16]:

$$\frac{\Gamma \vdash t/_U}{\Gamma \vdash \mathsf{unit}(t) := \tau^*(t) : t \Rightarrow t}\ \mathsf{unit},$$

- *left and right unitor rules*, modelling [*ibid.*, §4.17]:

$$\frac{\Gamma \vdash t/_U \qquad V \sqsubseteq \partial^- U \ \mathsf{round}}{\Gamma \vdash \mathsf{lunitor}_V(t) := (\ell^-_{V \hookrightarrow U})^* t}\ \mathsf{lunitor} \qquad \frac{\Gamma \vdash t/_U \qquad V \sqsubseteq \partial^+ U \ \mathsf{round}}{\Gamma \vdash \mathsf{runitor}_V(t) := (r^-_{V \hookrightarrow U})^* t}\ \mathsf{runitor}$$

  where V can be specified, for example, by the set of positions of its maximal elements.

We may also have extra rules for *simplex and cube degeneracy* maps and for *cube connection* maps, in the case where U is an oriented simplex or cube as in [*ibid.*, §3.33]. All of these are implemented as diagram methods in `rewalt`.

Example 3.13. As an example, we give a presentation in DiagSet of the theory of a left-unital binary operation, together with its implementation in `rewalt`. In the framework of diagrammatic sets, a many-sorted "monoidal theory" is presented by a diagrammatic complex with a single 0-cell; this is analogous to the way a monoidal category is a bicategory with a single 0-cell. The sorts are generating 1-cells, the basic operations are generating 2-cells, and "oriented equations" are generating 3-cells.

First, we add a single 0-cell x and a single sort a.

$$\dfrac{\dfrac{\dfrac{}{\langle\rangle \vdash}\ \mathsf{init}}{x : \varnothing}\ \mathsf{pt}}{x : \varnothing \vdash \widehat{x}}\ \mathsf{cell}' \qquad \dfrac{\dfrac{x : \varnothing \vdash \widehat{x} \qquad x : \varnothing \vdash \widehat{x}}{x : \varnothing, a : \widehat{x} \Rightarrow \widehat{x} \vdash}\ \mathsf{gen}}{x : \varnothing, a : \widehat{x} \Rightarrow \widehat{x} \vdash \widehat{a}}\ \mathsf{cell}'$$

```python
import rewalt
Lun = rewalt.DiagSet()
x = Lun.add('x')
a = Lun.add('a', x, x)
```

Let $\Gamma := x : \varnothing, a : \widehat{x} \Rightarrow \widehat{x}$. We add a binary operation m.

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \widehat{a} \qquad \Gamma \vdash \widehat{a}}{\Gamma \vdash \widehat{a} \#_0 \widehat{a}}\ \mathsf{paste}_0 \qquad \Gamma \vdash \widehat{a}}{\Gamma, m : \widehat{a} \#_0 \widehat{a} \Rightarrow \widehat{a} \vdash}\ \mathsf{gen}}{\Gamma, m : \widehat{a} \#_0 \widehat{a} \Rightarrow \widehat{a} \vdash \widehat{m}}\ \mathsf{cell}'$$

```python
m = Lun.add('m', a.paste(a), a)
```

Let $\Gamma' := \Gamma, m : \widehat{a} \#_0 \widehat{a} \Rightarrow \widehat{a}$. We produce a weak unit on x and add a nullary operation u.

$$\dfrac{\dfrac{\dfrac{\Gamma' \vdash \widehat{x}}{\Gamma' \vdash \mathsf{unit}(\widehat{x})}\ \mathsf{unit} \qquad \Gamma' \vdash \widehat{a}}{\Gamma', u : \mathsf{unit}(\widehat{x}) \Rightarrow \widehat{a} \vdash}\ \mathsf{gen}}{\Gamma', u : \mathsf{unit}(\widehat{x}) \Rightarrow \widehat{a} \vdash \widehat{u}}\ \mathsf{cell}'$$

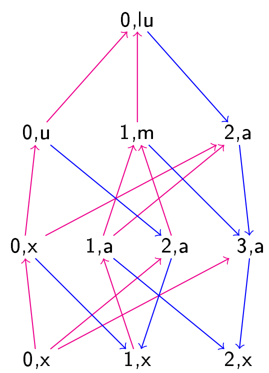
```python
u = Lun.add('u', x.unit(), a)
```

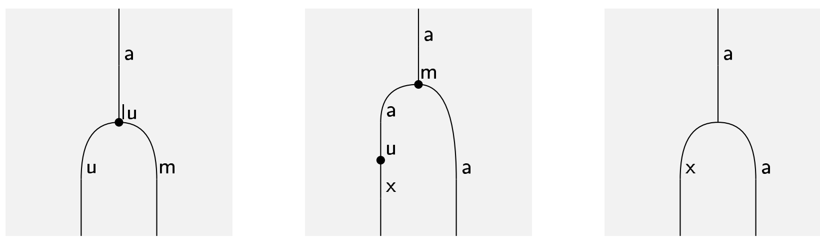
Let $\Gamma'' := \Gamma', u : \mathsf{unit}(\widehat{x}) \Rightarrow \widehat{a}$. We produce a left unitor 2-cell on a, and add an "oriented equation" exhibiting the fact that u is a left unit for m.

$$\dfrac{\dfrac{\dfrac{\Gamma''\vdash\widehat{u}\qquad\Gamma''\vdash\widehat{a}}{\Gamma''\vdash\widehat{u}\#_0\widehat{a}}\,\mathsf{paste}_0\qquad\Gamma''\vdash\widehat{m}}{\Gamma''\vdash(\widehat{u}\#_0\widehat{a})\#_1\widehat{m}}\,\mathsf{paste}_1\qquad\dfrac{\Gamma''\vdash\widehat{a}/\mathsf{arrow}}{\Gamma''\vdash\mathsf{lunitor}_{\partial^-\mathsf{arrow}}(\widehat{a})}\,\mathsf{lunitor}}{\Gamma'', lu : ((\widehat{u}\#_0\widehat{a})\#_1\widehat{m})\Rightarrow\mathsf{lunitor}_{\partial^-\mathsf{arrow}}(\widehat{a})\vdash}\,\mathsf{gen}$$

```
lu = Lun.add('lu', u.paste(a).paste(m), a.lunitor())
```

The following is a representation of *lu* as a term of DiagSet, that is, an oriented graded poset labelled with names, together with string diagram representations of *lu*, its input boundary, and its output boundary, and the `rewalt` code that generated them.






```
lu.hasse(tikz=True)
lu.draw(bgcolor='gray!10', tikz=True)
lu.input.draw(bgcolor='gray!10', tikz=True)
lu.output.draw(bgcolor='gray!10', tikz=True)
```

Comment 3.14. Provided we have a unique underlying representation of shapes, as described in Section 2, every term of DiagSet also has a unique representation. In this sense, terms of DiagSet are "noncomputational": all the computation, which consists exclusively of computing and matching shapes, happens under the hood before a term is even created, so the equality theory of terms is trivial.

This is intended. Rather than a computational theory in itself, DiagSet is intended as a *substrate for computational theories* according to the paradigm of higher-dimensional rewriting. A term $t : r^- \Rightarrow r^+$ can be seen as a rewrite of the "lower-dimensional" term r^- to the term r^+, and the extension of t via the paste_k rules establishes how the rewrite can happen in a wider context. In this sense, every well-formed context in DiagSet contains its own internal computational theory on terms of each dimension.

Remark 3.15. While "rewrites in context" can be built with the paste_k rules, this is quite impractical. In practice, one wants to start from a diagram and apply a generating rewrite directly to a subdiagram. This is modelled by *pasting along a subdiagram* [13, §4.12] in the theory of diagrammatic sets.

Pasting along a subdiagram is implemented in `rewalt` with methods `to_inputs` and `to_outputs`. These invoke a procedure for recognising subdiagrams, which currently uses a quite naive algorithm. The issue of recognising subdiagrams deserves further study, so we leave it to future work.

Conclusions and outlook

We have provided a formal implementation of "plain" diagrammatic sets. An obvious next step is the formalisation of *weakly invertible* cells, and then of diagrammatic sets with weak composites, a model of weak higher categories [13, Sections 5, 6]. This is in fact part of `rewalt`, but still lacks a formal analysis.

In addition, we still have a limited range of high-level methods for handling weak units. We may want, for example, flexible higher-dimensional versions of "Mac Lane triangle" rules for shuffling weak units around. Development of these methods, and others tailored to specific applications, will likely go hand in hand with practical experience in the use of `rewalt` as a proof assistant.

To conclude, we have only scratched the surface of the algorithm and complexity theory of diagram rewriting in higher dimensions. In particular, we have not yet studied the problem of searching for a subdiagram within another diagram, whose solution is essential to any form of fully automated or assisted diagram rewriting. We plan to tackle this problem in future work.

References

[1] L. Babai (2016): *Graph isomorphism in quasipolynomial time [extended abstract]*. In: *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, ACM, pp. 684–697, doi:10.1145/2897518.2897542.

[2] K. Bar & J. Vicary (2017): *Data structures for quasistrict higher categories*. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, doi:10.1109/lics.2017.8005147.

[3] T. Benjamin, E. Finster & S. Mimram (2021): *Globular weak ω-categories as models of a type theory*. arXiv:2106.04475.

[4] G. Bonfante & Y. Guiraud (2009): *Polygraphic programs and polynomial-time functions*. *Logical Methods in Computer Science* 5(2), doi:10.2168/lmcs-5(2:14)2009.

[5] A. Burroni (1993): *Higher-dimensional word problems with applications to equational logic*. *Theoretical Computer Science* 115(1), pp. 43–62, doi:10.1016/0304-3975(93)90054-w.

[6] P.L. Curien, C. Ho Thanh & S. Mimram (2019): *Syntactic approaches to opetopes*.

[7] C. Dorn (2018): *Associative n-categories*. Ph.D. thesis, University of Oxford.

[8] E. Finster & S. Mimram (2017): *A type-theoretical definition of weak ω-categories*. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, doi:10.1109/lics.2017.8005124.

[9] E. Finster, D. Reutter & J. Vicary (2020): *A type theory for strictly unital ∞-categories*. arXiv:2007.08307.

[10] E. Finster, A. Rice & J. Vicary (2021): *A type theory for strictly associative infinity categories*. arXiv:2109.01513.

[11] S. Forest (2019): *Unifying notions of pasting diagrams*.

[12] Y. Guiraud (2019): Rewriting methods in higher algebra. Thèse d'habilitation à diriger des recherches, Université Paris 7.

[13] A. Hadzihasanovic (2020): *Diagrammatic sets and rewriting in weak higher categories*. arXiv:2007.14505.

[14] A. Hadzihasanovic (2021): *The smash product of monoidal theories*. In: *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, doi:10.1109/lics52264.2021.9470575.

[15] C. Ho Thanh, P.L. Curien & S. Mimram (2019): *A Sequent Calculus for Opetopes*. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, doi:10.1109/lics.2019.8785667.

[16] G.L. Miller (1979): *Graph isomorphism, general remarks*. *Journal of Computer and System Sciences* 18(2), pp. 128–142, doi:10.1016/0022-0000(79)90043-6.

[17] D. Reutter & J. Vicary (2019): *High-level methods for homotopy construction in associative n-categories*. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, doi:10.1109/lics.2019.8785895.

[18] R. Steiner (1993): *The algebra of directed complexes*. *Applied Categorical Structures* 1(3), pp. 247–284, doi:10.1007/bf00873990.

Dependent Optics

Pietro Vertechi
pietro.vertechi@protonmail.com

A wide variety of bidirectional data accessors, ranging from *mixed optics* to *functor lenses*, can be formalized within a unique framework—*dependent optics*. Starting from two indexed categories, which encode what maps are allowed in the forward and backward directions, we define the category of dependent optics and establish under what assumptions it has coproducts. Different choices of indexed categories correspond to different families of optics: we discuss dependent lenses and prisms, as well as closed dependent optics. We introduce the notion of *Tambara representation* and use it to classify contravariant functors from the category of optics, thus generalizing the *profunctor encoding* of optics to the dependent case.

1 Introduction

Lenses [2, 10] are composable, bidirectional data accessors. They can be thought of as a collection of two methods: a get method, to access a particular field of a data structure, and a put method, to build a new instance of the data structure with an updated field value. Lenses and their more recent generalization, optics, have been implemented and explored in the popular *Haskell* library lens [12]. The possible fields of application vary widely, from game theory [9] to automatic differentiation [8].

The current formalization of optics [7, 20] extends the original theory of lenses from Cartesian categories to arbitrary symmetric monoidal categories, or even *actegories*, thus including under a unique formalism a wide variety of data accessors. Unfortunately, this approach fails to include a distinct elegant generalization of lenses, namely *functor lenses* [21]: every pseudofunctor $\mathscr{R}\colon \mathscr{C}^{\mathrm{op}} \to \mathbf{Cat}$ induces a fibration of categories $\mathbf{Lens}_{\mathscr{R}} \to \mathscr{C}$ via the Grothendieck construction on the pointwise opposite of $\mathscr{R}$. Classical categories of lenses, as well as novel examples, can be obtained with this approach.

The aim of this work is to develop a common generalization of the theories of optics and functor lenses, via the theory of bicategories and pseudofunctors. In section 2 we lay the fundamental definition of *dependent optic* and show that it encompasses both regular optics and functor lenses. We show under what conditions the category of dependent optics has coproducts. In section 3 we give some examples of dependent optics—dependent (monoidal) lenses, dependent (monoidal) prisms, and closed dependent optics—where the key ingredient is the operation of tensoring (co)modules over a (co)monoid in a symmetric monoidal category. Finally, in section 4, we establish the notion of *Tambara representation*. We show that $\mathscr{D}$-valued Tambara representations are equivalent to contravariant functors from the category of optics to an arbitrary category $\mathscr{D}$, thus generalizing the profunctor encoding of optics to the dependent case.

2 Dependent optics

Classically, a lens from a domain (X, X') to a codomain (Y, Y') has a get method get: $X \to Y$ and a put method put: $X \times Y' \to X'$. This definition suggests two broad classes of generalizations. One approach, *functor lenses* [21], replaces the domain and codomain with pairs (X, P) and (Y, Q), where the objects

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 128–144, doi:10.4204/EPTCS.380.8

P and Q live in categories parameterized by X and Y respectively. The put method is then encoded as a map put: $\mathrm{get}^*(Q) \to P$. Another approach, *mixed optics* [7, 20], replaces the Cartesian product $\times$ with two actions Ⓛ,Ⓡ of a shared monoidal category $\mathscr{M}$ on categories $\mathscr{C}_L, \mathscr{C}_R$. This requires to tweak the original definition of lens to an equivalent one expressed via a coend

$$\int^{M \in \mathscr{M}} \mathscr{C}_L(X, M Ⓛ Y) \times \mathscr{C}_R(M Ⓡ Y', X').$$

Remark 1. *Some authors (see for instance [7]) work in the setting of enriched categories, so that the above coend is not taken in* **Set** *but rather in some monoidal category $\mathscr{V}$. For simplicity, in this article we will work in the standard non-enriched setting.*

It follows from the Yoneda reduction lemma [20, Lm. 1.2.2] that this definition recovers classical lenses when a Cartesian category acts on itself. From a practical perspective, optics are equivalence classes of pairs of morphisms

$$l \colon X \to M Ⓛ Y \quad \text{and} \quad r \colon M Ⓡ Y' \to X',$$

where $M \in \mathrm{Ob}(\mathscr{M})$ is called the *representative*.

The aim of this section is to establish a general definition of *dependent optics* which encompasses both previous generalizations of lenses—functor lenses and optics. The definition is entirely analogous to the definition of optics, but the monoidal actions are replaced by $\mathscr{B}$-indexed categories, where $\mathscr{B}$ is a *bicategory* [1] (see also [13] for a more modern treatment). We will consider the bicategory as a *category weakly enriched in categories*, hence the notation $\mathscr{B}(A,B)$, for $A,B \in \mathrm{Ob}(\mathscr{B})$, will represent the category of morphisms from A to B.

To encode the data of a $\mathscr{B}$-indexed category $\mathscr{L}$, i.e. a pseudofunctor $\mathscr{L} \colon \mathscr{B}^{\mathrm{op}} \to \mathbf{Cat}$, we will use the following notation. $\mathscr{L}^A$ for $A \in \mathrm{Ob}(\mathscr{B})$ denotes the category $\mathscr{L}(A)$ and f^* for $f \in \mathrm{Ob}(\mathscr{B}(A,B))$ denotes the functor $\mathscr{L}(f)$. Throughout this manuscript, we will work with two pseudofunctors, $\mathscr{L}$ and $\mathscr{R}$. To avoid ambiguities, we will use the notation $f^{*'}$ to denote $\mathscr{R}(f)$.

Definition 1. *Let $\mathscr{B}$ be a bicategory. Let $\mathscr{L}, \mathscr{R}$ be $\mathscr{B}$-indexed categories. The category* $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$ *of* dependent optics *has, as objects, triplets $(X,X')^A$, with $A \in \mathrm{Ob}(\mathscr{B})$, $X \in \mathrm{Ob}(\mathscr{L}^A)$, $X' \in \mathrm{Ob}(\mathscr{R}^A)$. Morphisms between $(X,X')^A$ and $(Y,Y')^B$ are given by the following coend:*

$$\mathbf{Optic}_{\mathscr{L},\mathscr{R}}\left((X,X')^A,(Y,Y')^B\right) = \int^{f \in \mathscr{B}(A,B)} \mathscr{L}^A(X, f^*Y) \times \mathscr{R}^A(f^{*'}Y', X'). \tag{1}$$

To refer to specific morphisms explicitly, we denote by $\langle l \,|\, r \rangle$ the morphism given by $l \in \mathscr{L}^A(X, f^*Y)$ and $r \in \mathscr{R}^A(f^{*'}Y', X')$, and we say that it has representative f.

More explicitly, morphisms in $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}\left((X,X')^A,(Y,Y')^B\right)$ are equivalence classes of pairs (l,r) with $l \colon X \to f^*Y$ and $r \colon f^{*'}Y' \to X'$, where $f \colon A \to B$ is called the representative. The equivalence relation is generated by

$$(\mathscr{L}(m)_Y \circ l,\, r) \sim (l,\, r \circ \mathscr{R}(m)_{Y'}),$$

with $m \colon f \Rightarrow g$, $l \colon X \to f^*Y$ and $r \colon g^{*'}Y' \to X'$, where $f,g \colon A \rightrightarrows B$ are parallel 1-morphisms in $\mathscr{B}$.

Remark 2. *Here and in what follows we assume that the above coend exists, either because the category $\mathscr{B}(A,B)$ is small (and small colimits exist in* **Set***), or because we can compute it explicitly.*

The generalization of optics via $\mathscr{B}$-indexed categories, rather than monoidal actions, has been proposed in [17], where composition of optics is explained in terms of Kan extensions. Here, we will adopt a direct, explicit approach. While the chosen formalisms are different, the two definitions have been shown to be equivalent in [4, Ex. 4.2].

Let θ, θ' encode the coherence natural transformations for $\mathscr{L}, \mathscr{R}$ respectively. In particular, we have natural isomorphisms

$$\theta_A \colon \mathrm{Id}_{\mathscr{L}^A} \Rightarrow \mathrm{Id}_A^* \quad \text{and} \quad \theta_{f,g} \colon f^* \circ g^* \Rightarrow (g \circ f)^*.$$

θ'_A and $\theta'_{f,g}$ are defined in an analogous way. The identity optic is defined as follows:

$$\mathrm{Id}_{(X,X')^A} := \big\langle (\theta_A)_X \,|\, (\theta_A'^{-1})_{X'} \big\rangle. \tag{2}$$

The map

$$\begin{array}{c} \mathscr{L}^B(Y, g^*Z) \times \mathscr{R}^B(g^{*'}Z', Y') \times \mathscr{L}^A(X, f^*Y) \times \mathscr{R}^A(f^{*'}Y', X') \\ \downarrow \\ \mathbf{Optic}_{\mathscr{L},\mathscr{R}}\big((X,X')^A, (Z,Z')^C\big) \end{array}$$

given by

$$\langle l_2 \,|\, r_2 \rangle \circ \langle l_1 \,|\, r_1 \rangle = \Big\langle (\theta_{f,g})_Z \circ f^*(l_2) \circ l_1 \,|\, r_1 \circ f^{*'}(r_2) \circ (\theta'^{-1}_{f,g})_{Z'} \Big\rangle \tag{3}$$

is extranatural in f, g and thus induces a composition function

$$\begin{array}{c} \mathbf{Optic}_{\mathscr{L},\mathscr{R}}\big((Y,Y')^B, (Z,Z')^C\big) \times \mathbf{Optic}_{\mathscr{L},\mathscr{R}}\big((X,X')^A, (Y,Y')^B\big) \\ \downarrow \\ \mathbf{Optic}_{\mathscr{L},\mathscr{R}}\big((X,X')^A, (Z,Z')^C\big). \end{array}$$

Theorem 1. $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$ *is a category.*

Proof. Checking the category axioms (unity and associativity) is tedious but straightforward. Here, we denote λ_f, ρ_f the left and right unitors in $\mathscr{B}$. We use the fact that θ_A (resp. $\theta_A'^{-1}$) is a natural transformation $\mathrm{Id}_{\mathscr{L}^A} \Rightarrow \mathrm{Id}_A^*$ (resp. $\mathrm{Id}_A^{*'} \Rightarrow \mathrm{Id}_{\mathscr{R}^A}$), as well as the identity coherence law for a pseudofunctor, keeping in mind that **Cat** is a strict 2-category and hence has a trivial unitor. Given an optic $\langle l \,|\, r \rangle \colon (X,X')^A \to (Y,Y')^B$ with representative f:

$$\begin{aligned} \langle l \,|\, r \rangle \circ \mathrm{Id}_{(X,X')^A} &= \Big\langle (\theta_{\mathrm{Id}_A,f})_Y \circ \mathrm{Id}_A^*(l) \circ (\theta_A)_X \,|\, (\theta_A'^{-1})_{X'} \circ \mathrm{Id}_A^{*'}(r) \circ (\theta'^{-1}_{\mathrm{Id}_A,f})_{Y'} \Big\rangle \\ &= \Big\langle (\theta_{\mathrm{Id}_A,f})_Y \circ (\theta_A)_{f^*Y} \circ l \,|\, r \circ (\theta_A'^{-1})_{f^{*'}Y'} \circ (\theta'^{-1}_{\mathrm{Id}_A,f})_{Y'} \Big\rangle \\ &= \big\langle (\theta_{\mathrm{Id}_A,f})_Y \circ (\theta_A)_{f^*Y} \circ l \,|\, r \circ \mathscr{R}(\rho_f)_{Y'} \big\rangle \\ &= \big\langle \mathscr{L}(\rho_f)_Y \circ (\theta_{\mathrm{Id}_A,f})_Y \circ (\theta_A)_{f^*Y} \circ l \,|\, r \big\rangle \\ &= \langle l \,|\, r \rangle, \end{aligned}$$

where $\mathscr{R}(\rho_f)_{Y'}$ can be moved to the left as $\mathscr{L}(\rho_f)_Y$ thanks to the equivalence relation introduced by the coend. Analogously,

$$\begin{aligned}
\mathrm{Id}_{(Y,Y')^B} \circ \langle l \,|\, r \rangle &= \left\langle (\theta_{f,\mathrm{Id}_B})_Y \circ f^*((\theta_B)_Y) \circ l \,\middle|\, r \circ f^{*'}((\theta_B'^{-1})_{Y'}) \circ (\theta'^{-1}_{f,\mathrm{Id}_B})_{Y'} \right\rangle \\
&= \langle (\theta_{f,\mathrm{Id}_B})_Y \circ f^*((\theta_B)_Y) \circ l \,|\, r \circ \mathscr{R}(\lambda_f)_{Y'} \rangle \\
&= \langle \mathscr{L}(\lambda_f)_Y \circ (\theta_{f,\mathrm{Id}_B})_Y \circ f^*((\theta_B)_Y) \circ l \,|\, r \rangle \\
&= \langle l \,|\, r \rangle.
\end{aligned}$$

To prove associativity, let us consider a sequence of morphisms

$$(X,X')^A \xrightarrow{\langle l_1 \,|\, r_1 \rangle} (Y,Y')^B \xrightarrow{\langle l_2 \,|\, r_2 \rangle} (Z,Z')^C \xrightarrow{\langle l_3 \,|\, r_3 \rangle} (W,W')^D,$$

with choices of representatives f, g, h respectively. Then,

$$\begin{aligned}
(\langle l_3 \,|\, r_3 \rangle \circ \langle l_2 \,|\, r_2 \rangle) \circ \langle l_1 \,|\, r_1 \rangle &= \left\langle (\theta_{g,h})_W \circ g^*(l_3) \circ l_2 \,\middle|\, r_2 \circ g^{*'}(r_3) \circ (\theta'^{-1}_{g,h})_{W'} \right\rangle \circ \langle l_1 \,|\, r_1 \rangle \\
&= \langle (\theta_{f,g;h})_W \circ f^*((\theta_{g,h})_W) \circ f^*(g^*(l_3)) \circ f^* l_2 \circ l_1 \,| \\
&\quad r_1 \circ f^{*'}(r_2) \circ f^{*'}(g^{*'}(r_3)) \circ f^{*'}((\theta'^{-1}_{g,h})_{W'}) \circ (\theta'^{-1}_{f,g;h})_{W'} \rangle.
\end{aligned}$$

Whereas, when associating in a different order, one has

$$\begin{aligned}
\langle l_3 \,|\, r_3 \rangle \circ (\langle l_2 \,|\, r_2 \rangle \circ \langle l_1 \,|\, r_1 \rangle) &= \langle l_3 \,|\, r_3 \rangle \circ \left\langle (\theta_{f,g})_Z \circ f^*(l_2) \circ l_1 \,\middle|\, r_1 \circ f^{*'}(r_2) \circ (\theta'^{-1}_{f,g})_{Z'} \right\rangle \\
&= \langle (\theta_{f;g,h})_W \circ (f;g)^*(l_3) \circ (\theta_{f,g})_Z \circ f^*(l_2) \circ l_1 \,| \\
&\quad r_1 \circ f^{*'}(r_2) \circ (\theta'^{-1}_{f,g})_{Z'} \circ (f;g)^{*'}(r_3) \circ (\theta'^{-1}_{f;g,h})_{W'} \rangle \\
&= \langle (\theta_{f;g,h})_W \circ (\theta_{f,g})_{h^*W} \circ f^*(g^*(l_3)) \circ f^*(l_2) \circ l_1 \,| \\
&\quad r_1 \circ f^{*'}(r_2) \circ f^{*'}(g^{*'}(r_3)) \circ (\theta'^{-1}_{f,g})_{h^{*'}W'} \circ (\theta'^{-1}_{f;g,h})_{W'} \rangle.
\end{aligned}$$

The two optics are equal, thanks to the relationships

$$\begin{aligned}
\mathscr{L}(\alpha_{f,g,h})_W \circ (\theta_{f;g,h})_W \circ (\theta_{f,g})_{h^*W} &= (\theta_{f,g;h})_W \circ f^*((\theta_{g,h})_W), \\
(\theta'^{-1}_{f,g})_{h^{*'}W'} \circ (\theta'^{-1}_{f;g,h})_{W'} \circ \mathscr{R}(\alpha^{-1}_{f,g,h})_{W'} &= f^{*'}((\theta'^{-1}_{g,h})_{W'}) \circ (\theta'^{-1}_{f,g;h})_{W'},
\end{aligned}$$

where $\alpha_{f,g,h}$ is the associator of $\mathscr{B}$. □

Remark 3. *Unlike the dependent lenses case, here we generally do not have a pseudofunctor* $\mathbf{Optic}_{\mathscr{L},\mathscr{R}} \to \mathscr{B}$*. Such a pseudofunctor would be ill-defined on morphisms in* $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$ *due to the equivalence relation imposed by the coend. To obviate this issue, a possible approach worthy of future exploration would be to define a* bicategory *of dependent optics where, instead of identifying equivalent 1-morphisms, we add 2-morphisms between them. See [3] for a purely bicategorical approach to dependent optics.*

2.1 Comparison with existent constructions

Dependent optics simultaneously generalize both mixed optics [20] and functor lenses [21]. Intuitively, mixed optics are dependent optics where the bicategory $\mathscr{B}$ has a unique object, whereas functor lenses are dependent optics where $\mathscr{B}$ is a 1-category and the $\mathscr{B}$-indexed category $\mathscr{L}$ is trivial.

Proposition 1. *Mixed optics [20, Def. 6.1.1] are a particular case of dependent optics, where the source bicategory has a unique object.*

Proof. Let us consider two categories $\mathscr{C}_L$ and $\mathscr{C}_R$ acted on by a monoidal category $\mathscr{M}$. We can consider the bicategory $\mathbb{B}\mathscr{M}$ obtained by delooping. Explicitly, $\mathbb{B}\mathscr{M}$ has a unique object $*$ with endomorphism category $\mathbb{B}\mathscr{M}(*,*) = \mathscr{M}$, where composition is given by the monoidal structure of $\mathscr{M}$. Then the action of $\mathscr{M}$ on another category $\mathscr{C}$ induces a pseudofunctor $\mathscr{M} \to \mathbf{Cat}$. Under this correspondence, optics for the actions $\psi_L \colon \mathscr{M} \to [\mathscr{C}_L, \mathscr{C}_L]$ and $\psi_R \colon \mathscr{M} \to [\mathscr{C}_R, \mathscr{C}_R]$ are the same as optics for the corresponding pseudofunctors $\mathscr{L}, \mathscr{R} \colon \mathbb{B}\mathscr{M} \rightrightarrows \mathbf{Cat}$, hence they are a special case of dependent optics with $\mathscr{B} = (\mathbb{B}\mathscr{M})^{\mathrm{op}}$. ☐

Proposition 2. *Functor lenses, as defined in [21], are a particular case of dependent optics, where the source bicategory $\mathscr{B}$ is a category and the $\mathscr{B}$-indexed category $\mathscr{L}$ is trivial.*

Proof. Let $\mathscr{B}$ be a 1-category. In [21], functor lenses are defined as the Grothendieck construction of the pointwise opposite of a $\mathscr{B}$-indexed category $\mathscr{R}$. Let $\bullet$ be the terminal $\mathscr{B}$-indexed category. Then,

$$\mathbf{Lens}_{\mathscr{R}} \simeq \mathbf{Optic}_{\bullet,\mathscr{R}}.$$

Indeed, objects in $\mathbf{Optic}_{\bullet,\mathscr{R}}$ are simply pairs (A, X'), with $X' \in \mathscr{R}^A$, as there always is a unique object in $\bullet^A$. As $\mathscr{B}$ has no non-trivial 2-morphisms, we have

$$\int^{f \in \mathscr{B}(A,B)} \mathscr{R}^A(f^{*'}Y', X') \simeq \coprod_{f \in \mathscr{B}(A,B)} \mathscr{R}^A(f^{*'}Y', X').$$

☐

2.2 Coproducts

One fundamental motivation for dependent lenses and, more generally, dependent optics is the lack of coproducts in categories of ordinary lenses or optics. This situation is much improved in the dependent case: for instance, adding coproducts to the category of lenses leads naturally to dependent lenses [3]. In the following proposition, we show that there are general conditions to ensure that the category $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$ has coproducts.

Proposition 3. *Let $\mathscr{B}$ be a bicategory with finite coproducts. Let us assume that $\mathscr{L}, \mathscr{R}$ turn finite coproducts in $\mathscr{B}$ into finite products in* $\mathbf{Cat}$. *Then,* $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$ *has finite coproducts.*

Proof. Let $\left((X_i, X_i')^{A_i}\right)_{i \in I}$ be a finite family of objects in $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$. Let $A = \coprod_{i \in I} A_i$. Let $\iota_i \colon A_i \hookrightarrow A$ be the inclusions. Let $X \in \mathrm{Ob}(\mathscr{L}^A)$ and $X' \in \mathrm{Ob}(\mathscr{R}^A)$ be such that, for all $i \in I$,

$$\iota_i^*(X) \simeq X_i \quad \text{and} \quad \iota_i^{*'}(X') \simeq X_i'.$$

For all $(Y, Y')^B \in \mathrm{Ob}\left(\mathbf{Optic}_{\mathscr{L},\mathscr{R}}\right)$ the following holds:

$$\begin{aligned}
\mathbf{Optic}_{\mathscr{L},\mathscr{R}}\left((X,X')^A, (Y,Y')^B\right) &= \int^{f \in \mathscr{B}(A,B)} \mathscr{L}^A(X, f^*Y) \times \mathscr{R}^A(f^{*'}Y', X') \\
&\simeq \int^{(f_i)_{i \in I} \in \prod_{i \in I} \mathscr{B}(A_i,B)} \prod_{i \in I} \mathscr{L}^{A_i}(X_i, f_i^*Y) \times \mathscr{R}^{A_i}(f_i^{*'}Y', X_i') \\
&\simeq \prod_{i \in I} \mathbf{Optic}_{\mathscr{L},\mathscr{R}}\left((X_i, X_i')^{A_i}, (Y,Y')^B\right),
\end{aligned}$$

where the last isomorphism is Fubini's theorem for coends, hence $(X,X')^A$ is the coproduct of $\big((X_i,X_i')^{A_i}\big)_{i\in I}$. □

3 Examples

Different choices of bicategories and functors give rise to different types of optics, see [7] for an overview of the monoidal case, i.e., $\mathscr{B} = (\mathbb{B}\mathscr{M})^{\mathrm{op}}$, as in proposition 1. Here, we discuss *dependent lenses* [21], *dependent prisms*, and generalizations thereof. We then show how the existence of a right adjoint to a given functor can be used to construct further classes of examples of dependent optics. More examples of dependent optics, such as polynomial optics, are described in [17].

3.1 Dependent lenses

Definition 2. *Let $\mathscr{C}$ be a finitely complete category. Let $\mathbf{Span}_{\mathscr{C}}$ be its bicategory of spans. Let $\mathscr{C}/-$ be the $\mathbf{Span}_{\mathscr{C}}$-indexed category of slices. More explicitly,*

$$\mathscr{C}/-\colon \mathbf{Span}_{\mathscr{C}}^{\mathrm{op}} \to \mathbf{Cat}$$

is a pseudofunctor that associates to each object $A \in \mathrm{Ob}(\mathscr{C})$ the slice category $\mathscr{C}/A$. Functoriality is given by pulling back and then pushing forward along the legs of the span. We define the category of dependent lenses *as follows:*

$$\mathbf{DLens}_{\mathscr{C}} := \mathbf{Optic}_{\mathscr{C}/-,\mathscr{C}/-}.$$

Objects in $\mathbf{DLens}_{\mathscr{C}}$ are cospans $X \to A \leftarrow X'$. Morphisms between two cospans $X \to A \leftarrow X'$ and $Y \to B \leftarrow Y'$ are given by

$$\int^{M\in\mathscr{C}/(A\times B)} \mathscr{C}/A(X, M\times_B Y) \times \mathscr{C}/A(M\times_B Y', X'). \tag{4}$$

Equation (4) can be visualized as follows. A class of homomorphisms in $\mathbf{DLens}_{\mathscr{C}}$ with representative $A \leftarrow M \to B$ is given by a pair of dotted arrows that make the following diagram commute.

$$\begin{array}{ccccc} X & \longrightarrow & A & \longleftarrow & X' \\ \vdots\downarrow & & \uparrow & & \uparrow\vdots \\ M\times_B Y & \longrightarrow & M & \longleftarrow & M\times_B Y' \end{array}$$

Morphisms in $\mathbf{DLens}_{\mathscr{C}}$ can be computed explicitly:

$$\begin{aligned} \mathbf{DLens}_{\mathscr{C}}\big((X,X')^A,(Y,Y')^B\big) &= \int^{M\in\mathscr{C}/(A\times B)} \mathscr{C}/A(X, M\times_B Y) \times \mathscr{C}/A(M\times_B Y', X') \\ &\simeq \int^{M\in\mathscr{C}/(A\times B)} \coprod_{X\to Y} \mathscr{C}/(A\times B)(X,M) \times \mathscr{C}/A(M\times_B Y', X') \\ &\simeq \coprod_{X\to Y} \int^{M\in\mathscr{C}/(A\times B)} \mathscr{C}/(A\times B)(X,M) \times \mathscr{C}/A(M\times_B Y', X') \\ &\simeq \coprod_{X\to Y} \mathscr{C}/A(X\times_B Y', X'), \end{aligned} \tag{5}$$

where the last isomorphism follows from the Yoneda reduction lemma [20, Lm 1.2.2].

Even though this category is equivalent to the definition of dependent lenses via the Grothendieck construction in [21], we believe it can have independent practical value. Encoding dependent lenses as maps

$$X \to M \times_B Y \quad \text{and} \quad M \times_B Y' \to X'$$

can lead to a more efficient implementation of, for instance, reverse-mode automatic differentiation (as done in the *Julia* library Diffractor [8]), where the representative M is optimized to contain precisely the information about the input that is required to compute the backward map. Features of the input that are not needed can be discarded, and quantities computed in the forward map can be stored in M if they are useful for the backward map.

The documentation of the Diffractor library [8] hints at the need for dependent optics. Indeed, a key motivation for this work was to build a rigorous dependently-typed framework for bidirectional data transformations that would allow for reverse-mode automatic differentiation with an explicit notion of representative. However, to formalize the difference between our construction of dependent lenses and the one based on functor lenses, we would need to define the *bicategory* of dependent optics, where all the information about the representative is preserved (cp. remark 3).

Unlike lenses, categories of dependent lenses admit finite coproducts, provided that the base category is *lextensive* [6, Sect. 4.4].

Lemma 1. *If $\mathscr{C}$ is a lextensive category, then the inclusion $\mathscr{C} \hookrightarrow \mathbf{Span}_{\mathscr{C}}$ preserves coproducts.*

Proof. Let $A = \coprod_{i \in I} A_i$ be a coproduct in $\mathscr{C}$. Then, for all $B \in \mathrm{Ob}(\mathscr{C})$,

$$\mathbf{Span}_{\mathscr{C}}(A,B) = \mathscr{C}/(A \times B) \simeq \mathscr{C}/\coprod_{i \in I}(A_i \times B) \simeq \prod_{i \in I} \mathscr{C}/(A_i \times B) = \prod_{i \in I} \mathbf{Span}_{\mathscr{C}}(A_i, B),$$

therefore A is the coproduct of $(A_i)_{i \in I}$ in $\mathbf{Span}_{\mathscr{C}}$. □

Proposition 4. *If $\mathscr{C}$ is a lextensive category, then $\mathbf{DLens}_{\mathscr{C}}$ has finite coproducts.*

Proof. By lemma 1, $\mathbf{Span}_{\mathscr{C}}$ has finite coproducts, given by coproducts in $\mathscr{C}$. It is straightforward to show that $\mathscr{C}/-$ turns coproducts into products, as

$$\mathscr{C}/A = \mathscr{C}/\coprod_{i \in I} A_i \simeq \prod_{i \in I} \mathscr{C}/A_i.$$

Thanks to proposition 3, $\mathbf{DLens}_{\mathscr{C}}$ has finite coproducts. □

3.2 Dependent monoidal lenses

The construction in section 3.1 can be generalized to a symmetric monoidal category $(\mathscr{C}, \otimes)$ with reflexive equalizers that are preserved by the tensor product. This is analogous to the approach taken in [21] to generalize lenses to symmetric monoidal categories via commutative comonoids.

Let B be a comonoidal object in $\mathscr{C}$. Given a right B-comodule M and a left B-comodule N, we can define their tensor product over B as the following equalizer:

$$M \otimes_B N := \mathrm{eq}(M \otimes N \rightrightarrows M \otimes B \otimes N).$$

The category of commutative comonoids in $\mathrm{Ob}(\mathscr{C})$, denoted $\mathbf{CComon}_{\mathscr{C},\otimes}$, has finite limits: the pullback of two B-coalgebras Y_1, Y_2 is isomorphic to the tensor product $Y_1 \otimes_B Y_2$ (see [11, C1.1 Lm. 1.1.8] and subsequent discussion for the dual statement). Let $\mathbf{CCoalg}_{(-)}$ and $\mathbf{Comod}_{(-)}$ be the $\mathbf{Span}_{\mathbf{CComon}_{\mathscr{C},\otimes}}$-indexed categories of commutative coalgebras and comodules respectively, where functoriality is given by extension and restriction of scalars. We define the category of *dependent monoidal lenses* as follows:

$$\mathbf{DLens}_{\mathscr{C},\otimes} := \mathbf{Optic}_{\mathbf{CCoalg}_{(-)},\mathbf{Comod}_{(-)}}.$$

A computation analogous to the one in eq. (5) yields the following explicit formula:

$$\mathbf{DLens}_{\mathscr{C},\otimes}\left((X,X')^A,(Y,Y')^B\right) \simeq \coprod_{X\to Y} \mathbf{Comod}_A(X\otimes_B Y',X'),$$

where the morphism $X \to Y$ varies among comonoid homomorphisms.

Proposition 4 can be generalized to the monoidal case, establishing sufficient conditions for the existence of finite coproducts in $\mathbf{DLens}_{\mathscr{C},\otimes}$. In the following proposition, we rely on the fact that, whenever $\mathscr{C}$ has finite coproducts, the forgetful functor $\mathbf{CMon}_{\mathscr{C},\otimes} \to \mathscr{C}$ creates coproducts in $\mathbf{CMon}_{\mathscr{C},\otimes}$. See the proof of [15, Prop. 1.2.14] for the dual statement, which concerns limits of commutative monoids rather than colimits of commutative comonoids.

Proposition 5. *Let $(\mathscr{C},\otimes)$ be a symmetric monoidal category, with reflexive equalizers that are preserved by the tensor product. Let us assume that $\mathscr{C}$ has finite coproducts, and that for all finite coproduct of commutative comonoids $A = \coprod_{i\in I} A_i$, the map*

$$\mathbf{Comod}_A \to \prod_{i\in I} \mathbf{Comod}_{A_i}, \text{ given by } M \mapsto (M\otimes_A A_i)_{i\in I}, \tag{6}$$

is an equivalence of categories. Then, $\mathbf{DLens}_{\mathscr{C}}$ has finite coproducts.

Proof. The equivalence in eq. (6) is monoidal, hence the functor

$$\mathbf{CCoalg}_A \to \prod_{i\in I} \mathbf{CCoalg}_{A_i}, \text{ given by } X \mapsto (X\otimes_A A_i)_{i\in I}, \tag{7}$$

is an equivalence. As $\mathbf{CCoalg}_{(-)} = \mathbf{CComon}_{\mathscr{C},\otimes}/-$, the category $\mathbf{CComon}_{\mathscr{C},\otimes}$ is lextensive. Indeed, in the presence of pullbacks along coproduct injections, eq. (7) is a condition equivalent to extensivity, as shown in [14, Prop. 1.3]. Thanks to lemma 1, the category $\mathbf{Span}_{\mathbf{CComon}_{\mathscr{C},\otimes}}$ has finite coproducts, given by coproducts in $\mathscr{C}$. It follows from eqs. (6) and (7) that the functors $\mathbf{Comod}_{(-)}$ and $\mathbf{CCoalg}_{(-)}$ turn finite coproducts into products. Thanks to proposition 3, $\mathbf{DLens}_{\mathscr{C},\otimes}$ has finite coproducts. ☐

3.3 Dependent (monoidal) prisms

Dependent prisms are dual to dependent lenses. Let $\mathscr{C}$ be a finitely cocomplete category. Let $\mathbf{Cospan}_{\mathscr{C}} = \mathbf{Span}_{\mathscr{C}^{\mathrm{op}}}$ be its bicategory of cospans. Let $-/\mathscr{C}$ be the $\mathbf{Cospan}_{\mathscr{C}}$-indexed category of coslices. We define the category of *dependent prisms* as follows:

$$\mathbf{DPrism}_{\mathscr{C}} := \mathbf{Optic}_{-/\mathscr{C},-/\mathscr{C}}.$$

Objects are given by spans $X \leftarrow A \to X'$. Morphisms between two spans $X \leftarrow A \to X'$ and $Y \leftarrow B \to Y'$ are given by

$$\int^{M\in(A\sqcup B)/\mathscr{C}} A/\mathscr{C}(X, M\sqcup_B Y) \times A/\mathscr{C}(M\sqcup_B Y', X'). \tag{8}$$

As is the case for dependent lenses, the coend in eq. (8) can be computed explicitly:

$$\mathbf{DPrism}_{\mathscr{C}}\left((X,X')^A,(Y,Y')^B\right) \simeq \coprod_{Y'\to X'} A/\mathscr{C}(X,X'\sqcup_B Y).$$

Dependent monoidal prisms are dual to dependent monoidal lenses. Given a symmetric monoidal category $(\mathscr{C},\otimes)$ with reflexive coequalizers that are preserved by the tensor product, let $\mathbf{CMon}_{\mathscr{C},\otimes}$ be the category of commutative monoids in $\mathrm{Ob}(\mathscr{C})$. Let $\mathbf{CAlg}_{(-)}$ and $\mathbf{Mod}_{(-)}$ be the $\mathbf{Cospan}_{\mathbf{CMon}_{\mathscr{C},\otimes}}$-indexed categories of commutative algebras and modules respectively, where functoriality is given by extension and restriction of scalars. We define the category of *dependent monoidal prisms* as follows:

$$\mathbf{DPrism}_{\mathscr{C},\otimes} := \mathbf{Optic}_{\mathbf{Mod}_{(-)},\mathbf{CAlg}_{(-)}}.$$

Morphisms can be computed via the explicit formula

$$\mathbf{DPrism}_{\mathscr{C},\otimes}\left((X,X')^A,(Y,Y')^B\right) \simeq \coprod_{Y'\to X'} A/\mathscr{C}(X,X'\otimes_B Y),$$

where the morphism $Y'\to X'$ varies among monoid homomorphisms.

3.4 Closed dependent optics

Using a technique analogous to *coalgebraic optics* [20], it is sometimes possible to explicitly compute the coend in the **Optic** category using a right adjoint technique.

Let $\mathscr{B}$ be a bicategory, and $\mathscr{L},\mathscr{R}$ be $\mathscr{B}$-indexed categories. We say that $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$ is a category of *closed dependent optics* if, for any $A,B\in\mathrm{Ob}(\mathscr{B})$ and $Y'\in\mathrm{Ob}(\mathscr{L}(B))$, the functor $(-)^*Y'\colon \mathscr{B}(A,B)\to\mathscr{R}^A$ has a right adjoint $Y'\triangleright -\colon \mathscr{R}^A\to\mathscr{B}(A,B)$. Whenever that is the case, eq. (1) can be greatly simplified.

$$\begin{aligned}\int^{f\in\mathscr{B}(A,B)} \mathscr{L}^A(X,f^*Y)\times\mathscr{R}^A(f^{*'}Y',X') &\simeq \int^{f\in\mathscr{B}(A,B)} \mathscr{L}^A(X,f^*Y)\times\mathscr{B}(f,Y'\triangleright X')\\ &\simeq \mathscr{L}^A(X,(Y'\triangleright X')^*Y).\end{aligned}$$

A possible application of this technique is based on the bicategory of bimodules [1, Ex. 2.5] **Bimod** and on the **Bimod**-indexed category $\mathbf{Mod}_{(-)}$. There, $Y'\triangleright X' = [Y',X']$, considered as an (A,B)-bimodule, hence morphisms in $\mathbf{Optic}_{\mathbf{Mod}_{(-)},\mathbf{Mod}_{(-)}}$ can be computed explicitly:

$$\mathbf{Optic}_{\mathbf{Mod}_{(-)},\mathbf{Mod}_{(-)}}\left((X,X')^A,(Y,Y')^B\right) = \mathbf{Mod}_A(X,[Y',X']\otimes_B Y).$$

4 Tambara representations

Tambara modules [19] can be useful to define an *interface* for optics that does not depend on a choice of representative. Here, we adapt the notion of *generalized Tambara module* from [7] to the dependent case, and we generalize it to an arbitrary target category. For simplicity of notation, throughout this section we fix a bicategory $\mathscr{B}$ and two $\mathscr{B}$-indexed categories $\mathscr{L}$ and $\mathscr{R}$ (with coherence isomorphisms θ,θ' respectively), and we write **Optic** instead of $\mathbf{Optic}_{\mathscr{L},\mathscr{R}}$.

Definition 3. *Let $\mathscr{D}$ be a category. A $\mathscr{D}$-valued* Tambara representation *consists of*

- *a functor $P^A\colon \left(\mathscr{L}^A\right)^{\mathrm{op}}\times\mathscr{R}^A\to\mathscr{D}$, for each object A in $\mathscr{B}$,*

- *a natural transformation* $\zeta_f\colon P^B(-,=) \Rightarrow P^A(f^*-, f^{*'}=)$*, for each morphism* $f\colon A \to B$ *in* $\mathscr{B}$*,*

where ζ_f *is extranatural in* f *and satisfies the equations*

$$P^A(\theta_A, \theta_A'^{-1}) \circ \zeta_{\mathrm{Id}_A} = \mathrm{Id}_{P^A} \quad \text{and} \quad P^A(\theta_{f,g}, \theta_{f,g}'^{-1}) \circ \zeta_{g\circ f} = (\zeta_f)_{g^*(-),g^{*'}(=)} \circ \zeta_g,$$

for all $A, B, C \in \mathrm{Ob}(\mathscr{B})$*,* $f\colon A \to B$*, and* $g\colon B \to C$*.*

Remark 4. *As the target category is arbitrary, P does not correspond to a module over an enriched category, hence we find the name* representation *more appropriate.*

In more graphical terms, when $\mathscr{D} = \mathbf{Set}$ and thus P^A, P^B are profunctors, the relationship between P^A, P^B, and ζ_f can be visualized via the following 2-cell in **Prof**.

$$\begin{array}{ccc} \mathscr{L}^B & \overset{P^B}{\longrightarrow\!\!\!\!\!\!\!+\;\;} & \mathscr{R}^B \\ {\scriptstyle f^*}\big\downarrow & \Downarrow{\scriptstyle \zeta_f} & \big\downarrow{\scriptstyle f^{*'}} \\ \mathscr{L}^A & \underset{P^A}{\longrightarrow\!\!\!\!\!\!\!+\;\;} & \mathscr{R}^A \end{array}$$

Set-valued Tambara representations can therefore be thought of as *lax* $\mathscr{B}$*-indexed profunctors*.

Definition 4. *Morphisms between Tambara representations* $(P, \zeta), (Q, \zeta')$ *are natural transformations* $\eta^A\colon P^A \Rightarrow Q^A$ *satisfying*

$$\eta^A_{f^*(-),f^{*'}(=)} \circ \zeta_f = \zeta'_f \circ \eta^B. \tag{9}$$

$\mathscr{D}$*-valued Tambara representations and their morphisms form a category, which we denote* $\mathbf{Tamb}_{\mathscr{D}}$*.*

Functors from $\mathbf{Optic}^{\mathrm{op}}$ to an arbitrary category can be described explicitly, thanks to theorem 2. The rest of the section is devoted to proving that result, via some intermediate steps, and exploring its consequences.

4.1 The universal Tambara representation

We aim to establish that all Tambara representations can be expressed as a composition of a functor with a *universal Tambara representation* ι^{op}. Here, we will define ι and prove that ι^{op} is a Tambara representation. In section 4.2, we will show universality.

Definition 5. *Let* $A \in \mathrm{Ob}(\mathscr{B})$*. The functor* $\iota^A\colon \mathscr{L}^A \times (\mathscr{R}^A)^{\mathrm{op}} \to \mathbf{Optic}$ *is defined as follows. Given an object* (X, X')*,* $\iota^A(X, X') := (X, X')^A$*. Given a morphism*

$$(l, r)\colon (X_0, X_0') \to (X_1, X_1'),$$

where $l\colon X_0 \to X_1$ *and* $r\colon X_1' \to X_0'$*, we define* $\iota^A(l, r)$ *as the optic*

$$\left\langle (\theta_A)_{X_1} \circ l \,\middle|\, r \circ (\theta_A'^{-1})_{X_1'} \right\rangle\colon (X_0, X_0')^A \to (X_1, X_1')^A$$

with representative Id_A*.*

The following lemmas will make it much easier to do computations with ι and will allow us to prove that ι^A is indeed a functor and that ι^{op} is a Tambara representation.

Lemma 2. *Let* $(X_0,X_0')^A,(X_1,X_1')^A \in \mathrm{Ob}(\textbf{Optic})$. *Let* $l_1\colon X_0 \to X_1$ *and* $r_1\colon X_1' \to X_0'$. *Then, for all optic* $\langle l_2\,|\,r_2\rangle$ *with domain* $(X_1,X_1')^A$,

$$\langle l_2\,|\,r_2\rangle \circ \iota^A(l_1,r_1) = \langle l_2\circ l_1\,|\,r_1\circ r_2\rangle. \tag{10}$$

Proof. We use twice the fact that θ_A (resp. $\theta_A'^{-1}$) is a natural transformation $\mathrm{Id}_{\mathscr{L}^A} \Rightarrow \mathrm{Id}_A^*$ (resp. $\mathrm{Id}_A^{*'} \Rightarrow \mathrm{Id}_{\mathscr{R}^A}$). Specifically,

$$\mathrm{Id}_A^*(l_2)\circ(\theta_A)_{X_1}\circ l_1 = (\theta_A)_{X_2}\circ l_2\circ l_1 = \mathrm{Id}_A^*(l_2\circ l_1)\circ(\theta_A)_{X_0}$$
$$r_1\circ(\theta_A'^{-1})_{X_1'}\circ \mathrm{Id}_A^{*'}(r_2) = r_1\circ r_2\circ(\theta_A'^{-1})_{X_2'} = (\theta_A'^{-1})_{X_0'}\circ \mathrm{Id}_A^{*'}(r_1\circ r_2).$$

As a consequence,

$$\begin{aligned}\langle l_2\,|\,r_2\rangle \circ \iota^A(l_1,r_1) &= \langle l_2\,|\,r_2\rangle \circ \big\langle (\theta_A)_{X_1}\circ l_1\,|\,r_1\circ(\theta_A'^{-1})\big\rangle_{X_1'}\\ &= \langle l_2\circ l_1\,|\,r_1\circ r_2\rangle \circ \Big\langle (\theta_A)_{X_0}\,|\,(\theta_A'^{-1})_{X_0'}\Big\rangle\\ &= \langle l_2\circ l_1\,|\,r_1\circ r_2\rangle.\end{aligned}$$

□

Lemma 3. *Let* $(Y_1,Y_1')^B,(Y_2,Y_2')^B \in \mathrm{Ob}(\textbf{Optic})$. *Let* $l_2\colon Y_1 \to Y_2$ *and* $r_2\colon Y_2' \to Y_1'$. *Then, for all optic* $\langle l_1\,|\,r_1\rangle$ *with codomain* $(Y_1,Y_1')^B$ *and representative* f,

$$\iota^B(l_2,r_2)\circ\langle l_1\,|\,r_1\rangle = \Big\langle f^*(l_2)\circ l_1\,|\,r_1\circ f^{*'}(r_2)\Big\rangle. \tag{11}$$

Proof. As f^* and $f^{*'}$ are functors, eq. (3) implies that

$$\begin{aligned}\iota^B(l_2,r_2)\circ\langle l_1\,|\,r_1\rangle &= \Big\langle (\theta_B)_{Y_2}\circ l_2\,|\,r_2\circ(\theta_B'^{-1})_{Y_2'}\Big\rangle\circ\langle l_1\,|\,r_1\rangle\\ &= \Big\langle (\theta_B)_{Y_2}\,|\,(\theta_B'^{-1})_{Y_2'}\Big\rangle\circ\Big\langle f^*(l_2)\circ l_1\,|\,r_1\circ f^{*'}(r_2)\Big\rangle\\ &= \Big\langle f^*(l_2)\circ l_1\,|\,r_1\circ f^{*'}(r_2)\Big\rangle.\end{aligned}$$

□

Proposition 6. *For all* $A\in\mathrm{Ob}(\mathscr{B})$, ι^A *is a functor.*

Proof. We must verify that ι^A preserves identity and composition of morphisms. Preservation of identity is straightforward, as

$$\iota^A(\mathrm{Id}_X,\mathrm{Id}_{X'}) = \big\langle(\theta_A)_X\,|\,(\theta_A'^{-1})_{X'}\big\rangle = \mathrm{Id}_{(X,X')^A}\,.$$

Let us now consider morphisms

$$X_0 \xrightarrow{l_1} X_1 \xrightarrow{l_2} X_2 \quad\text{and}\quad X_2' \xrightarrow{r_2} X_1' \xrightarrow{r_1} X_0'.$$

Then, using eq. (10),

$$\begin{aligned}\iota^A(l_2,r_2)\circ\iota^A(l_1,r_1) &= \Big\langle(\theta_A)_{X_2}\circ l_2\,|\,r_2\circ(\theta_A'^{-1})_{X_2'}\Big\rangle\circ\iota^A(l_1,r_1)\\ &= \Big\langle(\theta_A)_{X_2}\circ l_2\circ l_1\,|\,r_1\circ r_2\circ(\theta_A'^{-1})_{X_2'}\Big\rangle\\ &= \iota^A(l_2\circ l_1, r_2\circ r_1).\end{aligned}$$

□

Proposition 7. ι^{op} *is a* $\mathbf{Optic}^{\mathrm{op}}$*-valued Tambara representation, whose associated natural transformations are* $\left\langle \mathrm{Id}_{f^*(-)} \,|\, \mathrm{Id}_{f^{*\prime}(=)} \right\rangle^{\mathrm{op}}$ *(with representative* f*).*

Proof. By inverting all morphisms in definition 3, we work with the category $\mathbf{Optic}$ rather than $\mathbf{Optic}^{\mathrm{op}}$. Let $f\colon A \to B$. First, we need to ensure that

$$\left\langle \mathrm{Id}_{f^*(-)} \,|\, \mathrm{Id}_{f^{*\prime}(=)} \right\rangle \colon \iota^A(f^*(-), f^{*'}(=)) \Rightarrow \iota^B(-,=)$$

is a natural transformation. Let us consider morphisms $l\colon Y_0 \to Y_1$ and $r\colon Y_1' \to Y_0'$. Then, by eqs. (10) and (11),

$$\iota^B(l,r) \circ \left\langle \mathrm{Id}_{f^*Y_0} \,|\, \mathrm{Id}_{f^{*\prime}Y_0'} \right\rangle = \left\langle f^*(l) \,|\, f^{*'}(r) \right\rangle = \left\langle \mathrm{Id}_{f^*Y_1} \,|\, \mathrm{Id}_{f^{*\prime}Y_1'} \right\rangle \circ \iota^A(f^*(l), f^{*'}(r)).$$

Next, we must show that ζ_f is extranatural in f. Let us consider $f, g\colon A \rightrightarrows B$ and $m\colon f \Rightarrow g$. The following diagram commutes for all $Y \in \mathrm{Ob}(\mathscr{L}^B)$, $Y' \in \mathrm{Ob}(\mathscr{R}^B)$.

$$\begin{array}{ccc}
\iota^A(f^*Y, g^{*'}Y') & \Longrightarrow & \iota^A(f^*Y, f^{*'}Y') \\
\Downarrow & & \Downarrow \\
\iota^A(g^*Y, g^{*'}Y') & \Longrightarrow & \iota^B(Y, Y')
\end{array}$$

This can be show by direct computation, using eq. (10):

$$\begin{aligned}
\left\langle \mathrm{Id}_{g^*Y} \,|\, \mathrm{Id}_{g^{*\prime}Y'} \right\rangle \circ \iota^A(\mathscr{L}(m)_Y, \mathrm{Id}_{g^{*\prime}Y'}) &= \left\langle \mathscr{L}(m)_Y \,|\, \mathrm{Id}_{g^{*\prime}Y'} \right\rangle \\
&= \left\langle \mathrm{Id}_{f^*Y} \,|\, \mathscr{R}(m)_{Y'} \right\rangle \\
&= \left\langle \mathrm{Id}_{f^*Y} \,|\, \mathrm{Id}_{f^{*\prime}Y'} \right\rangle \circ \iota^A(\mathrm{Id}_{f^*Y}, \mathscr{R}(m)_{Y'}).
\end{aligned}$$

Finally, we need to show the coherence laws for Tambara representations. The identity law is straightforward, as

$$\left\langle \mathrm{Id}_{\mathrm{Id}_A^* X} \,|\, \mathrm{Id}_{\mathrm{Id}_A^{*\prime} X'} \right\rangle \circ \iota^A((\theta_A)_X, (\theta_A'^{-1})_{X'}) = \left\langle (\theta_A)_X \,|\, (\theta_A'^{-1})_{X'} \right\rangle = \mathrm{Id}_{(X,X')^A}\,.$$

Thanks to eq. (10)

$$\left\langle \mathrm{Id}_{(g\circ f)^*Z} \,|\, \mathrm{Id}_{(g\circ f)^{*\prime}Z'} \right\rangle \circ \iota^A((\theta_{f,g})_Z, (\theta_{f,g}'^{-1})_{Z'}) = \left\langle (\theta_{f,g})_Z \,|\, (\theta_{f,g}'^{-1})_{Z'} \right\rangle.$$

Thanks to eq. (3),

$$\left\langle \mathrm{Id}_{g^*Z} \,|\, \mathrm{Id}_{g^{*\prime}Z'} \right\rangle \circ \left\langle \mathrm{Id}_{f^*(g^*Z)} \,|\, \mathrm{Id}_{f^{*\prime}(g^{*\prime}Z')} \right\rangle = \left\langle (\theta_{f,g})_Z \,|\, (\theta_{f,g}'^{-1})_{Z'} \right\rangle.$$

Hence, the composition law holds and ι is a $\mathbf{Optic}^{\mathrm{op}}$-valued Tambara representation. □

4.2 Tambara encoding

In this section, we establish that there is a functor $[\mathbf{Optic}^{\mathrm{op}}, \mathscr{D}] \to \mathbf{Tamb}_{\mathscr{D}}$ given by composition with the universal Tambara representation ι. Furthermore, this functor is an isomorphism of categories, which we will show in theorem 2. This isomorphism will allow us to recover a classical *end formula* linking Tambara representations and optics. We start by showing that composition of a functor and a Tambara representation yields a Tambara representation.

Proposition 8. *Let $\mathscr{C}, \mathscr{D}$ be arbitrary categories. There is a functor $[\mathscr{C}, \mathscr{D}] \times \mathbf{Tamb}_{\mathscr{C}} \to \mathbf{Tamb}_{\mathscr{D}}$ given by composition.*

Proof. All conditions for Tambara representations are preserved by a functorial transformation. Given natural transformation $\mu\colon F \Rightarrow G$ and a morphism of Tambara representations $\eta\colon P \Rightarrow Q$, it is straightforward to verify that the horizontal composition of μ and η is a morphism of Tambara representations $F \circ P \Rightarrow G \circ Q$:

$$\begin{aligned}
\mu_{Q^A(X,X')} \circ F(\eta^A_{X,X'}) \circ F((\zeta_f)_{Y,Y'}) &= \mu_{Q^A(X,X')} \circ F(\eta^A_{X,X'} \circ (\zeta_f)_{Y,Y'}) \\
&= \mu_{Q^A(X,X')} \circ F((\zeta'_f)_{Y,Y'} \circ \eta^B_{Y,Y'}) \\
&= G((\zeta'_f)_{Y,Y'} \circ \eta^B_{Y,Y'}) \circ \mu_{P^B(Y,Y')} \\
&= G((\zeta'_f)_{Y,Y'}) \circ G(\eta^B_{Y,Y'}) \circ \mu_{P^B(Y,Y')} \\
&= G((\zeta'_f)_{Y,Y'}) \circ \mu_{Q^B(Y,Y')} \circ F(\eta^B_{Y,Y'}).
\end{aligned}$$

□

Theorem 2. *Let $\mathscr{D}$ be a category. Let ι^{op} be the universal Tambara representation. Then the functor*

$$- \circ \iota^{\mathrm{op}}\colon [\mathbf{Optic}^{\mathrm{op}}, \mathscr{D}] \to \mathbf{Tamb}_{\mathscr{D}}$$

is an isomorphism of categories.

Proof. Let P be a $\mathscr{D}$-valued Tambara representation, with associated natural transformations ζ_f. Let us define $\tilde{P}\colon \mathbf{Optic}^{\mathrm{op}} \to \mathscr{D}$ as follows. On objects,

$$\tilde{P}\left((X,X')^A\right) = P^A(X,X').$$

To extend $\tilde{P}$ to morphisms, we proceed as follows. As ζ_f is extranatural in f, the map

$$\begin{aligned}
\mathscr{L}^A(X, f^*Y) \times \mathscr{R}^A(f^{*'}Y', X') &\to \mathscr{D}\left(P\left((Y,Y')^B\right), P\left((X,X')^A\right)\right) \\
\langle l \,|\, r\rangle &\mapsto P^A(l,r) \circ (\zeta_f)_{Y,Y'}
\end{aligned}$$

induces a map

$$\tilde{P}\colon \mathbf{Optic}\left((X,X')^A, (Y,Y')^B\right) \to \mathscr{D}\left(P\left((Y,Y')^B\right), P\left((X,X')^A\right)\right).$$

Preservation of identity and composition follows from the coherence laws of definition 3, hence $\tilde{P}$ is a functor.

It is straightforward to verify that $\tilde{P} \circ \iota^{\mathrm{op}} = P$ as Tambara representations. On objects,

$$(\tilde{P} \circ \iota^{\mathrm{op}})^A(X,X') = \tilde{P}((X,X')^A) = P^A(X,X').$$

On morphisms, given $l\colon X_0 \to X_1$ and $r\colon X_1' \to X_0'$,

$$\begin{aligned}(\tilde{P}\circ\iota^{\mathrm{op}})^A(l,r) &= \tilde{P}(\left\langle (\theta_A)_{X_1}\circ l \,|\, r\circ(\theta_A'^{-1})_{X_1'}\right\rangle)\\ &= P((\theta_A)_{X_1}\circ l, r\circ(\theta_A'^{-1})_{X_1'})\circ(\zeta_{\mathrm{Id}_A})_{X_1,X_1'}\\ &= P(l,r)\circ P((\theta_A)_{X_1},(\theta_A'^{-1})_{X_1'})\circ(\zeta_{\mathrm{Id}_A})_{X_1,X_1'}\\ &= P(l,r).\end{aligned}$$

Finally,

$$\tilde{P}(\left\langle \mathrm{Id}_{f^*Y}\,|\,\mathrm{Id}_{f^{*\prime}Y'}\right\rangle) = P^A(\mathrm{Id}_{f^*Y},\mathrm{Id}_{f^{*\prime}Y'})\circ(\zeta_f)_{Y,Y'} = (\zeta_f)_{Y,Y'},$$

hence $\tilde{P}\circ\iota^{\mathrm{op}} = P$ as Tambara representations, so $-\circ\iota^{\mathrm{op}}$ is surjective on objects.

Let P,Q be Tambara representations, with associated natural transformation families ζ,ζ' respectively. Let $\eta: P\Rightarrow Q$ be a morphism of Tambara representations. Then, we can define a natural transformation

$$\tilde{\eta}: \tilde{P}\Rightarrow\tilde{Q} \quad \text{given by} \quad \tilde{\eta}_{(X,X')^A} := \eta^A_{X,X'}.$$

Naturality can be verified via a direct computation:

$$\begin{aligned}Q(\langle l\,|\,r\rangle)\circ\tilde{\eta}_{(Y,Y')^B} &= Q^A(l,r)\circ(\zeta_f')_{Y,Y'}\circ\eta^B_{Y,Y'}\\ &= Q^A(l,r)\circ\eta^A_{f^*Y,f^{*\prime}Y'}\circ(\zeta_f)_{Y,Y'}\\ &= \eta^A_{X,X'}\circ P^A(l,r)\circ(\zeta_f)_{Y,Y'}\\ &= \tilde{\eta}_{(X,X')^A}\circ P(\langle l\,|\,r\rangle).\end{aligned}$$

Here, we have used eq. (9) and the fact that η^A is a natural transformation from P^A to Q^A. As $\tilde{\eta}_{\iota^A(X,X')} = \tilde{\eta}_{(X,X')^A} = \eta^A_{X,X'}$, the functor $-\circ\iota^{\mathrm{op}}$ is full. Finally, if for all $A\in\mathrm{Ob}(\mathscr{B}), X\in\mathrm{Ob}(\mathscr{L}^A), X'\in\mathrm{Ob}(\mathscr{R}^A)$,

$$\eta^A_{X,X'} = \mu_{\iota^A(X,X')}$$

with $\mu: \tilde{P}\Rightarrow\tilde{Q}$, then $\mu = \tilde{\eta}$. Hence, the functor $-\circ\iota^{\mathrm{op}}$ is faithful. □

Theorem 2 has wide practical applications, as it implies that a Tambara representation P can be used as an *interface* for optics. In other words, an optic $o\in\mathbf{Optic}\big((X,X')^A,(Y,Y')^B\big)$ can be encoded as a family of morphisms

$$P^B(Y,Y')\to P^A(X,X'),$$

where P varies among $\mathscr{D}$-valued Tambara representations. This encoding has several practical advantages. It does not depend on a choice of representative, it simplifies composition of optics—replacing the rule in eq. (3) with standard function composition—and it can be used to compose optics of different types, as discussed in [7]. Furthermore, the dependent version of a classical result—the *profunctor representation theorem* [7, Thm. 4.14]—is a direct consequence of theorem 2, specialized to the case $\mathscr{D} = \mathbf{Set}$.

Lemma 4. *Let $\mathscr{C}$ be a locally small category, and let $\hat{\mathscr{C}}$ denote its category of presheaves. Then, for all $S,T\in\mathrm{Ob}(\mathscr{C})$,*

$$\mathscr{C}(S,T)\simeq\int_{F\in\hat{\mathscr{C}}}\mathbf{Set}(F(T),F(S)). \tag{12}$$

Proof. Let us consider the representable presheaf $\mathscr{C}(-,T)$. Applying the Yoneda reduction lemma [20, Lm. 1.2.2] to the evaluation-at-S functor $\hat{\mathscr{C}} \to \mathbf{Set}$, we obtain

$$\mathscr{C}(S,T) = \mathscr{C}(-,T)(S) \simeq \int_{F \in \hat{\mathscr{C}}} \mathbf{Set}(\hat{\mathscr{C}}(\mathscr{C}(-,T),F),F(S)) \simeq \int_{F \in \hat{\mathscr{C}}} \mathbf{Set}(F(T),F(S)),$$

where in the last step we have used the Yoneda lemma to compute $\hat{\mathscr{C}}(\mathscr{C}(-,T),F)$. □

Theorem 3. *[4, Thm. 3.5] Let $(X,X')^A$ and $(Y,Y')^B$ be objects in* **Optic***. Then,*

$$\mathbf{Optic}\left((X,X')^A,(Y,Y')^B\right) \simeq \int_{P \in \mathbf{Tamb_{Set}}} \mathbf{Set}\left(P^B(Y,Y'),P^A(X,X')\right). \tag{13}$$

Proof. By theorem 2, $\mathbf{Tamb_{Set}}$ is isomorphic to the category of presheaves over **Optic**. Hence, eq. (13) follows from eq. (12), with $\mathscr{C} = \mathbf{Optic}$, $S = (X,X')^A$, and $T = (Y,Y')^B$. □

5 Discussion

In this work, we developed a theory of *dependent optics* that simultaneously generalizes (mixed) optics and functor lenses. Natural examples of this construction arise from finitely complete (or finitely co-complete) categories or, more generally, from symmetric monoidal categories with reflexive equalizers (or reflexive coequalizers) preserved by the tensor product. Motivated by the practical applicability of coproducts of dependent lenses [3, 5, 8], we showed sufficient conditions under which the category of dependent optics admits finite coproducts. In [18, 21], it was shown that the category of functor lenses admits a monoidal structure whenever the underlying indexed category is monoidal. Although we did not pursue this direction here, we believe that the key ingredient used in [18]—the notion of *pseudomonoid* in the 2-category of indexed categories—can be adapted to our setting to obtain an analogous result for the category of dependent optics.

Aiming to mimic the *profunctor encoding* [7, 16] of optics, we defined a generalization of Tambara modules—*Tambara representations*. We showed that contravariant functors from **Optic** to an arbitrary category $\mathscr{D}$ can equivalently be described as $\mathscr{D}$-valued Tambara representations. Using this result, we established a *representative-free* interface for dependent optics, where each dependent optic is encoded as a polymorphic function, and recovered the *profunctor representation theorem* [7] in our setting. In the future, it will be interesting to explore in which particular cases of dependent optics this general result can be specialized to yield simplified encodings, akin to the *van Laarhoven encoding* for lenses [20].

A more general definition of optics, *fiber optics*, has been developed in [3]. Furthermore, the authors sketched a possible formalization of the *bicategory* of dependent optics to simultaneously generalize dependent lenses, mixed optics, and fiber optics. We believe that novel avenues of research can arise from the interplay between the two works. From the construction in [3, Sect. 4.3], it is straightforward to see that fiber optics are a particular case of dependent optics, as developed here, and they can probably be used to unify many examples of dependent optics. Hopefully, the work done here can help fine-tune the technical details of the definition of the bicategory of dependent optics.

Even though they appear different on the surface, our definition of dependent optics and the notion of *compound optics* [17] are equivalent (see [4, Ex. 4.2]). In our view, this has several beneficial consequences. On the one hand, our manuscript can be used to fill the gaps left in [17], such as the study of the properties of the category of dependent optics or the generalization of Tambara modules and of the profunctor representation theorem. On the other hand, the approach taken in [17] offers a different perspective on dependent optics and their composition in terms of Kan extensions, which can help form an intuitive understanding of our direct, explicit definitions. Finally, the existence of two equivalent, independently-developed definitions supports the intuition that this is indeed a principled adaptation of optics to the dependent case.

Acknowledgements

The author is grateful to Mattia G. Bergomi for helpful comments and feedback, and to Keno Fischer for interesting discussions on the interplay between the optics formalism and reverse-mode automatic differentiation. Thanks too to Bartosz Milewski for valuable feedback on the first version of the manuscript.

References

[1] Jean Bénabou (1967): *Introduction to bicategories*. In: *Reports of the midwest category seminar*, Springer, pp. 1–77, doi:10.1007/BFb0074299.

[2] Aaron Bohannon, Benjamin C Pierce & Jeffrey A Vaughan (2006): *Relational lenses: a language for updatable views*. In: *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 338–347, doi:10.1145/1142351.1142399.

[3] Dylan Braithwaite, Matteo Capucci, Bruno Gavranović, Jules Hedges & Eigil Fjeldgren Rischel (2021): *Fibre optics*. *arXiv preprint arXiv:2112.11145*, doi:10.48550/arXiv.2112.11145.

[4] Matteo Capucci (2022): *Seeing double through dependent optics*. *arXiv preprint arXiv:2204.10708*, doi:10.48550/arXiv.2204.10708.

[5] Matteo Capucci, Neil Ghani, Jérémy Ledent & Fredrik Nordvall Forsberg (2021): *Translating extensive form games to open games with agency*. *arXiv preprint arXiv:2105.06763*, doi:10.48550/arXiv.2105.06763.

[6] Aurelio Carboni, Stephen Lack & Robert FC Walters (1993): *Introduction to extensive and distributive categories*. *Journal of Pure and Applied Algebra* 84(2), pp. 145–158, doi:10.1016/0022-4049(93)90035-R.

[7] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore & Mario Román (2020): *Profunctor optics, a categorical update*. *arXiv preprint arXiv:2001.07488*, doi:10.48550/arXiv.2001.07488.

[8] Keno Fischer: *Diffractor*. Available at https://github.com/JuliaDiff/Diffractor.jl. Available from https://github.com/JuliaDiff/Diffractor.jl. With contributions from: S. Schaub, M. Abbott, M. Zgubic, O. Smith, J. Liu, M. Schanen, N. Schmitz, A. Berdyshev, V. B. Shah, J. Weidner, S. Ainsworth and F. C. White.

[9] Neil Ghani, Jules Hedges, Viktor Winschel & Philipp Zahn (2018): *Compositional game theory*. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 472–481, doi:10.1145/3209108.3209165.

[10] Michael Johnson, Robert Rosebrugh & Richard J Wood (2012): *Lenses, fibrations and universal translations*. *Mathematical Structures in Computer Science* 22(1), pp. 25–42, doi:10.1017/S0960129511000442.

[11] Peter T Johnstone (2002): *Sketches of an Elephant: A Topos Theory Compendium: Volume 2*. 2, Oxford University Press.

[12] Edward Kmett: *lens: Lenses, folds and traversals*. Available at https://hackage.haskell.org/package/lens. Available from https://hackage.haskell.org/package/lens. With contributions from: S. Ben-Kiki, E. Hird, J. Kiviniemi, B. Dirks, E. Mertens, M. Sloan, A. Altman, A. Seipp, D. Odenhall, A. Breitkreuz, S. Hengel, L. Dixon, M. Ledger, N. Haas, A. Keet, M. B. Aksenov, E. Lotem, O. Charles, L. HU, C. Schonwald, M. Wright, N. van Doorn, V. Tirronen, M. Vorozhtsov, B. Yorgey, D. Rosén, Y. Chuchem, M. Thompson, J. Wiegley, J. Fischoff, B. Larsen, A. Mason and R. Scott.

[13] Stephen Lack (2010): *A 2-categories companion*. In: *Towards higher categories*, Springer, pp. 105–191, doi:10.1007/978-1-4419-1524-5_4.

[14] Stephen Lack & Paweł Sobociński (2005): *Adhesive and quasiadhesive categories*. *RAIRO-Theoretical Informatics and Applications* 39(3), pp. 511–545, doi:10.1051/ita:2005028.

[15] Florian Marty (2009): *Des ouverts Zariski et des morphismes lisses en géométrie relative*. Ph.D. thesis, Toulouse 3.

[16] Bartosz Milewski (2017): *Profunctor optics: The categorical view.* Available from https://bartoszmilewski.com/2017/07/07/profunctor-optics-the-categorical-view/.

[17] Bartosz Milewski (2022): *Compound Optics*. *arXiv preprint arXiv:2203.12022*, doi:10.48550/arXiv.2203.12022.

[18] Joe Moeller & Christina Vasilakopoulou (2020): *Monoidal Grothendieck construction*. *Theory and Applications of Categories* 35(31), pp. 1159–1207.

[19] Craig Pastro & Ross Street (2007): *Doubles for monoidal categories*. *arXiv preprint arXiv:0711.1859*, doi:10.48550/arXiv.0711.1859.

[20] Mitchell Riley (2018): *Categories of optics*. *arXiv preprint arXiv:1809.00738*, doi:10.48550/arXiv.1809.00738.

[21] David I Spivak (2019): *Generalized Lens Categories via functors* $\mathscr{C}^{\mathrm{op}} \to \mathsf{Cat}$. *arXiv preprint arXiv:1908.02202*, doi:10.48550/arXiv.1908.02202.

Diegetic Representation of Feedback in Open Games

Matteo Capucci
University of Strathclyde

We improve the framework of open games with agency [4] by showing how the players' counterfactual analysis giving rise to Nash equilibria can be described in the dynamics of the game itself (hence *diegetically*), getting rid of devices such as equilibrium predicates. This new approach overlaps almost completely with the way gradient-based learners [6] are specified and trained. Indeed, we show feedback propagation in games can be seen as a form of backpropagation, with a crucial difference explaining the distinctive character of the phenomenology of non-cooperative games. We outline a functorial construction of arena of games, show players form a subsystem over it, and prove that their 'fixpoint behaviours' are Nash equilibria.

1 Motivation

In narratology, *diegetic* is what exists or occurs within the world of a narrative [7] (such as dialog, thoughts, etc.), as opposed to *extra-diegetic elements* which happens outside that world (such as voiceovers, soundtrack, etc.). Open games represent the situations of classical game theory in a compositional and purportedly 'diegetic' way, i.e. explicitly codifying the development of the game actions and payoff distribution phases in their specification. Hedges proposed a framework in [12] which evolved first by adopting the language of lenses [9], and then that of parametric lenses [3] to describe the bidirectional flow of information in games. In their last iteration [4, 3], *open games with agency* are defined to be given by three functions (for concreteness, we assume to work in **Set**):

$$\mathsf{play}_{\mathscr{G}} : \Omega \times X \to Y, \quad \mathsf{coplay}_{\mathscr{G}} : \Omega \times X \times R \to S \times \mho, \quad \varepsilon_{\mathscr{G}} : (\Omega \to \mho) \to P\Omega. \tag{1.1}$$

The set Ω represent *strategies*, X and Y *states* of the game, while R and S *utility* and '*coutility*', respectively. The play function has an obvious role, choosing a next state $y \in Y$ (a *move*) given the current state $x \in X$ and according to a strategy $\omega \in \Omega$. Coplay is a bit more mysterious. If we think of S and R as the type of utilities a player can expect to receive at the end of the game while at stage X and Y respectively, coplay translates between these. Finally, $\varepsilon_{\mathscr{G}}$ is a *selection function* that encodes a player's preferences: given a valuation of strategies in $\mho$ (called *costrategies* or *intrinsic utility*), $\varepsilon_{\mathscr{G}}$ returns the subset of strategies with satisfactory outcome. This data defines a parametric lens [4]:

$$\mathscr{G} = (\Omega, \mho, \varepsilon_{\mathscr{G}}, \mathsf{play}_{\mathscr{G}}, \mathsf{coplay}_{\mathscr{G}}) : (X,S) \rightleftarrows (Y,R). \tag{1.2}$$

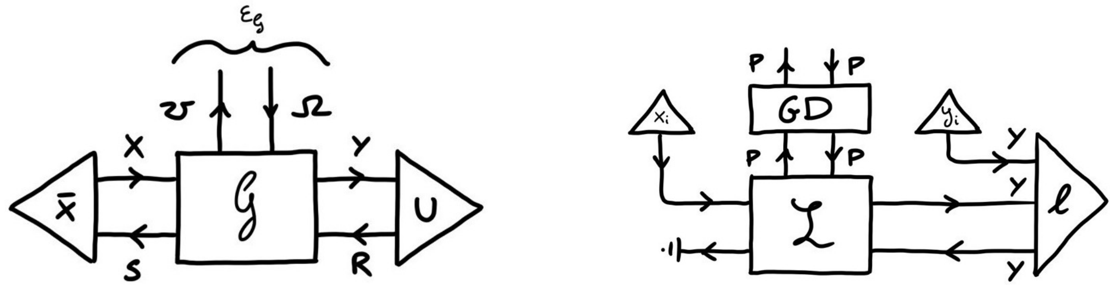
To analyse the game $\mathscr{G}$, that is, to extract its Nash equilibria, we then close the game by specifying an initial state $\bar{x} \in X$ and a payoff function $u : Y \to R$, and then apply $\varepsilon_{\mathscr{G}}$ to the composite $\bar{x} \fatsemi \mathscr{G} \fatsemi u$.

Since open games have been introduced, similar models have been proposed for learners [6] and Bayesian reasoners [22, 2], so that a general framework has been proposed in [3] to gather all these examples of 'cybernetic systems'.[1] Despite having inspired this framework, open games remain quite singular when compared to their siblings.

[1] Here we call 'cybernetic' systems having a distinguished part controlling the rest.

J. Master & M. Lewis (Eds.): Fifth International Conference on Applied Category Theory (ACT 2022). EPTCS 380, 2023, pp. 145–158, doi:10.4204/EPTCS.380.9

First of all, **their payoff dynamics lacks a well-defined role**. This shows in the way coutilities, costrategies and utilities are all different in theory but very rarely in practice, and coplay is very often simply an identity or, even worse, a discard map, which makes hard motivating the existence of a backward pass at all (see e.g. the translation process explained in [4]).



(a) (b)

Figure 1: *On the left, a gradient-based learner defined as in [3, 6], and on the right, an open game with agency as defined in [4].*

Secondly, and crucially, **the dynamics they express reflect the actions happening in the game but not the game-theoretic analysis we are actually interested in**. There's no way to know which equilibria an open game will converge to unless we pack-up the arena[2] and then feed it to the selection function. All of this happens outside of the dynamics of the game, hence *extra-diegetically*.

This issue grows into a serious conceptual flaw when we realize that according to the very notion of 'system with agency' proposed by the author and his collaborators in [3], *'open games with agency' have no agents!* In fact, agents are supposed to be systems modelled as *morphisms* plugged to the top boundary of the arena whereas in open games with agency players' preferences are embodied in the parameters, which are mere *objects* (Figure 1a). Contrast this with gradient-based learners (Figure 1b) where gradient descent, which implements the dynamics of an agent's learning, is explicitly represented *in* the system.

Contributions. In this work we correct the aforementioned problems by describing the entirety of play, payoff distribution and players' counterfactual analysis *diegetically*, thus in the dynamics of the game system itself.

We achieve this by introducing two fundamental innovations.

First, we observe that **feedback propagating in an open game has to contain information about the entirety of the payoff function of the game**, hence we replace S and R in Figure 1a with P^X and P^Y, where P is a specified payoff object. This allows to define coplay functorially from play as precomposition with a partially-evaluated play. This simple mechanism is enough to reproduce the information on payoffs available at each stage of a sequential or concurrent game. Moreover, we recognize the crucial role of the lax monoidal structure of this functor, which can be blamed for the complexity of even small game-theoretic situations.

Secondly, we describe how players are embodied inside the game by their selection functions, which are now expressed as parts of a 'reparameterisation' describing each player's optimization dynamic. This fully realizes what was already intued in [14] ('agents are their selection function') and in the drawings in [3, §6], and vindicates the ideas behind open games with agency introduced in [4]. In fact we find out

[2]The *arena* of an open game with agency is the parametric lens left after forgetting about the selection function.

the workhorse of open games with agency, the Nash product of selection functions, decomposes in three elementary parts, the key one being 'just' monoidal product of lenses.

We then show how this story shares many formal analogies with (a refinement of) gradient-based learners. There is a formal analogy between loss covectors and payoff functions, reverse derivatives and functorially-determined coplays, 'raising indices' (in the differential-geometric sense) and selection functions. Ultimately this traces out the contours of an abstract/synthetic theory of backpropagation.

Acknowledgements. We thank Jules Hedges, Philipp Zahn, Neil Ghani and Bruno Gavranović for their helpful suggestions and enthusiasm towards this work. A special mention goes to Bruno's insistence in pointing out the conceptual flaws in the use we made of selection functions in open games with agency, as well as the numerous conversations we had together on the topic, which eventually lead me to this work. Finally, we thank the ACT22 reviewers for their patience in reviewing an early version of this manuscript.

2 Diegetic open games

We start by describing our proposed notion of *diegetic open games*. As anticipated, the key idea is to recognize that in a strategic game, players have to observe the entirety of their payoff functions with other players' actions taken into account. This is done by fixing utility, coutility and intrinsic utility types to be of the form P^Y, P^X and P^Ω, representing entire payoff functions. Then such functions are propagated through the game in a way which is formally identical to backpropagation in learners, and thus amenable to the same mathematical treatement. Thus $\mathsf{coplay}_\mathscr{G}$ is actually functorially determined from $\mathsf{play}_\mathscr{G}$, as a kind of reverse derivative.

2.1 Preliminaries

Fix a finitely complete category $\mathscr{S}$. The category $\mathbf{DLens}(\mathscr{S})$ of *dependent lenses* over $\mathscr{S}$ has objects given by pairs of an object $Y : \mathscr{S}$ and a map $p : R \to Y$, and maps given by diagrams of the form:

$$\begin{array}{ccccc} S & \xleftarrow{f^\sharp} & R \times_Y X & \longrightarrow & R \\ \big\downarrow & & {\scriptstyle f^*(p)}\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle p} \\ X & = & X & \xrightarrow[f]{} & Y \end{array} \tag{2.1}$$

In the internal language of $\mathscr{S}$ [21], these maps can be denoted as $f : X \to Y$ and $f^\sharp_x : (x : X) \times R\, f(x) \to S x$. The full subcategory of $\mathbf{DLens}(\mathscr{S})$ spanned by those p which are projections is the category of *simple lenses* over $\mathscr{S}$, $\mathbf{Lens}(\mathscr{S})$. The $f^\sharp$ part of simple lenses has type $X \times R \to S$.

Dependent lenses can be built from any indexed category $F : \mathscr{S}^{\mathsf{op}} \to \mathbb{C}\mathbf{at}$, in which case we denote them by $\mathbf{DLens}(F)$. A detailed definition and intuition is given in [23].

The 2-category $\mathbf{Para}(\mathscr{S})$ [3, §2] is the strictification of the bicategory whose objects are given by objects of $\mathscr{S}$, morphisms X to Y by a choice of parameter $\Omega : \mathscr{S}$ and a map $f : \Omega \times X \to Y$, and 2-morphisms $(\Omega, f) \Rightarrow (\Xi, g) : X \to Y$ by maps $\Omega \to \Xi$ making the obvious triangle commute (see *loc. cit.*, though we have reversed the direction of 2-cells here), which are called *reparameterisations*. Composition of morphisms $(\Omega, f) : X \to Y$ and $(\Xi, g) : Y \to Z$ is given by

$$(\Xi \times \Omega, \Xi \times \Omega \times X \xrightarrow{\Xi \times f} \Xi \times Y \xrightarrow{g} Z) \tag{2.2}$$

This makes it associative only up to coherent isomorphism, hence the strictification. Same applies to the identites, which are given by $(1, 1 \times X \xrightarrow{\pi_X} X)$.

Notice the construction of $\mathbf{Para}(\mathscr{S})$ only used the cartesian monoidal structure of $\mathscr{S}$. In fact such a construction is functorial over cartesian monoidal categories. Given a lax monoidal functor [15, Definition 1.2.14] $F : \mathscr{S} \to \mathscr{T}$, with laxators $\ell_{X,Y} : F(X) \times F(Y) \to F(X \times Y)$, we get a lax 2-functor [15, Definition 4.1.2] $\mathbf{Para}(F) : \mathbf{Para}(\mathscr{S}) \to \mathbf{Para}(\mathscr{T})$ defined on objects as F and on a morphisms $(\Omega, f) : X \to Y$ as

$$\mathbf{Para}(F)(\Omega, f) = (F(\Omega), F(\Omega) \times F(X) \xrightarrow{\ell_{\Omega,X}} F(\Omega \times X) \xrightarrow{F(f)} Y). \tag{2.3}$$

Since $\ell_{\Omega,X}$ is, in principle, not invertible, this means $\mathbf{Para}(F)$ preserves composition only up to coherent non-invertible morphism. Explicitly, there is a reparameterisation $\mathbf{Para}(F)(\Omega, f) \mathbin{⨟} \mathbf{Para}(F)(\Xi, g) \Rightarrow \mathbf{Para}(F)((\Omega, f) \mathbin{⨟} (\Xi, g))$, given by $\ell_{\Xi,\Omega}$. Likewise applies to preservation of identities. The well-definedness of these reparameterisations followz from the axioms of lax monoidal structure ℓ [15, Diagram 1.2.14].

2.2 Building arenas

We now describe the most simple form of games, deterministic, complete information games, with our new machinery.

Fixing a *payoff object* P (often $P = \mathbb{R}^N$, with N the number of players), to a map $f : X \to Y$ we can associate the map $P^f : P^Y \to P^X$ given by precomposition with f. This defines a functor $P^{(-)} : \mathbf{Set} \to \mathbf{Set}^{\mathsf{op}}$, which we can lift to a lax monoidal functor

$$P^* : \mathbf{Set} \longrightarrow \mathbf{Lens}(\mathbf{Set}) \tag{2.4}$$

sending $f : X \to Y$ to $(f, \pi_2 \mathbin{⨟} P^f) : (X, P^X) \rightleftarrows (Y, P^Y)$. Abusing notation, we'll denote by $P^* f$ both this lens and its backward part, and same with objects: $P^* X := P^X$. Notice landing in lenses is crucial to give P^* a lax monoidal structure: while its unitor $\eta : (1,1) \rightleftarrows (1,P)$, given by $(1, !_P)$ would be definable anyway; the laxator $(1_{X,Y}, \mathbf{n}_{X,Y}) : (X, P^* X) \otimes (Y, P^* Y) \rightleftarrows (X \times Y, P^*(X \times Y))$, which we call *Nashator*, is defined by partial evaluation at the residuals:

$$\begin{aligned} \mathbf{n}_{X,Y} : X \times Y \times P^*(X \times Y) &\longrightarrow P^* X \times P^* Y \\ (\bar{x}, \bar{y}, u) &\longmapsto \langle u(-, \bar{y}),\ u(\bar{x}, -) \rangle \end{aligned} \tag{2.5}$$

Ideally, this functor promotes a play function into a lens obtained by canonically adding a 'coplay' function; but since play functions are actually *parametric*, we need to apply **Para** to P^* to obtain the lax 2-functor

$$\mathbf{Para}(P^*) : \mathbf{Para}(\mathbf{Set}) \longrightarrow \mathbf{Para}(\mathbf{Lens}(\mathbf{Set})) \tag{2.6}$$

so that a play function $(\Omega, \mathsf{play}_{\mathscr{G}}) : X \to Y$ is turned into a full-blown parametric lens:

$$\mathbf{Para}(P^*)(\Omega, \mathsf{play}_{\mathscr{G}}) = (\Omega, P^*\Omega,\ (1_{\Omega,X}, \mathbf{n}_{\Omega,X}) \mathbin{⨟} (\mathsf{play}_{\mathscr{G}}, P^* \mathsf{play}_{\mathscr{G}})) \tag{2.7}$$

where the backward part of the right hand side boils down to

$$\begin{aligned} \mathbf{Para}(P^*)(\Omega, \mathsf{play}_{\mathscr{G}})^\sharp : \Omega \times X \times P^* Y &\longrightarrow P^*\Omega \times P^* X \\ (\bar{\omega}, \bar{x}, u) &\longmapsto \langle u_\Omega,\ u_X \rangle \text{ where } u_\Omega = u(\mathsf{play}_{\mathscr{G}}(\bar{x}, -)) \\ & \qquad\qquad\qquad\qquad\quad u_X = u(\mathsf{play}_{\mathscr{G}}(-, \bar{\omega})) \end{aligned} \tag{2.8}$$

This definition is the workhorse of diegetic open games. Notice how u_X encapsulates $\bar{\omega}$ as a fixed parameter, so that an opponent receiving such function later has that strategy fixed. Dually, u_Ω has $\bar{x}$ fixed so the player playing at this stage can probe u by varying their own strategy but not the state the game, something determined, in turn, by other players' strategies.

Remark 2.1. A word is due regarding the opportunity of fixing a payoff object P for all games. This actually defeats the point of compositionality, as games with a different number of players would naturally require a different payoff object, and this without even mentioning how 'dangerous' it is to allow all players to observe everybody else's payoff! In fact, one can develop a better version of the theory we describe here in which **Set** is replaced by a category of 'objects with payoffs', so that we restore freedom in the payoff object we use for each game. For expositional reasons, here we stick to the simpler version in which P is fixed.

Example 2.2 (Pure sequential game)**.** Consider a very simple game in which two players make one move each, in succession. The first player has strategy space Ω and play function $(\Omega, \mathsf{play}_{\mathscr{G}}) : X \to Y$, whereas the second player has strategies Ξ and play $(\Xi, \mathsf{play}_{\mathscr{H}}) : Y \to Z$:

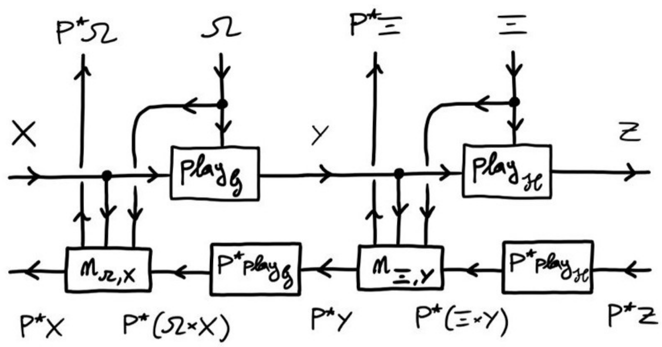


Figure 2

Figure 2 depicts the parametric lens $\mathbf{Para}(P^*)(\Omega, \mathsf{play}_{\mathscr{G}}) ⨟ \mathbf{Para}(P^*)(\Xi, \mathsf{play}_{\mathscr{H}})$. This is what we call the *arena* of the game.

Suppose a $\bar{x} \in X$ and a $u \in P^*Z$ are given, so as to close the open input horizontal wires in Figure 2. These two pieces of data amount to a so-called *context* for the game, and mathematically correspond to a further (trivially parameterised) lens $(\bar{x}, !_{P^*X}) : (1,1) \rightleftarrows (X, P^*X)$ and $(!_Z, u) : (Z, P^*Z) \rightleftarrows (1,1)$.

Then the remaining parametric lens has type $(\Xi \times \Omega, P^*\Xi \times P^*\Omega, \mathscr{A}) : (1,1) \rightleftarrows (1,1)$, which one can easily prove being equivalent to a function $\Xi \times \Omega \to P^*\Xi \times P^*\Omega$. Following $\bar{x}$ and u around the arena, one can see what this function is given by

$$(\bar{\xi}, \bar{\omega}) \mapsto \langle u_\Xi,\, u_\Omega \rangle \text{ where } \begin{aligned} u_\Xi &= \lambda \xi \,.\, u(\mathsf{play}_{\mathscr{H}}(\xi, \mathsf{play}_{\mathscr{G}}(\bar{x}, \bar{\omega})) \\ u_\Omega &= \lambda \omega \,.\, u(\mathsf{play}_{\mathscr{H}}(\bar{\xi}, \mathsf{play}_{\mathscr{G}}(\bar{x}, \omega)) \end{aligned} \tag{2.9}$$

These two functions are thus giving, to each player, all the information needed to compute their optimal strategies *given the other player's strategy*. $\mathbf{Para}(P^*)$ makes these payoff functions emerge automatically from the information flow of lenses and from the careful use of Nashators.

Payoff costates. As we've seen in the latter example, an arena needs, eventually, to be closed by a context. The data of an initial state is not particularly interesting, but we need to spend a few words on the construction of *payoff costates*. Until now, open games shared the definition of payoff function with traditional strategic games: a payoff costate $(Z,P) \rightleftarrows (1,1)$ encodes exactly the information of a payoff function $Z \to P$. Now, however, a costate has to emit not just the the payoff corresponding to a given outcome of the game, but the entire payoff function.

The most direct way to do so is to have a payoff function $u : Z \to P$ being promoted to a costate $\mathsf{const}\, u : (Z, P^*Z) \rightleftarrows (1,1)$ in **Lens**(**Set**) by

$$\mathsf{const}\, u = P^* u \fatsemi (!_P, \mathsf{const}\,\mathrm{id}) \tag{2.10}$$

where $\mathsf{const}\,\mathrm{id} : P \to P^P$ is the constant map picking the identity of P. This costate effectively ignores the outcome of the game, and returns u regardless. Alternatively, if P has the structure of a group, we can keep the information about the outcome and define

$$\Delta u = P^* u \fatsemi (!_P, \mathrm{curr}(-)) \tag{2.11}$$

where $\mathrm{curr}(-) : P \to P^P$ is the curried subtraction of P. This effectively composes to the costate corresponding to the function

$$\begin{aligned} \Delta u : Z &\longrightarrow P^* Z \\ \bar{z} &\longmapsto \lambda z.(u(z) - u(\bar{z})). \end{aligned} \tag{2.12}$$

which is a sort of 'discrete differential' of u. Eventually this would get to players as a continuation describing their possible *increment* in payoff as a function of their deviation. In traditional game theory Δu is known as *regret* [16, §3.2]. We believe it to be more conceptually convicing than the constant costate, especially as we compare games with other cybernetic systems in Section 3.

2.3 Adding players

Once an arena is built, we can add players in it. At this stage, we only deal with the 'vertical' part of a game, i.e. we draw *above* the arena (which constitutes the 'horizontal' part of a game). Here's where we specify how players team up, what they observe about each others' strategies and payoffs and, most importantly, how players process all this information to update their strategies.

The first thing to notice is that, since **Para**(P^*) is not strongly functorial, lifting the whole play function to an arena in one fell swoop versus lifting it piece by piece makes a difference in how players end up being segregated in coalitions. In fact, if $\mathsf{play}_{\mathscr{G}} : X \to Y$ and $\mathsf{play}_{\mathscr{H}} : Y \to Z$ are parameterised by Ω and Ξ respectively, then **Para**$(P^*)(\mathsf{play}_{\mathscr{G}} \fatsemi \mathsf{play}_{\mathscr{H}})$ is parameterised by $(\Xi \times \Omega,\ P^*(\Xi \times \Omega))$ whereas **Para**$(P^*)(\mathsf{play}_{\mathscr{G}}) \fatsemi$ **Para**$(P^*)(\mathsf{play}_{\mathscr{H}})$ is parameterised by $(\Xi \times \Omega,\ P^*\Omega \times P^*\Xi)$.

Effectively, **Para**$(P^*)(\mathsf{play}_{\mathscr{G}} \fatsemi \mathsf{play}_{\mathscr{H}})$ **represents a game featuring a coalition of two players** with strategy space $\Xi \times \Omega$ (hence acting as one player), while **Para**$(P^*)(\mathsf{play}_{\mathscr{G}}) \fatsemi$ **Para**$(P^*)(\mathsf{play}_{\mathscr{H}})$ **represents a game with two competing players**, with strategy spaces, respectively, Ω and Ξ.

The difference stems from the way feedback is received by players, and in their possible deviations. In the first case, the two players can evaluate joint deviations since their feedback has type $\Xi \times \Omega \to P$. In the second case, the two players can only evaluate unilateral deviations, because they receive two feedbacks $\Omega \to P$ and $\Xi \to P$ obtained by fixing either player's strategy. We turn the first to the latter by reparameterising along the Nashator $\mathbf{n}_{\Xi,\Omega} : (\Xi \times \Omega, P^*\Xi \times P^*\Omega) \Rightarrow (\Xi \times \Omega, P^*(\Xi \times \Omega))$. Thus, when used as a reparameterisation, *the Nashator breaks down coalitions of players*.

Example 2.3 (Sequential game)**.** Suppose we extend Example 2.2 with another move by the first player (decided by the same staregy space Ω, hence the copy in Figure 3. Contrary to the previous case, if we lifted the three play functions separately and then composed, we would have ended up splitting player one into two players: the long-range correlation between the first and third stage of the game forces us to lift the arena monolithically, as depicted in Figure 3.

We then reparameterise along Δ_Ω to clone the strategies of the first player into the third stage, and only then use $\mathbf{n}_{\Omega,\Xi}$ to make sure players are split into two different coalitions.

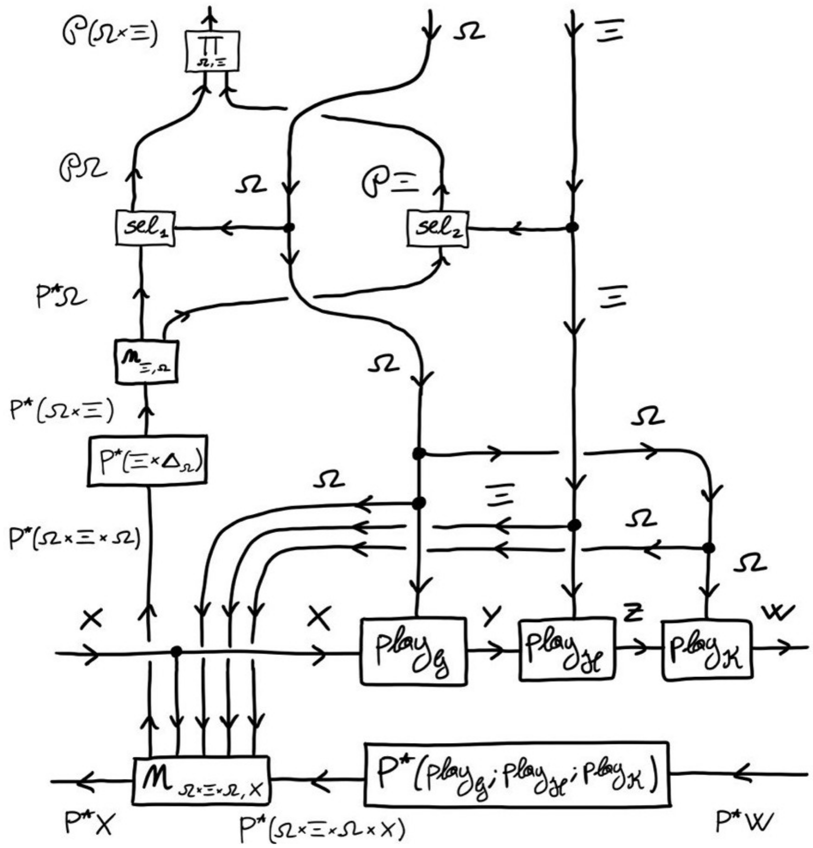


Figure 3

Remark 2.4. Observe coalitions can always be *broken* canonically, but there's no canonical way to form them. This is to be expected, since creating coalitions requires non-canonical agreements on how to distribute payoffs among its members (so-called *imputations* [16, Chapter 8]).

Finally, the last bit of the game specification concerns the process each player uses to turn the feedback they receive into *strategic deviations*. Usually, payoffs are numerical and players seek to maximize them. A bit more generally, players have some preferences encoded by a selection function $\varepsilon : P^*\Omega \to \mathscr{P}\Omega$. We warn the reader that $P^*\Omega = P^\Omega$ is the set of P-valued function to Ω, while $\mathscr{P}\Omega$ is the powerset of Ω.

A selection function fits very well in the setting we devised so far, since it has (almost) the type of the backward part of a lens $\mathsf{sel} : (\Omega, \mathscr{P}\Omega) \rightleftarrows (\Omega, P^*\Omega)$. We thus call such a lens a *selection lens*.

Remark 2.5. Notice the object $(\Omega, \mathscr{P}\Omega)$ can be considered the 'state boundary' for the player system, in the sense of [19], and betrays an implicit non-determinism in the game system. In fact, we can generalize away from sets by replacing the powerset monad $\mathscr{P} : \mathbf{Set} \to \mathbf{Set}$ with other (commutative) monads, like the Giry monad on measurable spaces (yielding stochastic games) or the tangent space monad on smooth spaces (yielding differential games).

Remark 2.6. The backward part of a selection lens is actually of the form $\mathsf{sel} : \Omega \times P^*\Omega \to \mathscr{P}\Omega$, hence a *parametric selection function*. This suggests that Ω is even more than a set of strategies, it represents the *epistemic type* of a player in the sense of Harsanyi [11], that is, an element $\omega \in \Omega$ encodes not only the way a player plans to play but also their preferences (for instance, their aversion to risk). Harsanyi's games of incomplete information, at the moment codified in the framework of open games in [1], can potentially benefit a lot from the new ideas we introduced here.

2.4 Games as systems

Let's wrap up the construction we sketched so far. The first step to specify a game is to fix the players involved (N) and their payoff type P. The arena is built canonically from a play function $\mathsf{play}_{\mathscr{G}} : \Omega \times X \to Y$, where $\Omega = \Omega_1 \times \cdots \times \Omega_N$ is the product of a strategy space per player, X is a type of initial states and Y a type of possible final outcomes of the game. Given this, we apply $\mathbf{Para}(P^*)$ to $\mathsf{play}_{\mathscr{G}}$, and get back a parametric lens $(\Omega, P^*\Omega, \mathscr{A}) : (X, P^*X) \rightleftarrows (Y, P^*Y)$, the arena.

Remark 2.7. One might object that an initial state $\bar{x} \in X$ and a utility function $\mathsf{const}\, u$ (or Δu) deserve to be part of the arena too, but experience tells this data is something to provide only when we want to move on to the analysis of the game, since closing an arena prematurely hinders further composition. The difference between a closed and an open arena is remindful of the subtle difference between a normal (resp. extensive) *form* and a normal (resp. extensive) *form game*: the latter is the data of the first plus a utility function.

Once the game arena has been built, we assemble the system of players over it. Usually, such a lens will be of the form $(\bigotimes_{i=1}^N \mathsf{sel}_i) \fatsemi \mathbf{n}_{\Omega_1,\ldots,\Omega_N}$, where $\mathsf{sel}_i : (\Omega_i, \mathscr{P}\Omega_i) \rightleftarrows (\Omega_i, P^*\Omega_i)$ are N selection lenses. Notice such a lens has domain $(\Omega, \mathscr{P}\Omega_1 \times \cdots \times \mathscr{P}\Omega_N)$, so we precompose it with

$$\prod_{i=1}^N (-) : (\Omega, \mathscr{P}\Omega) \rightleftarrows (\Omega, \mathscr{P}\Omega_1 \times \cdots \times \mathscr{P}\Omega_N), \tag{2.13}$$

which is the identity on the forward part and cartesian product[3] in the backward part (see again Figure 3).

We denote the resulting lens $(\Omega, \mathscr{P}\Omega) \rightleftarrows (\Omega, P^*\Omega)$ as $\mathscr{G}$, and this constitutes a *diegetic open game* in **Set**. Abstractly, we can consider this is a 'system with boundary $\mathscr{A}$' (Figure 4), and any such system can rightfully be called a game.

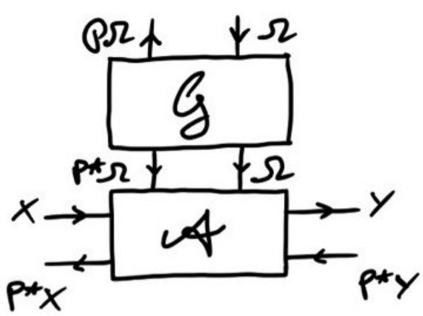


Figure 4: *The look of a generic diegetic open game, as a system $\mathscr{G}$ living over an arena $\mathscr{A}$.*

We stress this lens deserves to be called a system since its left ('top' in the drawings) boundary has a canonical form: the *deviations* $\mathscr{P}\Omega$ canonically associated to the given strategy profiles Ω (what Myers calls *changes* in [19]).

[3] Or better, the canonical lax monoidal structure of the powerset endofunctor.

Nash equilibria. So far, we never mentioned Nash equilibria. We have claimed that the way we have woven together the various pieces of a game reproduces, diegetically, the counterfactual analysis players do in a non-cooperative strategic game.

To see why our claim holds, let's analyze a game system constructed from a normal form (N,Ω), following the above recipe. Here N is a finite set of players and $\Omega = \Omega_1 \times \cdots \times \Omega_N$.

Since normal forms dispense completely with dynamical information, the associated arena will be trivial: we set $X = 1$, $Y = \Omega$[4] and $\mathsf{play}_{(N,\Omega)} := \pi_\Omega : \Omega \times 1 \to \Omega$. Hence the arena of the game is $\mathscr{A}_{(N,\Omega)} = \mathbf{Para}(\mathbb{R}^{N^*})(\mathsf{play}_{(N,\Omega)})$.

Now we focus on players. In a traditional non-cooperative game, they simply maximize their payoff, so that player i acts according to the selection lens

$$\mathsf{sel}_i = (1_{\Omega_i}, \lambda(\bar{\omega}_i, u).\operatorname{argmax}^{\mathbb{R}} u_i) : (\Omega_i, \mathscr{P}\Omega_i) \rightleftarrows (\Omega_i, \mathbb{R}^{N\times\Omega_i}). \tag{2.14}$$

We package this into a systems of players

$$\mathscr{G}_{(N,\Omega)} = \textstyle\prod_{i=1}^{N}(-) \fatsemi \left(\bigotimes_{i=1}^{N} \mathsf{sel}_i\right) \fatsemi \mathbf{n}_{\Omega_1,\dots,\Omega_N} : (\Omega, \mathscr{P}\Omega) \rightleftarrows (\Omega, \mathbb{R}^{N\times\Omega}). \tag{2.15}$$

The translation of (N,Ω) is then given by the parametric lens $(\Omega, \mathscr{P}\Omega, \mathscr{G}^*_{(N,\Omega)}\mathscr{A}_{(N,\Omega)}) : (1,1) \rightleftarrows (\Omega, \mathbb{R}^{N\times\Omega})$ obtained by plugging $\mathscr{G}_{(N,\Omega)}$ on the top boundary of $\mathscr{A}_{(N,\Omega)}$.

Theorem 2.8. *Let $(N, \Omega = \Omega_1 \times \cdots \times \Omega_N, u : \Omega \to \mathbb{R}^N)$ be an N-players, strategic game in normal form [16, Definition 1.2.1]. Let $\mathscr{G}^*(\mathscr{A}_{(N,\Omega)} \fatsemi \mathsf{const}\, u)$ be its translation to a diegetic open game, as described above, where $\mathsf{const}\, u$ has been in defined in (2.10). Let $\mathscr{G}_{(\Omega,u)} : \Omega \to \mathscr{P}\Omega$ be the set-valued function corresponding to such a closed parametric lens. Then a strategy profile $\bar{\omega} \in \Omega$ is a Nash equilibrium for (Ω, u) if and only if $\bar{\omega} \in \mathscr{G}_{(\Omega,u)}(\bar{\omega})$.*

Proof. The set-valued function equivalent to $\mathscr{G}$ is obtained by following around a given strategy profile $\bar{\omega} \in \Omega$ along the arena, which doesn't need any other input by virtue of being closed:

$$\begin{aligned}\mathscr{G}_{(\Omega,u)}(\bar{\omega}) &= \{\omega \mid \forall i \in N, \omega_i \in \operatorname{argmax}^{\mathbb{R}}(\mathbf{n}_\Omega(\bar{\omega},u)_i))\} \\ &= \{(\omega_1,\dots,\omega_N) \mid \forall i \in N, \forall \omega'_i \in \Omega_i, u_i(\bar{\omega}_1,\dots,\omega_i,\dots\bar{\omega}_N) \geq u_i(\bar{\omega}_1,\dots,\omega'_i,\dots\bar{\omega}_N)\}\end{aligned} \tag{2.16}$$

In other words, this is the set of best responses to the strategy profile $\bar{\omega}$. By definition, Nash equilibria are fixpoints of the best response function. □

In forthcoming work, we describe a principled, general framework to extract Nash equilibria as 'behaviours' of the system $\mathscr{G}$ over the arena $\mathscr{A}$, in the style of [18, 19]. Specifically, we show that Nash equilibria coincide, unsurprisingly, with non-deterministic fixpoints of such systems, i.e. simulations of the trivial game. Most importantly, from such a characterization we can automatically deduce the compositionality of equilibria which is the key strength of open games. In other words, we can show how equilibria of a composite game can be expressed in simple terms of the equilibria of its parts.

[4] Note usually this set is called A for *actions*, but we prefer to keep notation consistent.

3 Diegetic feedback as backpropagation

The conceptual story behind the diegetic representation of feedback in games is not at all specific to them. On the contrary, it opens a window on a broader conceptual story linking the categorical description of cybernetic systems featuring a 'backpropagation-like' feedback dynamics (which is most of them, notable exception being open servers [25]). Here we outline how gradient-based learners [6] share the same abstract features, in a striking example of category theory enabling a rigorous description of a previously only informal analogy.

In gradient-based learning, a smooth function $X \to Y$ is learned by optimizing a model $f : \Omega \times X \to Y$ smoothly parameterised by the variable $\omega \in \Omega$. Conceptually, this is only possible because differential structure leaks information about the loss $\ell : Y \times Y \to \mathbb{R}$ 'in a neighbourhood' of $(y, f(\omega, x))$, and this can be used to evaluate which changes in parameter the learner should implement to improve. Hence it is paramount that ℓ is known 'locally', and not just pointwise. In practice, the value of ℓ at $(y, f(\omega, x))$ is not even used! Only the covector $\mathrm{d}_{f(\omega,x)}\ell(y,-)$ is needed.

This covector is then backpropagated across the various components of the learner until a covector on Ω is obtained. As for games, this backpropagation mechanism is effortlessly assembled by deploying the functor

$$T^* : \mathbf{Smooth} \longrightarrow \mathbf{DLens}(\mathbf{Vec}_{\mathbb{R}}) \tag{3.1}$$

sending each manifold X to its *cotangent vector bundle* $T^*X \to X$ (the fiberwise dual of its tangent bundle) and each map $f : X \to Y$ to its reverse derivative, i.e. pullback of covectors along f [24], naturally expressed as a dependent lens $(f, T^*f) : (X, T^*X) \rightleftarrows (Y, T^*Y)$.[5]

Remark 3.1. In [6], a functor very similar to T^* is obtained from the structure of *reverse differential category* (RDC) on the base category, but **Smooth** is not such a category. Therefore, in *ibid.* the authors confine themselves to its wide subcategory **Euc** of Euclidean spaces. In light of our findings for games, it seems that considering functors $\mathscr{S} \to \mathbf{DLens}(\mathscr{S})$ splitting the view fibration to be more fundamental than reverse differential structure in the sense of [5]. Already in [5, §4] and [6, Proposition 2.12], it is shown how reverse differential structures can be encoded as sections of the view fibration of lenses, with extra conditions account for the 'additivity' necessary in the framework of RDCs. It seems reasonable, therefore, to reformulate RDCs as particularly nice instances of *section of feedbacks*, dualizing that of *section of changes* defined by Myers in [18, 19].

Remark 3.2. The functor T^* is strong monoidal and thus is associated to a pseudofunctor $\mathbf{Para}(T^*)$ that promotes a smooth parametric function straight into a backpropagating model. Compare this with the functor $\mathbf{Para}(P^*)$, whose laxity is, ultimately, the source of the many interesting phenomena in non-cooperative strategic games. The fact T^* is *not* lax is attributable to the additive structure involved in each fiber of a cotangent bundle, whereby $T^*(X \times Y) \cong T^*(X + Y)$.

In [20] the authors consider what amounts to a different lax monoidal structure on T^*, one with respect to the fiberwise tensor product of vector bundles.[6] That structure is strictly lax, like that of P^*. Indeed, the resulting learners behave as if they are 'competing', and this is found to be better adapted for training GANs, as their game-theoretic interpretation would suggest.

Once an arena $\mathscr{L} := \mathbf{Para}(T^*)(\Omega, f)$ has been defined, the dynamic of an agent (which is what really deserves the name of 'learner') actually doing the learning is given by a *gradient flow* lens $\mathsf{GF} : (\Omega, T\Omega) \rightleftarrows (\Omega, T^*\Omega)$ which defines a system over $\mathscr{L}$, by reparameterisation (as in Figure 1b). The

[5] Specifically, the codomain of T^* is the category of dependent lenses [23] obtained from the indexed category of smooth $\mathbb{R}$-vector bundles $\mathbf{Vec}_{\mathbb{R}} : \mathbf{Smooth}^{\mathsf{op}} \to \mathbb{C}\mathbf{at}$.

[6] This also entails replacing $\mathbf{Vec}_{\mathbb{R}}$ with its subfunctor of vector bundles and fiberwise linear maps.

backward part of such a lens is a fiberwise linear morphism $(-)^\sharp : T^*\Omega \to T\Omega$. The most common way such a morphism arises is when Ω is endowed with a Riemannian metric g, in which case $(-)^\sharp$ (known as 'raising indices' [24]) selects the direction of steepest ascent associated to a covector, so that $u^\sharp$ is $\operatorname{argmax}_{v\in T_\omega\Omega} u(v)/\|v\|_g$ for a given $u \in T^*_\omega\Omega$.

As highlighted in Table 1, $(-)^\sharp$ is formally analogous to a selection function $\mathsf{sel} : \Omega \times P^*\Omega \to \mathscr{P}\Omega$, which indeed has the same role for games. This is corroborated by the type signatures of GF and sel, both going from an object of 'states and feedbacks' to an object of 'states and changes'.

games	**gradient-based learners**
strategies Ω	parameters Ω
deviations $\mathscr{P}\Omega$	vectors $T\Omega$
payoff functions $P^*\Omega := P^\Omega$	covectors $T^*\Omega$
precomposition $P^*f : X \times P^*Y \to P^*X$	reverse derivative $T^*f : f^*(T^*Y) \to T^*X$
selection function $\mathsf{sel} : \Omega \times P^*\Omega \to \mathscr{P}\Omega$	sharp (iso)morphism $(-)^\sharp : T^*\Omega \to T\Omega$ (of vector bundles over Ω)

Table 1

What might look odd is the asymmetry between $\mathscr{P}\Omega$ and Ω in the signature of sel, something not present in $(-)^\sharp$. Indeed, if Ω is the set of 'states' of a player, then there is a dissimilarity between $T^*\Omega$ being the set of $\mathbb{R}$-valuations of $T\Omega$ and $P^*\Omega$ being the set of valuations on $\mathscr{P}\Omega$. This discrepancy requires a bit more scaffolding to be explained, but intuitively it amounts to observing $T^*\Omega$ is the set of *linear* valuations on $T\Omega$, an likewise, when we consider only maps $f : \mathscr{P}\Omega \to P$ that satisfy $f(A) = \sum_{a\in A} f(\{a\})$, these are determined by maps $\Omega \to P$.

Let us remark on another aspect, regarding discretization of such systems. Usually learners are trained with gradient *descent*, not gradient *flow*, due to the evident impossibility of actually performing an infinitesimal step in the gradient direction. Thus an important role is played by the exponential map of the Riemannian manifold of parameters, since it allows to move for a definite length along a given direction. To us, this amounts to another lens $\exp_\alpha : (\Omega, \Omega \times \Omega) \rightleftarrows (\Omega, T\Omega)$ on top of a learner, whose backward part $(\omega : \Omega) \times T_\omega\Omega \to \Omega$ is indeed given by moving for an interval of time α along the geodesic.Doing this turns the differential system GF into the deterministic and discrete GD described in [6]. In fact, this can be seen as a general move from differential to discrete given by a forward Euler integration scheme, similar to what is described in [17].

Similarly can be done for games: the analogous structure would be that of a $\mathscr{P}$-algebra.[7] Concretely, this map collapses the multiple possibilities of deviations to a choice of a next strategy to 'try'. This can be used to define a lens analogous to $\exp_\alpha$ that transforms a non-deterministic system into a deterministic one.

[7] Algebra of the $\mathscr{P}$ *endofunctor*, not necessarily the monad.

4 Conclusions

In this work we described a new approach to the specification of compositional games in the style of open games [10, 4]. It corrects some of the conceptual shortcomings of open games with agency, and uncovers deeper analogies with gradient-based learners and, speculatively, a wider range of cybernetic systems.

The new approach provides a way to specify a game using machinery analogous to reverse-mode automatic differentiation, abstractly given by a functor $P^* : \mathbf{Set} \to \mathbf{DLens}(\mathbf{Set})$. We observed how the lax monoidal structure of such functor plays a profound role in determining the dynamics of non-cooperative games, by hiding 'cooperative' information.

We have shown how classical strategic games can be naturally represented as non-deterministic systems over their arenas, systems given by the dynamics of players observing their payoffs and pondering if and how to deviate from their current strategy. The resulting parametric lens is hence a full realization of the ideas in [14, 4, 3], and brings the framework of categorical cybernetics (born with [3]) closer to that of categorical systems theory (detailed in [18, 19]).

Future directions. The new ideas brought about in this paper are not fully formed yet. In preparing this work, three more follow-up works naturally spawned.

The first, which has already been anticipated at the end of Section 2, concerns laying down a proper general theory of specification and simulation of cybernetic systems, in the wake of Myers' work on dynamical systems [18, 19]. In the first place, this would allow to extract Nash equilibria from diegetic open games in a principled and compositional way, with practical implications in the way these are computed. Secondly, using analogous tools we would then be able to talk about simulations of games and more generally of non-equilibrium trajectories of game dynamics. Lastly, we will have in place a unifying notion of 'morphism of open games', which from preliminary discussions with Hedges, seems to reproduce the most important features of those in [13] and [10].

The second work concerns the pure game-theoretic aspects of this new definition. Can we improve the toolset of compositional game theory by leveraging a more accurate reproduction of the dynamics involved? We believe the answer to be yes, with exciting connections to the topic of Bayesian games [11] and learning theory for games [8].

The third work is an exploration of the ideas roughly outlined in Section 3, with the aim of crystallizing the analogy between learners and games. Such an abstract theory of backpropagation would formalize the intuitive picture whereby such systems come with a notion of 'type of states' on which a 'type of changes', a 'type of scalars' depend, which together give rise to a 'type of feedbacks' obtained as valuations of the first in the latter.

References

[1] Joe Bolt, Jules Hedges & Philipp Zahn (2019): *Bayesian open games*. Available at https://arxiv.org/abs/1910.03656.

[2] Dylan Braithwaite & Jules Hedges (2022): *Dependent Bayesian Lenses: Categories of Bidirectional Markov Kernels with Canonical Bayesian Inversion*. Available at https://arxiv.org/abs/2209.14728.

[3] Matteo Capucci, Bruno Gavranović, Jules Hedges & Eigil Fjeldgren Rischel (2022): *Towards Foundations of Categorical Cybernetics*. Electronic Proceedings in Theoretical Computer Science 372, pp. 235–248, doi:10.4204/EPTCS.372.17. Available at http://arxiv.org/abs/2105.06332. ArXiv:2105.06332 [math].

[4] Matteo Capucci, Neil Ghani, Jérémy Ledent & Fredrik Nordvall Forsberg (2022): *Translating Extensive Form Games to Open Games with Agency*. Electronic Proceedings in Theoretical Computer Science 372, pp. 221–234, doi:10.4204/EPTCS.372.16. Available at http://arxiv.org/abs/2105.06763. ArXiv:2105.06763 [cs, math].

[5] Robin Cockett, Geoffrey Cruttwell, Jonathan Gallagher, Jean-Simon Pacaud Lemay, Benjamin MacAdam, Gordon Plotkin & Dorette Pronk (2020): *Reverse Derivative Categories*. In: 28th EACSL Annual Conference on Computer Science Logic, doi:10.4230/LIPIcs.CSL.2020.18.

[6] Geoffrey S. H. Cruttwell, Bruno Gavranović, Neil Ghani, Paul Wilson & Fabio Zanasi (2022): *Categorical Foundations of Gradient-Based Learning*. In Ilya Sergey, editor: Programming Languages and Systems, Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 1–28, doi:10.1007/978-3-030-99336-8_1.

[7] Merriam-Webster.com Dictionary: *Diegetic*. Available at https://www.merriam-webster.com/dictionary/diegetic.

[8] Drew Fudenberg, Fudenberg Drew, David K. Levine & David K. Levine (1998): *The theory of learning in games*, 1 edition. 2, MIT press.

[9] Neil Ghani, Jules Hedges, Viktor Winschel & Philipp Zahn (2018): *Compositional game theory*. In: Proceedings of the 33rd annual ACM/IEEE symposium on logic in computer science, pp. 472–481, doi:10.1145/3209108.3209165.

[10] Neil Ghani, Clemens Kupke, Alasdair Lambert & Fredrik Nordvall Forsberg (2018): *A compositional treatment of iterated open games*. Theoretical computer science 741, pp. 48–57, doi:10.1016/j.tcs.2018.05.026.

[11] John C. Harsanyi (1967): *Games with incomplete information played by "Bayesian" players, I–III Part I. The basic model*. Management science 14(3), pp. 159–182, doi:10.1287/mnsc.14.3.159. Publisher: INFORMS.

[12] Jules Hedges (2016): *Towards compositional game theory*. Ph.D. thesis, Queen Mary University of London.

[13] Jules Hedges (2018): *Morphisms of open games*. Electronic Notes in Theoretical Computer Science 341, pp. 151–177.

[14] Jules Hedges, Paulo Oliva, Evguenia Shprits, Viktor Winschel & Philipp Zahn (2017): *Higher-order decision theory*. In: International Conference on Algorithmic Decision Theory, Springer, pp. 241–254, doi:10.1007/978-3-319-67504-6_17.

[15] Niles Johnson & Donald Yau (2021): *2-dimensional Categories*. Oxford University Press, USA, doi:10.1093/oso/9780198871378.001.0001.

[16] Kevin Leyton-Brown & Yoav Shoham (2008): *Essentials of Game Theory: A Concise Multidisciplinary Introduction*. Morgan & Claypool, doi:10.1007/978-3-031-01545-8.

[17] Sophie Libkind, Andrew Baas, Evan Patterson & James Fairbanks (2022): *Operadic Modeling of Dynamical Systems: Mathematics and Computation*. Electronic Proceedings in Theoretical Computer Science 372, pp. 192–206, doi:10.4204/EPTCS.372.14. Available at http://arxiv.org/abs/2105.12282. ArXiv:2105.12282 [math].

[18] David Jaz Myers (2021): *Double Categories of Open Dynamical Systems (Extended Abstract)*. Electronic Proceedings in Theoretical Computer Science 333, pp. 154–167, doi:10.4204/EPTCS.333.11. Available at http://arxiv.org/abs/2005.05956. ArXiv:2005.05956 [math].

[19] David Jaz Myers (2022): *Categorical Systems Theory*. Available at http://davidjaz.com/Papers/DynamicalBook.pdf. (Work in progress).

[20] Florian Schaefer & Anima Anandkumar (2019): *Competitive Gradient Descent*. In: Advances in Neural Information Processing Systems, 32, Curran Associates, Inc. Available at https://proceedings.neurips.cc/paper/2019/hash/56c51a39a7c77d8084838cc920585bd0-Abstract.html.

[21] Michael Shulman (2021): *Homotopy Type Theory: The Logic of Space*. Cambridge University Press, p. 322, doi:10.1017/9781108854429.009.

[22] Toby St Clere Smithe (2021): *Compositional Active Inference I: Bayesian Lenses. Statistical Games*. Available at https://arxiv.org/abs/2109.04461.

[23] David I Spivak (2019): *Generalized Lens Categories via functors $F : \mathscr{C}^{\mathrm{op}} \to \mathsf{Cat}$*. Available at https://arxiv.org/abs/1908.02202.

[24] Michael Spivak (1973): *A comprehensive introduction to differential geometry*. *Bulletins of the American Mathematical Society* 79, pp. 303–306, doi:10.1090/S0002-9904-1973-13149-0.

[25] Andre Videla & Matteo Capucci (2022): *Lenses for Composable Servers*. Available at https://arxiv.org/abs/2203.15633.

Differential 2-rigs

Fosco Loregian
folore@ttu.ee
Tallinn University of Technology, Tallinn, Estonia*

Todd Trimble
topological.musings@gmail.com
Western Connecticut State University, Danbury, CT

We study the notion of a *differential 2-rig*, a category $\mathcal{R}$ with coproducts and a monoidal structure distributing over them, also equipped with an endofunctor $\partial : \mathcal{R} \to \mathcal{R}$ that satisfies a categorified analogue of the Leibniz rule. This is intended as a tool to unify various applications of such categories to computer science, algebraic topology, and enumerative combinatorics. The theory of differential 2-rigs has a geometric flavour but boils down to a specialization of the theory of tensorial strengths on endofunctors; this builds a surprising connection between apparently disconnected fields. We build *free 2-rigs* on a signature, and we prove various initiality results: for example, a certain category of colored species is the free differential 2-rig on a single generator.

1 Introduction

The aim of the present paper can be shortly summarized as follows: study a pair $(\mathcal{R}, \partial)$, where $\mathcal{R}$ is a 'categorified ring' and $\partial : \mathcal{R} \to \mathcal{R}$ an endofunctor preserving coproducts and satisfying the 'Leibniz rule'.

Adapting terminology from classical ring theory, such a pair $(\mathcal{R}, \partial)$ could be termed a *differential 2-rig*, and ∂ a *derivation* on $\mathcal{R}$; the study of such structures could thus be viewed as a categorified version of *differential algebra* [53, Ch. 1], an important part of modern commutative algebra [9, p. III.10], finding applications (among other areas) in Galois theory [46] and in symbolic computation [11].

Building on this, in our work, a 2-rig[1] will be a category $\mathcal{R}$ equipped with two structures, one additive and one multiplicative, such that the latter 'distributes' over the former: at its most basic level, this is the requirement that, for objects $A, B \in \mathcal{R}$, the endofunctors $A \otimes -$ and $- \otimes B$ distribute over coproducts, i.e. there are natural isomorphisms $A \otimes (B + C) \cong A \otimes B + A \otimes C$ and $(B + C) \otimes A \cong B \otimes A + C \otimes A$. Nevertheless, our main definition will be fairly more general, treating other shapes of colimits apart from this basic one.

Literature on 2-rigs. A motivating example of 'categorified calculus on a 2-rig' is Joyal's theory of species and analytic functors [28, 27, 6] providing a categorical foundation for enumerative combinatorics and finding concrete applications as a model of PCF [23].[2] The category of combinatorial species (functors $\Sigma^{\text{op}} \to \mathsf{Set}$ from the category Σ of finite sets and bijections) is a prominent example of a 2-rig which supports a viable notion of derivative functor, and it will always be our motivating example and test-bench for definitions.

This situates our work on a different ground than another important piece of literature dealing with notions of derivation on a category, namely the theory of *differential categories* of Blute, Cockett et al.

*Supported by the ESF funded Estonian IT Academy research measure (project 2014–2020.4.05.19–0001).

[1]An important difference with classical ring theory is that the request that $\mathcal{R}$ admits 'additive inverses' is an extremely restrictive one. This motivates our choice of terminology: a *rig* $(R, +, \cdot)$ –also called a *semiring*– is a ri*n*g without *n*egatives, i.e. an algebraic structure that satisfies all the axioms of a ring, but where $(R, +)$ is just a commutative monoid.

[2]The equivalence between analytic functors, regarded as 'categorified formal power series', and species is a long-established result first proved in [28]; see also [1].

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 159–182, doi:10.4204/EPTCS.380.10

[8]. Differential categories were developed to provide a categorical doctrine for *differential linear logic*; as a rule of thumb, the fundamental difference between the two approaches lies in where the categorified derivative operation acts. In differential categories, every *morphism* has a derivative assigned via a so-called differential combinator; instead, we focus on deriving *objects* functorially and naturally.[3]

Elsewhere, terms like '2-rig' or 'rig categories' have been appropriated by different authors to mean different things. For example, [3] defines a 2-rig to be a cocomplete symmetric monoidal category in which the monoidal product distributes over all colimits, and in [4], '2-rig' has meant a Vect-enriched symmetric monoidal category with biproducts and idempotent splittings (where the distributivity is automatic). On the other hand, the term 'rig category' or 'distributive category' [36] has been used to mean a category with two monoidal products, one called 'multiplicative', which distributes over the other, called 'additive'. It is easy to imagine variations on these themes: 2-rigs which are only finitely cocomplete, or that are assumed only to have *finite* coproducts (which we consider to be a baseline assumption).

Alternatively, on the multiplicative side, one might want infinite products and a complete distributivity law over infinite coproducts. This type of '2-rig' would be germane to the study of polynomial functors in the sense of [20, 31, 50, 51, 49], which have provided a unifying setting for studying numerous structures in applied category theory. Given the multiplicity of possible definitions of 2-rig, we believe it makes sense not to fix a single notion of 2-rig but to be flexible and contemplate a whole spectrum of possible theories, or 'doctrines' of $\boldsymbol{D}$*-rigs* parametrized by $\boldsymbol{D}$, a 2-monad on Cat (locally small categories) whose algebras will possess colimits of a certain shape.

Our main contributions. The first goal of this paper is to provide a generalized framework in which each of these instances can be studied on the same foot; our main definition for a 'doctrine of 2-rigs', 2.5, is geared in this direction. Besides unifying most notions of 2-rig under a common framework, in this paper we are also interested in seeing how different $\boldsymbol{D}$-rigs $\mathcal{R}$, for different doctrines $\boldsymbol{D}$, interact with an accompanying notion of derivation $\partial : \mathcal{R} \to \mathcal{R}$ (roughly, functors which obey a Leibniz rule, $\partial(A \otimes B) \cong \partial A \otimes B + A \otimes \partial B$, see 4.1). Unexpectedly, depending on the doctrine, derivations may be either virtually nonexistent (cf. 4.8 and 4.9), as is the case when the multiplicative structure is cartesian, or may exist in great plenitude, typically when the multiplicative structure enjoys a more 'linear' character (in the sense of Girard's linear logic [22]). Within a doctrine $\boldsymbol{D}$ where derivations are prevalent, they may also be used to give a notion of 'dimension of a $\boldsymbol{D}$-rig' (cf. 4.10).

In such cases, one generally expects derivations to be potent and unifying tools. We show that this is the case, once again guided by the theory of species as motivating example, and the usage of derivatives in the hands of the 'Montreal school of categorical combinatorics' [33, 37, 7, 38, 34] (see also the more recent [48, 42]), where differential equations written in the category of species, as well as their solutions, are fruitfully interpreted combinatorially. We prove that categories of species are 'necessary objects' in a general theory of 2-rigs, because they arise as free objects for specific doctrines of 2-rigs and acquire a canonical choice of a differential structure. Moreover, in 5.16 we prove that the category of species on a countable set, equipped with a 'shifting' derivation operation, is the free differential 2-rig on one generator.

Derivations can also be used to shed light on the theory of operads; for example, recent results by Obradovich [43] show that ordinary (permutative) operads are certain types of monoids for a skew

[3]For the sake of completeness, we shall mention yet another approach to 'categorical differentiation' recently developed in [54] with applications to ZX calculus in mind; again, there seems to be no relation with our theory of differential 2-rigs, since derivations on their category Mat-S are not Leibniz on objects.

monoidal structure $F' \otimes G$ defined using the derivative, and that cyclic operads [21] also admit an efficient description in terms of derivatives of species. In spite the large effort to understand the properties of a *specific* instance of differential 2-rig (see [33, 37, 7, 38, 34] for the theory of ODEs in the category of species, and a variety of works by M. Fiore [15, 17, 16] that explored in detail the meaning of *bijective proofs* in terms of datatype structures), a systematic study of general properties of differential 2-rigs (a 'synthetic 2-rig theory', so to speak) has never been attempted.

Thus, one first aim of the present paper is to give all the various notions of 2-rig and derivations their proper due, while balancing generality and applicability, and unifying diverse approaches. In developing the rudiments of this framework, we aim to clarify what is specific to the category Spc of species and what instead follows from a general theory of 2-rigs and concentrate on the latter to generalize the former (to other doctrines, other flavours of monoidality, other flavours of species –colored [41] or linear [37]).

As a showcase example, in 5.18 we present a 'chain rule' that categorifies the well-known calculus theorem $(f \circ g)'(x) = f'(g(x))g'(x)$ and that holds good across a broad spectrum of doctrines of 2-rigs, thus generalizing the chain rule true for species and proved by Joyal in his early works.

Structure of the paper. In 2 we introduce the main object of discussion of our paper: a 2-rig for a 'doctrine' $\boldsymbol{D}$, i.e. a specified class of colimits, is a category having all colimits specified by $\boldsymbol{D}$, and a monoidal structure $\otimes$ that distributes over said colimits; we show how this formalism is capable of encompassing most of the various notions of 2-rig scattered in the literature; in 3 we outline the fundamental definition in order to arrive at a definition of derivation on a $\boldsymbol{D}$-rig $\mathcal{R}$, a pair of *tensorial strengths* interacting well with each other; to the best of our knowledge, the characterization of tensorial strengths as lax natural transformations in (4) and (5) has not been accounted elsewhere. Section 4 is the heart of the paper: a differential 2-rig is defined in 4.1; after this we concentrate on the major example of combinatorial species, in 4.3, and we prove that for every $\otimes$-monoid M, its derivative ∂M is a M-module. In 5 we provide the construction of free $\boldsymbol{D}$-rigs and prove that free 2-rigs acquire differential structures (5.13) as well as various initiality results: for example, the category of (S-colored) species is the free cocomplete 2-rig on a single (on $|S|$) generator; in 6 we draw the conclusions of the paper and sketch ideas for future development: the opportunity to gain a geometric view on applicatives, through derivations on a 2-rig seems to be a promising prospect, as well as the application of our general theory to a synthetic approach to combinatorial differential equations.

2 Doctrines of 2-rigs

Before defining a notion of 2-rig doctrine, we present a few examples that play a guiding role and show a need for such a general notion.

Example 2.1 (A list of motivating examples)**.**

- The category of presheaves $[\mathcal{M}^{\text{op}}, \mathsf{Set}]$ over a monoidal category $\mathcal{M}$, equipped with the Day convolution, and its $\mathcal{V}$-enriched analogue $[\mathcal{M}^{\text{op}}, \mathcal{V}]$ [12]. Note how the convolution product tends to inherit other structures of the monoidal product $\otimes$; e.g., the Day convolution $(F,G) \mapsto F * G$ is symmetric (or braided, or cartesian monoidal) if $\otimes$ is so, and it acts as the free monoidally cocomplete [24] category on $\mathcal{M}$.
- The category of finite-dimensional vector bundles over a space or finitely generated projective modules over a commutative ring. These categories admit coproducts and tensor products, but not general colimits. Nor would one necessarily want to impose general colimits because of phenomena

like 'Eilenberg swindles' [45]. These examples of '2-rigs' are typically enriched in vector spaces or the like, and typically the only colimits envisaged are *absolute colimits*: those that are preserved by every (enriched) functor.

- Between these two extremes, one sometimes considers '2-rigs' which have colimits over diagrams that are bounded in size: for example, the categories of finite G-sets for some group G that may be infinite, or of continuous finite G-sets for some topological group G, admit only finite colimits. Or, in the theory of locally κ-presentable categories, the subcategory of compact objects will admit colimits over diagrams bounded in size by κ.

Guided by such examples, the following definitions are meant to encompass a spectrum of notions of 2-rigs that have arisen in practice.

Definition 2.2 (Additive doctrine)**.** An *additive doctrine* is a 2-category whose objects are categories that admit all colimits of diagrams belonging to a prescribed class, including at least finite discrete diagrams – whose colimits are finite coproducts, denoted with the infix +, and 0 for the empty coproduct. Morphisms of a doctrine $\boldsymbol{D}$ are functors that preserve colimits of that class, and 2-cells are natural transformations between such functors.[4]

In each case, we may instead work with a stricter notion of additive doctrine where objects are categories with *chosen* colimits: these are strict algebras of a strict 2-monad, which is often technically convenient. Strict algebra morphisms preserve those chosen colimits strictly, which is not what one wants, but pseudomorphisms preserve colimits in the usual sense [35].

So, an additive doctrine is determined by a (strict or pseudo) 2-monad $\boldsymbol{A}$ on Cat, of which we consider the category of algebras. In short, the notion of an additive doctrine takes care of the additive monoid part of a 2-rig; as for the multiplicative part, we can similarly state the following definition.

Definition 2.3 (Multiplicative doctrine)**.** A *multiplicative doctrine* is a 2-category that is monadic (in the 2-categorical sense) over the 2-category MCat_s of monoidal categories, strong monoidal functors, and monoidal transformations, such that the composition of monadic functors,

$$U_{\mathcal{M}} = (\mathcal{M} \to \mathsf{MCat}_s \to \mathsf{Cat}), \tag{1}$$

is also 2-monadic.

Intuitively, a multiplicative doctrine consists of a category of monoidal categories, possibly equipped with additional structure, that arises as the category of algebras for a monad on Cat. So, a multiplicative doctrine is given by a 2-monad $\boldsymbol{M}$ on Cat modelled over the 2-monad whose algebras are monoidal categories, of which we consider the 2-category of algebras.

The 2-category MCat of monoidal categories is trivially an example of multiplicative doctrine; so are the 2-category of symmetric, braided, or strict monoidal categories. For symmetric monoidal categories, algebra pseudomorphisms coincide with strong symmetric monoidal functors.[5] Finally, we need a notion of what it means for a multiplicative doctrine to *distribute* over an additive doctrine. Intuitively, this is taken care by a *distributive law* in the sense of [5] between the two doctrines.

[4]A more general notion of additive doctrine is obtained by considering enriched analogues as well; in this paper, we mostly focus on the unenriched (i.e., Set-enriched) case.

[5]One might also want to replace MCat_s with the 2-category MCat_l (having lax monoidal functors as 1-cells) or MCat_c (colax functors), but we do not explore such a generalization here. Also, it is well-known that the composition of monadic functors can fail to be monadic; to correct this shortcoming, various flatness conditions such as 'preserving codescent objects' may be imposed on a (2-)monadic functor $G : \mathcal{M} \to \mathsf{MCat}$ to guarantee that the composition $U_{\mathcal{M}} = UG : \mathcal{M} \to \mathsf{Cat}$ is also monadic, but this issue is somewhat technical, and it will not be pursued here.

Let $\boldsymbol{A}$ be the 2-monad for any additive doctrine in the sense above, and let $\mathbf{P}$ be the 2-monad for the additive doctrine of all small-cocomplete categories, whose underlying functor P takes a locally small category C to the category consisting of presheaves $C^{\mathrm{op}} \to \mathsf{Set}$ that are small colimits of representable functors. We have an inclusion of 2-monads $j : \boldsymbol{A} \to \mathbf{P}$. Temporarily, let $\boldsymbol{M}$ denote the 2-monad whose algebras are monoidal categories, with underlying functor M. Now, the Day convolution monoidal structure provides for each monoidal category C a monoidal structure on the free small-cocompletion on its underlying category, PUC, and this construction also works as free cocompletion for monoidal categories [24]. In other words, PUC carries a canonical $\boldsymbol{M}$-algebra structure $MPUC \to PUC$ pseudonatural in C, thus leading to an action $MPU \Rightarrow PU$ and such an action is equivalent to a canonical distributive law between monads $\delta : \boldsymbol{M}\mathbf{P} \Rightarrow \mathbf{P}\boldsymbol{M}$. The only thing required to set up this distributive law is that Day-convolving on either side, $A * -$ or $- * A$, preserves all small colimits. This remains true [52] for any restricted class of colimits coming from an additive doctrine given by a monad $\boldsymbol{A}$ on Cat; thus, we obtain by restriction a distributive law

$$\delta' : \boldsymbol{MA} \Rightarrow \boldsymbol{AM} \tag{2}$$

or what is essentially the same, a canonical lifting $\hat{\boldsymbol{A}}$ of $\boldsymbol{A}$ as follows: there is a functor $\hat{A} : \mathsf{MCat} \to \mathsf{MCat}$ such that $U \circ \hat{A} \cong A \circ U$.

Definition 2.4. A *distributivity* of a multiplicative doctrine $\mathcal{M}$ over an additive doctrine $\mathcal{A} = \boldsymbol{A}\text{-}\mathsf{Alg}$ is a choice of lift $\tilde{\boldsymbol{A}}$ in the diagram

$$\begin{array}{ccc} \mathcal{M} & \xrightarrow{\tilde{A}} & \mathcal{M} \\ \downarrow & & \downarrow \\ \mathsf{MCat} & \xrightarrow[\hat{A}]{} & \mathsf{MCat} \end{array} \tag{3}$$

Thanks to [56, Definition 33, Remark 34] such distributivities are essentially unique. This is particularly the case when the unit of the monad for the multiplicative doctrine over MCat is essentially surjective on objects (eso). In this case, the distributivity is uniquely given by the Day convolution structure at the underlying monoidal category level.

Now let $\boldsymbol{M}$ denote the monad for the monadic functor $U_{\mathcal{M}} \to \mathsf{Cat}$ in (1). As above, a distributivity $\tilde{A}$ amounts to an $\boldsymbol{M}$-action $MAU_{\mathcal{M}} \Rightarrow AU_{\mathcal{M}}$, which corresponds to a 2-distributive law $\delta : \boldsymbol{MA} \Rightarrow \boldsymbol{AM}$ between 2-monads.

Definition 2.5 (Doctrine of 2-rigs)**.** A *doctrine of 2-rigs* consists of an additive doctrine $\boldsymbol{A}$, a multiplicative doctrine $\boldsymbol{M}$, and a distributivity of $\boldsymbol{M}$ over $\boldsymbol{A}$.

Using the distributive law, one obtains a structure of 2-monad $\boldsymbol{AM}$ for the composition of functors. If one uses strict versions of the 2-monads $\boldsymbol{A}$ and $\boldsymbol{M}$, one obtains a strict 2-monad $\boldsymbol{D} = \boldsymbol{AM}$, and a strict notion of $\boldsymbol{D}$-rig, with pseudomorphisms as an appropriate notion of *morphism* of $\boldsymbol{D}$-rigs.

Remark 2.6. The original notion of 'distributive category' given by Laplaza in [36] is more general because it only asks for the presence of two monoidal structures (the 'additive' one is not necessarily cocartesian). In such a setting, the complexity of the diagrams required to ensure coherence is daunting (cf. [14, p. 2.1.1]); our choice, where 'coherence conditions' follow automatically from universal properties, avoids this problem by design.

Warning 2.7. Whereas ordinary rigs form discrete distributive categories, ordinary rigs do not give discrete 2-rigs in our sense, since the only discrete category admitting finite coproducts is the singleton. Nor is a 2-rig with a single object an exciting notion: a monoidal category with a single object is a commutative monoid; by the Eckmann-Hilton argument, the two operations of 2-rig collapse in one single commutative monoid structure, for which multiplication $a \cdot - : M \to M$ is a monoid homomorphism.

We can build on the definition of a doctrine of 2-rigs and turn our attention to some specific examples of interest, where we assume something more on the additive or multiplicative doctrine in study (symmetry, or the presence of more shapes of colimit).

Definition 2.8 (Terminological conventions for doctrines of 2-rigs)**.**

- A doctrine is *symmetric* (resp., *braided*, *cartesian*) if the underlying multiplicative doctrine is the doctrine of symmetric (resp., braided, cartesian) monoidal categories.
- A doctrine is (κ-)*cocomplete* if the class of all (κ-)small colimits gives the underlying additive doctrine.
- When the multiplicative doctrine is that of monoidal categories, then for an additive doctrine $\boldsymbol{A}$ we may refer to the 2-rigs as *monoidally* $\boldsymbol{A}$*-cocomplete* categories.
- By default, '*the*' doctrine of 2-rigs refers to the minimal notion of 2-rigs, where the multiplicative doctrine is just the doctrine of monoidal categories, and the additive doctrine is the ω-additive doctrine.
- A *closed* 2-rig is a category $\mathcal{R}$ as in 2.1 such that each $A \otimes -$ and $- \otimes B$ have right adjoints; in this case, of course, they preserve all colimits that exist in $\mathcal{R}$.

Notation 2.9. With a small abuse of language, when we refer to a 2-rig as symmetric, cocomplete, . . . , we declare that we intend to consider it as an object of a 2-rig doctrine thus designated. When necessary, we call just '2-rig' an object of the minimal 2-rig doctrine.

Example 2.10. The following are examples of 2-rigs:

RA1) Any monoidal category $(\mathcal{V}, \otimes, I)$ with the property that $\otimes$ preserves κ-ary coproducts is a monoidally κ-additive category. This includes the category of sets, any cartesian closed category with finite coproducts, the category of modules over a ring R or, more generally, the category $\mathsf{Mod}_R^{\mathcal{V}}$ of modules over a monoid R in a suitable monoidal base $\mathcal{V}$.

RA2) In the same notation, the category $[\mathcal{A}, \mathcal{V}]$ of $\mathcal{V}$-enriched presheaves over a (symmetric) monoidal $\mathcal{V}$-category $(\mathcal{A}, \oplus, j)$, endowed with the Day convolution monoidal structure is a (symmetric) closed 2-rig.

RA3) An example of a non-symmetric 2-rig is the category $[\mathcal{A}, \mathcal{A}]_+ \subseteq [\mathcal{A}, \mathcal{A}]$ of endofunctors $F : \mathcal{A} \to \mathcal{A}$ that commute with finite coproducts.

3 Modules and strengths

Just as ordinary rings and rigs act on modules, so 2-rigs or $\boldsymbol{D}$-rigs (for a 2-rig doctrine $\boldsymbol{D} = (\boldsymbol{A}, \boldsymbol{M}, \delta)$) act on 2-modules, sometimes called *actegories* (cf. [26]). For the same additive doctrine $\boldsymbol{A}$, if C is an $\boldsymbol{A}$-algebra, then we may form the endohom $[C, C]$ of $\boldsymbol{A}$-algebra maps or $\boldsymbol{A}$-cocontinuous functors, and this endohom forms a monoidally $\boldsymbol{A}$-cocomplete category. If in addition C is a $\boldsymbol{D}$-rig, then it has an underlying monoidally $\boldsymbol{A}$-cocomplete category.

Keeping this in mind, we give the following definition to capture an action of $\mathcal{R}$ on a category C as a suitable rig endomorphism.

Definition 3.1. A (left) $\mathcal{R}$-module structure (or actegory structure) on C is a monoidally $\boldsymbol{A}$-cocontinuous map $\mathcal{R} \to [C, C]$.

Example 3.2. A simple example is $\mathcal{R}$ acting on itself, so the map above takes an object R to the functor $R \otimes - : \mathcal{R} \to \mathcal{R}$. This is called the *left Cayley action*. For objects R of $\mathcal{R}$ and C of $\mathcal{C}$, we sometimes use $R \otimes C$ to denote values of left module actions. In some tautological cases, for example, the left Cayley action, we use ordinary tensor product notation $R \otimes R'$.

Remark 3.3. As a monoidal category, $\mathcal{R}$ may also be construed as a one-object bicategory $B\mathcal{R}$, and an $\mathcal{R}$-module may be construed as a pseudofunctor of bicategories $B\mathcal{R} \to \boldsymbol{A}\text{-}\mathsf{Alg}$ that is locally $\boldsymbol{A}$-cocontinuous.

In this notation, we can provide a sensible notion for a morphism of modules.

Definition 3.4. Given $\mathcal{R}$-modules $\mathcal{C}, \mathcal{D} : B\mathcal{R} \to \boldsymbol{A}\text{-}\mathsf{Alg}$, a *morphism* from $\mathcal{C}$ to $\mathcal{D}$ is a lax natural transformation $\mathcal{C} \to \mathcal{D}$.

It is worth unpacking this very terse definition. Here a lax natural transformation takes the unique object of $B\mathcal{R}$ to a 1-cell $F : \mathcal{C} \to \mathcal{D}$, in other words an $\boldsymbol{A}$-continuous functor of this form. It takes 1-cells of $B\mathcal{R}$, i.e. objects R of $\mathcal{R}$, to 2-cells which take the form of families in $\mathcal{D}$,

$$R \otimes FC \to F(R \otimes C), \tag{4}$$

that are natural in C. This 2-cell constraint is often called a *strength* on F; we call it a *left strength*. The lax naturality axioms provide the usual axioms for a tensorial strength as defined, e.g. in [30].

One can define right module structures by reversing the 1-cells of $B\mathcal{R}$, i.e., reversing the order of tensoring, $(B\mathcal{R})^{\mathrm{op}} \to \boldsymbol{A}\text{-}\mathsf{Alg}$. For example, we have a right Cayley action that takes an object R to $- \otimes R$. Then, a 2-cell constraint for a lax natural transformation between right module structures is called a *right strength*. It involves natural families, sometimes written as

$$FC \oslash R \to F(C \oslash R). \tag{5}$$

Similarly, one can define bimodules as homomorphisms $(B\mathcal{R})^{\mathrm{op}} \times B\mathcal{R} \to [\mathcal{C}, \mathcal{C}]$ (for example, there is an evident Cayley bimodule with $\mathcal{R}$ acting on itself on both the left and right), and consider bistrengths.

Example 3.5. Here is one type of example that recurs frequently for us. Suppose given a $\boldsymbol{D}$-rig map $\varphi : \mathcal{R} \to \mathcal{S}$. This induces a homomorphism $\varphi^{\mathrm{op}} \times \varphi : B\mathcal{R}^{\mathrm{op}} \times B\mathcal{R} \to B\mathcal{S}^{\mathrm{op}} \times B\mathcal{S}$, which composes with the Cayley bimodule of $\mathcal{S}$. Letting $\alpha_{\mathcal{R}}$, $\alpha_{\mathcal{S}}$ denote the Cayley bimodules, the data of a morphism from $\alpha_{\mathcal{R}}$ to $\alpha_{\mathcal{S}}(\varphi^{\mathrm{op}} \times \varphi)$ entails an $\boldsymbol{A}$-cocontinuous functor $G : \mathcal{R} \to \mathcal{S}$ with a ('φ-augmented') left and right strength $\varphi(R) \otimes GR' \to G(R \otimes R')$ and $GR' \otimes \varphi(R) \to G(R' \otimes R)$.

4 Differential 2-rigs: basic theory

We now turn to the main definition of the present paper, that of a *derivation* on a 2-rig; in simple terms, if a 2-rig categorifies the notion of ri(n)g R, a derivation on a 2-rig categorifies the notion of derivation on R, widely used in commutative algebra and finding applications to the Galois theory of differential equations (see [46, 47]).

Definition 4.1 (Derivation on a 2-rig)**.** Let $\boldsymbol{D}$ be a 2-rig doctrine, let $\mathcal{R}$ be a $\boldsymbol{D}$-rig, and let $\mathcal{M}$ be a $\mathcal{R}$-bimodule. An $\mathcal{M}$-valued *derivation* of $\mathcal{R}$ is a bimodule morphism ∂ from the Cayley bimodule of $\mathcal{R}$ (cf. 3.2) to $\mathcal{M}$, such that the canonical natural maps

$$\mathfrak{l} : \partial C \oslash C' + C \otimes \partial C' \to \partial(C \otimes C') \qquad \mathfrak{i} : 0 \to \partial(I) \tag{6}$$

are isomorphisms.

We will refer to the map $\mathfrak{l}$ above as the *leibnizator* map of the derivation.

Here the first arrow is defined by pairing the module right strength $\partial C \oslash C' \to \partial(C \otimes C')$ with the module left strength $C \otimes \partial C' \to \partial(C \otimes C')$.

Definition 4.2 (Differential $\boldsymbol{D}$-rig)**.** A *differential* $\boldsymbol{D}$*-rig* is a $\boldsymbol{D}$-rig $\mathcal{R}$ equipped with a derivation from the Cayley bimodule of $\mathcal{R}$ to itself.

Example 4.3. A paradigmatic example of a differential 2-rig is given by the category of Joyal species with its standard derivative functor, sending $F : \Sigma^{\mathrm{op}} \to \mathsf{Set} : n \mapsto Fn$ to $F' : n \mapsto F(n+1)$, where $n \in \Sigma$ is an n-element set.

In this example, the doctrine is that of symmetric monoidally cocomplete categories, and the category of species is the free symmetric monoidally cocomplete category on one object, i.e. the category of finite sets and bijections. This is the category of presheaves $\Sigma^{\mathrm{op}} \to \mathsf{Set}$ on the category of finite sets and bijections, equipped with the Day convolution product induced from the monoidal product on Σ.

Besides the Leibniz rule, whose validity can be proved via elementary methods, the differential structure in the category of species satisfies two additional properties reminiscent of formal power series theory.[6] If S,T,U,V are objects of Σ, we can prove the following result (cf. [2, §8.11] and [57, §4.5.4]). (We provide a proof of this and of 4.5 appear in Appendix B, page 180.)

Proposition 4.4 (Generalised Leibniz rule for species)**.** Let ∂ be the standard derivation on species. We can think of the n-th derivative $\partial^n F$ as a derivative 'with respect to a n-element set U', since in case $|U| = n$ one has $\partial^n F[A] = F[A+n] \cong F[A+U]$. Define $F^{(U)}$ by the formula $F^{(U)}[A] = F[A+U]$. Now, let $F,G : \Sigma \to \mathsf{Set}$ be two combinatorial species; we have

$$(F * G)^{(U)}[C] \cong \sum_{S+T=U} (F^{(S)} * G^{(T)})[C]. \tag{7}$$

Theorem 4.5 (A Taylor-Maclaurin formula for species)**.** Every species $F : \Sigma \to \mathsf{Set}$ has a 'Taylor-Maclaurin' expansion

$$F(X+A) \cong \int^{n} F(A+n) \times X^n \cong \int^{n \in \mathbf{P}} \partial^n F(A) \times X^n. \tag{8}$$

The name of this result is motivated by the fact that when the coend in (8) is unwound, we end up with the Taylor expansion $F(X+A) \cong \sum_{n=0}^{\infty} \frac{\partial^n F(A)}{n!} X^n$.

There is a notion of morphism of differential 2-rig, and a notion of morphism of derivations: together, these define the category 2-Rig of differential 2-rigs, and the category $\mathsf{Der}(\mathcal{R},\partial)$ of derivations on a given 2-rig. We will not investigate 2-categorical properties of 2-Rig, but the notion of morphism of derivation is necessary to turn 5.10 into an equivalence of categories instead of just a correspondence on objects.

Definition 4.6 (Morphism of differential 2-rigs)**.** Given differential 2-rigs $(\mathcal{R},\partial) \to (\mathcal{S},\partial')$, morphisms of differential 2-rigs are morphisms of 2-rigs $F : \mathcal{R} \to \mathcal{S}$ such that $\partial' \circ F = F \circ \partial$.

Definition 4.7 (Morphism of derivations)**.** Let $\mathcal{R}$ be a 2-rig, and $\partial,\partial' : \mathcal{R} \to \mathcal{R}$ two derivations in the sense of 4.1. A *morphism of derivations* $\alpha : \partial \Rightarrow \partial'$ is a natural transformation of functors such that the equality of 2-cells $\mathfrak{l}' \circ (\alpha \otimes 1 + 1 \otimes \alpha) = (\alpha * \otimes) \circ \mathfrak{l}$ holds if $\mathfrak{l}$ (resp., $\mathfrak{l}'$) is the leibnizator of ∂ (resp., ∂').

[6] Given the elementary nature of their proof, we believe both these results pertain to 'folklore' in circles of combinatorialists, but we could not find an appropriate reference for them.

Now we observe how some notions bearing on 2-rigs, particularly property-like notions for the multiplicative monoidal product, make sense independent of which doctrine of 2-rigs is considered. For example, a 2-rig (relative to any 2-rig doctrine) is *cartesian* if its multiplicative monoidal product is cartesian and is *closed* if tensoring with an object on either side has a right adjoint.

A derivation $\partial : \mathcal{R} \to \mathcal{R}$ is *trivial* if it is constantly 0. A *∂-constant* object is such that $\partial X \cong 0$. Clearly, a derivation is trivial if and only if every object is a ∂-constant. The description of derivations on a 2-rig in terms of tensorial strengths leads to two fundamental 'no-go theorem' for derivations on a 2-rig: the proofs appear one after the other in Appendix B, page 177.

Proposition 4.8. A derivation $\partial : \mathcal{R} \to \mathcal{R}$ on a cartesian 2-rig must be trivial.

Proposition 4.9. Suppose that $\mathcal{R}$ is a closed 2-rig and that the functor $\mathcal{R}(I,-) : \mathcal{R} \to \mathsf{Set}$ is faithful. Then any functor $\partial : \mathcal{R} \to \mathcal{R}$ can carry at most one left strength and one right strength.

Remark 4.10. The intuition behind 4.8 and 4.9 is that in certain categories, every object arises as a coequalizer of maps between coproducts of copies of I. If derivations preserve colimits and take I to 0, then, of course, every object maps to 0.

This allows for yet another analogy with differential/algebraic geometry: categories satisfying 4.8, 4.9 are 'categories of constants' hence are '0-dimensional' from the point of view of categorified 'dimension theory'.

The connection between derivations on 2-rigs and tensorial strengths deserves to be spelt out more explicitly: to this end, we provide a general procedure to turn every endofunctor $F : \mathcal{R} \to \mathcal{R}$ on a 2-rig into a derivation.

The universal construction of tensorial strengths

This subsection shows that to every endofunctor $F : \mathcal{A} \to \mathcal{A}$ one can associate another endofuctor ΘF, carrying the structure of a cofree coalgebra for a comonad Θ on $[\mathcal{A},\mathcal{A}]$. This, in turn, follows from the fact that there is a comonad Θ (resp., Θ^{R}) on the category $[\mathcal{A},\mathcal{A}]$ of endofunctors of a monoidal category $\mathcal{A}$, that equips an endofunctor F with a cofree left (resp., right) tensorial strength. The proof appears in Appendix B, page 179.

Proposition 4.11. Let $\mathcal{A}$ be a complete and left (resp., right) symmetric monoidal closed category; then, there exists a comonad Θ on the category $[\mathcal{A},\mathcal{A}]$ of endofunctors of $\mathcal{A}$, whose coalgebras are exactly the endofunctors equipped with a right (resp., left) tensorial strength.

Remark 4.12. This result entails that given an endofunctor $F : \mathcal{R} \to \mathcal{R}$ on a symmetric 2-rig, F is 'best-approximated' by an endofunctor ΘF that we might think as a 'lax derivation' (by this we mean a fairly weak concept: it's a functor equipped with both a left and right tensorial strength, and as a consequence with a noninvertible leibnizator map $\left[\begin{smallmatrix} t \\ t^{\mathrm{R}} \end{smallmatrix}\right] : \Theta FA \otimes B + A \otimes \Theta FB \to F(A \otimes B)$), obtained by endowing the functor F it with the cofree strength

$$\Theta FA \otimes B \xrightarrow{t_{AB}} \Theta F(A \otimes B) \xleftarrow{t^{\mathrm{R}}_{AB}} A \otimes \Theta FB \tag{9}$$

using the universal property of coproducts, now one gets the desired map $\left[\begin{smallmatrix} t \\ t^{\mathrm{R}} \end{smallmatrix}\right]$.

The result remains true when the 2-rig $\mathcal{R}$ is not symmetric, but a little more care is needed; in that case, one must define Θ_{R} (resp., Θ) exploiting a right (resp., left) closed structure on $\mathcal{A}$.

Given any category C and a morphism $f : X \to Y$ one can consider the category $C[f^{-1}]$ obtained as the 'smallest' category where f becomes an isomorphism (cf. [19, p. 1.1]). In this light, the relevance of this result lies in the fact that one can then formally invert the map $\left[\begin{smallmatrix} t \\ t^{\mathrm{R}} \end{smallmatrix}\right]$ above and endow $\mathcal{R}$ with a derivation canonically obtained from the pair $(\mathcal{R}, F)$.

We conclude the section by turning our attention to the following result, which to the best of our knowledge is new, despite the simplicity of its proof: let M be an internal $\otimes$-monoid in a differential 2-rig $(\mathcal{R}, \otimes, \partial)$; the derivative ∂M is an M-module. The proof appears in Appendix B, page 177.

Proposition 4.13. Let $\mathcal{R}$ be a 2-rig, and M a internal semigroup (resp., monoid) in $\mathcal{R}$, with multiplication $m : M \otimes M \to M$ (and unit $e : I \to M$); then the map $\partial m : \partial M \otimes M + M \otimes \partial M \to \partial M$ amounts to a pair of actions $i_R : \partial M \otimes M \to \partial M$ and $i_L : M \otimes \partial M \to \partial M$ of M on its derivative object ∂M.

5 The construction of free 2-rigs

In this subsection, we turn our attention to constructions of derivations and differentials, restricting focus to symmetric 2-rig doctrines $\boldsymbol{D}$. Our main technique is to exploit the representability of derivations in the sense of 5.1 and 5.2.

There are several reasons for restricting to symmetric 2-rigs $\mathcal{R}$. First, in ordinary algebra, the vast majority of applications of derivations are to commutative algebra; categorifying, it is then natural to consider symmetric monoidal structures. Moreover, tensoring functors $R \otimes - : \mathcal{R} \to \mathcal{R}$ carry canonical (co)strengths, on account of the symmetry.

In the symmetric case, we can turn any left $\mathcal{R}$-module $\mathcal{M}$ into a right module or a bimodule by defining $M \oslash R$ to be $R \otimes M$. We call bimodules arising this way *symmetric*. (Here, it seems pointless to distinguish between $\otimes$ and $\oslash$, so we write $\otimes$ instead.)

Definition 5.1 (Square-zero extensions)**.** Let $\mathcal{R}$ be a $\boldsymbol{D}$-rig, and let $\mathcal{M}$ be a symmetric $\mathcal{R}$-bimodule. Define the *square-zero extension* $\mathcal{R} \ltimes \mathcal{M}$ of $\mathcal{M}$ to be $\mathcal{R} \times \mathcal{M}$ as an $\boldsymbol{A}$-algebra, and equipped with a symmetric monoidal product defined by the formula

$$(A, M) \boxtimes (B, N) := (A \otimes B, A \otimes N + M \otimes B), \tag{10}$$

and with monoidal unit $(I, 0)$. The first projection $\pi : \mathcal{R} \ltimes \mathcal{M} \to \mathcal{R}$ makes this a $\boldsymbol{D}$-rig over $\mathcal{R}$.

A straightforward computation allows determining the associators and unitors for the $\boxtimes$ monoidal structure (one must use the compatibility between the left and right module structure on $\mathcal{M}$) and the left and right distributive maps.

An alternative presentation of the square zero extension, in the case where $\mathcal{M}$ is the Cayley bimodule of $\mathcal{R}$ acting on itself, can be given as a 'quotient' 2-rig $\mathcal{R}[Y]/(Y^2)$: a categorification of an algebra of 'dual numbers', as explained in the following subsection. This 2-rig is denoted $\mathcal{R}[\varepsilon]$.

Proposition 5.2. For a $\boldsymbol{D}$-rig $\mathcal{S}$ over $\mathcal{R}$, say $\psi : \mathcal{S} \to \mathcal{R}$, there is a natural equivalence between maps $\Phi : \mathcal{S} \to \mathcal{R} \ltimes \mathcal{M}$ in $\boldsymbol{D}$-$\mathsf{Rig}/\mathcal{R}$, and ψ-augmented derivations ∂ of $\mathcal{S}$ valued in $\mathcal{M}$, where $\partial = \pi_2 \Phi : \mathcal{S} \to \mathcal{M}$.

The proof is fairly routine since (ψ, ∂) being a (strong) symmetric monoidal functor means that we obtain isomorphisms $\partial(S) \otimes \psi(S') + \psi(S) \otimes \partial(S') \cong \partial(S \otimes S')$ whose restrictions to the summands satisfy the strength coherence conditions, on account of the coherence conditions that obtain for a symmetric monoidal functor.

For example, we can use this proposition to reconstruct the standard derivative on Joyal species Spc, working over the doctrine $\boldsymbol{D}$ of symmetric monoidally cocomplete categories. Consider Spc as a Cayley bimodule over itself, and form $\mathsf{Spc}[\varepsilon] = \mathsf{Spc} \ltimes \mathsf{Spc}$.

As Spc is the free symmetric monoidally cocomplete category on one generator X (the representable functor $\Sigma(-,1)$), there is an equivalence of categories

$$\boldsymbol{D}\text{-}\mathsf{Rig}(\mathsf{Spc},\mathsf{Spc}[\varepsilon]) \simeq \mathsf{Spc}[\varepsilon]. \tag{11}$$

This means any object (F,G) whatsoever of $\mathsf{Spc}[\varepsilon]$ induces a $\boldsymbol{D}$-rig map $\Phi_{(F,G)} : \mathsf{Spc} \to \mathsf{Spc}[\varepsilon]$, hence (by the Proposition) a ψ-augmented derivation for some $\boldsymbol{D}$-rig map $\psi : \mathsf{Spc} \to \mathsf{Spc}$. Let us be more explicit. First we calculate $\psi = \pi\Phi_{(F,G)} : \mathsf{Spc} \to \mathsf{Spc}$. The pseudonaturality of the equivalence $\boldsymbol{D}\text{-}\mathsf{Rig}(\mathsf{Spc},\mathcal{R}) \simeq \mathcal{R}$ means $\pi\Phi_{(F,G)}$ is the unique (*essentially* unique, i.e., unique up to unique isomorphism) symmetric monoidally cocontinuous functor $\psi_F : \mathsf{Spc} \to \mathsf{Spc}$ that carries X to F. Proceeding in stages, the functor $F : 1 \to \mathsf{Spc}$ extends essentially uniquely to a symmetric monoidal functor $\tilde{F} : \Sigma \to \mathsf{Spc}$, taking n to the n-fold Day convolution $F^{\otimes n}$. Then this extends essentially uniquely to a *cocontinuous* symmetric monoidal functor $\mathsf{Spc} = [\Sigma^{\mathrm{op}},\mathsf{Set}] \to \mathsf{Spc}$, according to the formula

$$W = \int^{n:\Sigma} W(n)\cdot\Sigma(-,n) \;\mapsto\; \int^{n:\Sigma} W(n)\cdot F^{\otimes n}. \tag{12}$$

The last coend is an instance of the *substitution* product of species, denoted $W \circ F$. Whatever it is, the point is that $\psi = (-)\circ F$, where the right side is the essentially unique $\boldsymbol{D}$-rig map $\mathsf{Spc} \to \mathsf{Spc}$ that extends $F : 1 \to \mathsf{Spc}$. In particular, if F is the generator X, then ψ_X is the identity on Spc.

Now a derivation $\partial : \mathsf{Spc} \to \mathsf{Spc}$ augmented by the identity is just an ordinary derivation, i.e., satisfies $\partial(A\otimes B) \cong \partial(A)\otimes B + A\otimes\partial(B)$. The composite $1 \xrightarrow{X} \mathsf{Spc} \xrightarrow{\langle \mathrm{id},\partial\rangle} \mathsf{Spc}[\varepsilon] \xrightarrow{\pi_2} \mathsf{Spc}$ is the component G of (F,G), whereas the composition of the last two arrows is ∂. In other words, $G = \partial(X)$. If we want ∂ to match the standard derivative of species, then we must have $G = X' = I$, the unit of Day convolution.

Therefore, under the natural equivalence of the proposition, the standard derivative of species corresponds to the $\boldsymbol{D}$-rig map $\mathsf{Spc} \to \mathsf{Spc}[\varepsilon]$ that takes the generator X to (X,I). If we take X to some other element (X,G) instead, then the corresponding derivation ∂ is defined by $\partial(F) = F'\otimes G$, because this is after all a derivation, and because $\partial(X) \cong X'\otimes G \cong G$ is correct. Note then that every differential structure, i.e., every derivation on Spc augmented over the identity, is obtained by tensoring the standard derivative by some object.

Remark 5.3. In the analogy between species F,G and formal power series f,g, the substitution product corresponds to functional substitution $(f\circ g)(x) = f(g(x))$. The derivative of a substitution can be computed via the *chain rule*, known since Joyal [27]:

$$(F\circ G)' = (F'\circ G)\otimes G'. \tag{13}$$

We will provide a proof for the chain rule, formulated not only for species but valid in any $\boldsymbol{D}$-rig, in Appendix B, page 180.

Presentations of $\boldsymbol{D}$-rigs

Here we provide a construction of free $\boldsymbol{D}$-rigs and give a few sample constructions of other $\boldsymbol{D}$-rigs. We freely employ the definitions we have introduced so far, and in particular 2.2, 2.3. Our main result, 5.9, is guided by an analogy with classical algebra: to provide a presentation of an ordinary rig is tantamount to providing a coequalizer of two maps between free rigs since rigs form a category 2-monadic over Set.

The fact that under mild assumptions on $\boldsymbol{D}$ –for example, if its multiplicative monad $\boldsymbol{M}$ is finitary– the 2-category $\boldsymbol{D}\text{-}\mathsf{Rig}$ has bicolimits, ensures that similar such constructions exist and can provide presentations of 2-rigs as suitable 2-dimensional colimits [29] of diagrams of free 2-rigs.

If $\boldsymbol{A}$ denotes the monad on Cat for the additive doctrine, then for a category C, the $\boldsymbol{A}$-cocompletion $\boldsymbol{A}(C)$ is equivalent to the full subcategory of the small presheaf category $P(C)$ obtained by taking the closure of the representable functors under the class of $\boldsymbol{A}$-colimits.

Remark 5.4. Using the distributive law, the monad for $\boldsymbol{D}$ is the composite $\boldsymbol{AM}$. Hence, for *every* doctrine $\boldsymbol{D}$, the free $\boldsymbol{D}$-rig $\boldsymbol{D}[C]$ on a category C is always formed according to a simple two-step procedure: first, take the free multiplicative structure generated by C, i.e. the category $\boldsymbol{M}(C)$. Then, take the free $\boldsymbol{A}$-cocompletion of $\boldsymbol{M}(C)$.

We have already seen an example of this in the case of Joyal species (in the doctrine $\boldsymbol{D}$ of symmetric monoidally cocomplete categories): it is the free cocompletion $[\Sigma^{\mathrm{op}}, \mathsf{Set}]$ of the free symmetric monoidal category Σ on a single generator. Likewise, we may define multivariate species, say for example species in two variables, as the category $[\Sigma(2)^{\mathrm{op}}, \mathsf{Set}]$ equipped with Day convolution, where incidentally $\Sigma(2)$ is equivalent to $\Sigma \times \Sigma$.

For the remainder of this section, we return to symmetric 2-rigs (relative to some additive doctrine $\boldsymbol{A}$), and proceed to categorify some commutative algebra. The 2-category of $\boldsymbol{A}$-algebras, being a 2-category of algebras for a KZ-monad, carries a monoidal product $\odot$ (see [18]) characterized by the fact that for $\boldsymbol{A}$-algebras $\mathcal{A}$, $\mathcal{B}$, C, functors $\mathcal{A} \times \mathcal{B} \to C$ that are $\boldsymbol{A}$-cocontinuous in the separate $\mathcal{A}$-, $\mathcal{B}$-arguments are equivalent to $\boldsymbol{A}$-cocontinuous functors $\mathcal{A} \odot \mathcal{B} \to C$.

Proposition 5.5 (Coproduct of $\boldsymbol{D}$-rigs)**.** Using the universal property one can show that if $\mathcal{R}$, $\mathcal{S}$ are $\boldsymbol{D}$-rigs, meaning here symmetric monoidally $\boldsymbol{A}$-cocomplete categories, then $\mathcal{R} \odot \mathcal{S}$ naturally acquires a $\boldsymbol{D}$-rig structure and is the coproduct of $\mathcal{R}$ and $\mathcal{S}$ in $\boldsymbol{D}$-Rig.

Notation 5.6 (Extension of scalars)**.** In particular, let $\mathcal{S} = \boldsymbol{D}[Y]$ be the free $\boldsymbol{D}$-rig on a single generator Y. We write $\mathcal{R} \odot \boldsymbol{D}[Y]$ as $\mathcal{R}[Y]$; this plays a role analogous to a polynomial rig $C[Y]$ with coefficients in a rig C, and the construction is analogous to the 'extension of scalars' from the initial rig $\mathbb{N}[Y]$ to the rig $C[Y]$ obtained as a coproduct in the category of rigs.

Remark 5.7. The formation of $\mathcal{R}[Y]$ does not require working with symmetric 2-rigs: just as one can form a polynomial algebra $R[x]$ over a noncommutative rig R, so one can form a 'polynomial' 2-rig $\mathcal{R}[Y]$ over a monoidal 2-rig $\mathcal{R}$, by taking a tensor product $\mathcal{R} \odot \boldsymbol{D}[Y]$. However, this tensor product will generally not be a coproduct in $\boldsymbol{D}$-Rig if we work outside the symmetric context.

Kähler differentials

Next, we sketch the construction of a $\boldsymbol{D}$-rig of Kähler differentials on a $\boldsymbol{D}$-rig $\mathcal{R}$. Again, we borrow ideas from the analogous construction in algebraic geometry. Let $\boldsymbol{D}[Y]$ be the free $\boldsymbol{D}$-rig on a single generator $\{Y\}$, treated as a generic 'indeterminate'.

Let $0 : \boldsymbol{D}[Y] \to \boldsymbol{D}[Y]$ denote the essentially unique $\boldsymbol{D}$-rig morphism that takes Y to 0, and similarly let $Y^2 : \boldsymbol{D}[Y] \to \boldsymbol{D}[Y]$ denote the morphism that takes Y to $Y^{\otimes 2}$. The unique map $0 \to Y^2$ in $\boldsymbol{D}[Y]$ transports across the equivalence

$$\boldsymbol{D}\text{-Rig}(\boldsymbol{D}[Y], \boldsymbol{D}[Y]) \simeq \boldsymbol{D}[Y] \tag{14}$$

to a symmetric monoidal natural transformation $0 \Rightarrow Y^2$ between $\boldsymbol{D}$-rig maps $0, Y^2 : \boldsymbol{D}[Y] \to \boldsymbol{D}[Y]$. Extending scalars like in 5.6, we obtain a 2-cell in $\boldsymbol{D}$-Rig:

$$\mathcal{R}[Y] \underset{Y^2}{\overset{0}{\rightrightarrows}} \!\!\Downarrow\; \mathcal{R}[Y] \xrightarrow{\;q\;} \mathcal{R}[Y]/(Y^2) \tag{15}$$

The 'quotient' construction $q : \mathcal{R}[Y] \to \mathcal{R}[Y]/(Y^2)$ we are after is a coinverter of this 2-cell in the 2-category $\boldsymbol{D}$-Rig. In fact, diagram (15) satisfies precisely the universal property of a *coinverter* ([29, dual of (4.6)]) when we adopt for $\mathcal{R}[Y]/(Y^2)$ the concrete model deduced from 5.14: each object of $\mathcal{R}[Y]/(Y^2)$ is of the form $A + B \otimes Y$ we prove this in Appendix B, page 181.

Remark 5.8. Observe that for some 2-rig doctrines $\boldsymbol{D}$, this coinverter may be somewhat degenerate. For example, in the doctrine of cartesian 2-rigs (for any additive doctrine $\boldsymbol{A}$), the condition that an arrow $0 \to C^2$ is invertible in $\mathcal{R}$ forces $C \cong 0$ (because C is a retract of C^2), and in this case, the coinverter will be the 2-rig map $\mathcal{R}[Y] \to \mathcal{R}$ taking Y to 0 (cf. the fact that there are no nontrivial differentials on a cartesian 2-rig).

Proposition 5.9. For a doctrine $\boldsymbol{D}$ of symmetric 2-rigs, there is an equivalence $\mathcal{R}[Y]/(Y^2) \simeq \mathcal{R} \ltimes \mathcal{R}$.

In combination with 5.2, this means that $\mathcal{R}[\varepsilon] = \mathcal{R}[Y]/(Y^2)$, equipped with the evident $\boldsymbol{D}$-rig map $\mathcal{R}[\varepsilon] \to \mathcal{R}$ taking Y to 0, represents augmented derivations.

Corollary 5.10. There is an equivalence of categories

$$\mathsf{Der}(\mathcal{R},\mathcal{R}) \cong 2\text{-}\mathsf{Rig}(\mathcal{R},\mathcal{R}[Y]/(Y^2)) \tag{16}$$

or in other words, the category of derivations $\mathcal{R} \to \mathcal{R}$ as in 4.7 correspond to 2-rig morphisms $\mathcal{R} \to \mathcal{R}[Y]/(Y^2)$. More generally, there is an equivalence between derivations $\mathcal{R} \to \mathcal{M}$ values in a $\mathcal{R}$-module $\mathcal{M}$, and algebra morphisms between $\mathcal{R}$ and the square-zero extension of 5.1.

The construction of free $\boldsymbol{D}$-rigs and 5.10 allow to provide examples of differentials on categories of multivariate (or 'colored', cf. [41]) species.

Definition 5.11 (Partial derivative)**.** Let $\boldsymbol{D}[S]$ be the free $\boldsymbol{D}$-rig on a set or discrete category of generators S. For $s \in S$, define the *partial derivative*

$$\frac{\partial}{\partial s} : \boldsymbol{D}[S] \to \boldsymbol{D}[S] \tag{17}$$

to be the derivation that corresponds to the $\boldsymbol{D}$-rig map $\boldsymbol{D}[S] \to \boldsymbol{D}[S][\varepsilon]$ that takes s to (s, I) and $t \in S$, $t \neq s$, to $(t, 0)$.[7]

Every differential on $\boldsymbol{D}[S]$ is similarly formed from the $\boldsymbol{D}[S]$-rig maps $\boldsymbol{D}[S] \to \boldsymbol{D}[S][\varepsilon]$ taking each s to (s, a_s) for some choice of 'coefficients' $a_s \in \boldsymbol{D}[S]$. In the case where the additive doctrine admits arbitrary coproducts, this differential may be denoted

$$\partial = \sum_{s \in S} a_s \frac{\partial}{\partial s}. \tag{18}$$

Here is one more example of a differential 2-rig, bearing witness that differential structures on a symmetric 2-rig tend to be plentiful. The idea goes as follows: let $\boldsymbol{D}[X,Y]$ be the free $\boldsymbol{D}$-rig over two generators; given any two polynomials $p(X,Y), q(X,Y)$ we can build the 'quotient 2-rig' killing off the 'ideal' generated by $\{p,q\}$ as a suitable 2-colimit.

Example 5.12. We consider the 2-rig $\mathcal{H} := \boldsymbol{D}[X,Y]/(Y^2 + 1 \cong X^2)$ where we categorify the coordinate ring of an hyperbola. Here we have two morphisms $\boldsymbol{D}[T] \to \boldsymbol{D}[X,Y]$ to the free $\boldsymbol{D}$-rig on two generators,

[7] One can prove that the 'Schwarz-Clairaut's theorem' of commutativity of composition of derivatives with respect different 'indeterminates'. We refrain to provide such a proof in detail, as it is completely straightforward.

one taking T to Y^2+1, the other taking T to X^2; to form $\boldsymbol{D}[X,Y]/(Y^2+1\cong X^2)$, construct a co-iso-inserter ([29, 10]) between these two $\boldsymbol{D}$-rig maps.

The differential $\partial : \mathcal{H} \to \mathcal{H}$ is defined by $\partial(X) = Y$, $\partial(Y) = X$, and taking the co-iso-inserter $\varphi : Y^2+1 \to X^2$ to a canonical isomorphism $\partial(Y^2+1) \to \partial(X^2)$ obtained as follows:

$$\begin{aligned}\partial(Y^2+1) &\cong \partial(Y^2)+\partial(1) \cong \partial(Y^2) \cong \partial Y \otimes Y + Y \otimes \partial Y \\ &\cong X \otimes Y + Y \otimes X \xrightarrow{\sigma+\sigma} Y \otimes X + X \otimes Y = \partial X \otimes X + X \otimes \partial X \cong \partial(X^2) \quad (19)\end{aligned}$$

where σ denotes an instance of the symmetry isomorphism.

Proposition 5.13 (Free 2-rigs are differential)**.** The free 2-rig $\Sigma[Y]$ and its cocompletion $\Sigma[\![Y]\!]$ with respect to arbitrary coproducts both admit at least one nontrivial derivation, which is uniquely determined by the request that the 'generator' Y goes to the monoidal (Day convolution) unit.

From the universal property of $\mathcal{R}[Y]$, we deduce that it is the category generated under coproducts by formal expressions $A_n \otimes Y^n$ where $n \geq 0$ is an integer and $A_n \in \mathcal{R}$.

Proposition 5.14. Every object in the differential 2-rig $\mathcal{R}[Y]$ admits a unique representation as a formal sum like $\sum_{i=0}^{d} A_i \otimes Y^i$.

Proof. In Appendix B, page 178. □

A particularly interesting example of a free 2-rig construction as differential 2-rig is where S is a countable set whose elements we denote $\{Y, Y^{(1)}, Y^{(2)} \dots, Y^{(n)}, \dots\}$, that we interpret as the stock of all subsequent derivatives of a unique indeterminate Y. In other words, we construct a differential $\partial : \boldsymbol{D}[S] \to \boldsymbol{D}[S]$ via the $\boldsymbol{D}$-rig map

$$\boldsymbol{D}[S] \to \boldsymbol{D}[S][\varepsilon] \quad (20)$$

that takes $Y^{(i)}$ to $(Y^{(i)}, Y^{(i+1)})$, in effect defining $\partial(Y^{(i)}) = Y^{(i+1)}$. This construction has a parallel in differential algebra, see e.g. [47, Ch. 1]. Hence we obtain, by 'scalar extension' (tensoring with $\mathcal{R}$)

Example 5.15 (The 2-rig of differential polynomials)**.** We can define the *2-rig of differential polynomials (with coefficients in a 2-rig $\mathcal{R}$)* using an infinite set of 'indeterminates' $\mathcal{Y} := \{Y = Y^{(0)}, Y^{(1)}, Y^{(2)} \dots, Y^{(n)}, \dots\}$ as above, and defining the 2-rig $\mathcal{R}[Y^\partial]$ as the free 2-rig of polynomials over $\mathcal{Y}$. This is a differential 2-rig where the differential ∂ takes every 'constant' $C \odot I \in \mathcal{R} \odot \boldsymbol{D}[\mathcal{Y}]$ to 0, and $\partial(Y^{(i)})$ to $Y^{(i+1)}$.

The 2-rig $\boldsymbol{D}[Y^\partial]$ defined above enjoys the following universal property: given a differential $\boldsymbol{D}$-rig $\mathcal{S}$ and an element $A \in \mathcal{S}$, there exists a unique morphism of differential 2-rigs $\bar{X} : \Sigma[Y^\partial] \to \mathcal{S}$ with the property that $Y \mapsto A$. In other words,

Theorem 5.16. The free $\boldsymbol{D}$-rig of polynomials $\Sigma[Y^\partial]$ of 5.15 is the free differential 2-rig on a single generator $\{Y\}$.

Remark 5.17. A slightly different way to put this theorem is: the monad $\mathbf{E}$ on Cat, whose algebras are categories $\mathcal{R}$ equipped with an endofunctor $D : \mathcal{R} \to \mathcal{R}$, distributes over the 2-rig monad $\boldsymbol{AM}$ according to the Leibniz rule.[8] If $\boldsymbol{D}$ is a symmetric 2-rig doctrine, then the free differential $\boldsymbol{D}$-rig on a set of generators S is $\bigodot_{s\in S} \boldsymbol{D}[Y_s^\partial] = \boldsymbol{D}[\{Y_s^{(i)}\}_{s\in S, i\in\mathbb{N}}]$.

[8] Intuitively, treat D as a differential operator so that D applied to a polynomial operator can be rewritten as a polynomial operator applied to D.

We conclude the section concentrating on the proof of a *chain rule* on free $\boldsymbol{D}$-rigs. If, following [28], we shall think about combinatorial species as categorified formal power series, a 'chain rule' of the form $(f \circ g)'(x) = f'(g(x))g'(x)$ shall hold; it follows from an easy computation that this is the case when the substitution $F \circ G$ is interpreted as a *substitution product* (cf. for example [6, §1.4]). The present subsection provides a conceptual argument proving a chain rule valid for an abstract symmetric 2-rig doctrine.

Let $\boldsymbol{D}$ be a symmetric 2-rig doctrine, and recall equation (14). To each object G of $\boldsymbol{D}[1]$, there is a corresponding $\boldsymbol{D}$-rig map denoted $- \circ G : \boldsymbol{D}[1] \to \boldsymbol{D}[1]$. Indeed, endofunctor composition on the left side $\boldsymbol{D}$-Rig$(\boldsymbol{D}[1], \boldsymbol{D}[1])$ transports to a monoidal structure on $\boldsymbol{D}[1]$ which, by abuse of notation, we denote as $\circ : \boldsymbol{D}[1] \times \boldsymbol{D}[1] \to \boldsymbol{D}[1]$; variously called the *substitution* monoidal product or *plethystic* monoidal product [41]. The unit for the substitution product is the generator $X : 1 \to \boldsymbol{D}[1]$.

The standard derivative $\partial : \boldsymbol{D}[1] \to \boldsymbol{D}[1]$ is defined by $\partial(X) = \boldsymbol{I}$, i.e., is given by the unique $\boldsymbol{D}$-rig map $\boldsymbol{D}[1] \to \boldsymbol{D}[1][\varepsilon]$ that takes X to (X, I). The proof of the chain rule appears in Appendix B, page 180.

Theorem 5.18. Given species F, G, there is a canonical isomorphism $(F \circ G)' = (F' \circ G) \otimes G'$.

6 Conclusions and future work

We introduced the notion of differential 2-rig as a unifying structure for many diverse instances of a category equipped with a 'derivation', an endofunctor that satisfies the Leibniz property.

The link between the Leibniz property for an endofunctor and a pair of tensorial strengths thereon hints at a connection between differential structures and *applicative* structures, widely used in functional programming [40, 44]. Given the 'geometric' flavour of differential 2-rig theory, this is a surprising connection between apparently disconnected fields that will be further investigated.

Another enticing future direction of investigation involves *differential equations*: one can define a 'differential polynomial endofunctor' (DPE) in a similar fashion in which polynomial functors are defined inductively (cf. [25, §2.2]), by declaring that all polynomial expressions $\sum_{i=0}^{n} A_i \otimes \partial^i$ obtained from a differential $\partial : \mathcal{R} \to \mathcal{R}$ on a 2-rig form the category $\mathsf{DPE}(\mathcal{R}, \partial)$. The theory of differential equations in the category of species has a long and well-established history: it was mostly developed by Leroux and Viennot [39, 38, 55, 7] Labelle [32] and other authors built on that [13, 42]. The general theory of combinatorial differential equations studied in these papers might fruitfully be framed into a more general theory of DPEs and their solutions.

References

[1] J. Adámek and J. Velebil. "Analytic functors and weak pullbacks". In: *Theory Appl. Categ.* 21 (2008), No. 11, 191–209 (cit. on p. 159).

[2] M. Aguiar and S. Mahajan. "Monoidal Functors, Species and Hopf Algebras". In: CRM Monograph Series 29. American Mathematical Society, Providence, R.I., 2010. Chap. 15 (cit. on p. 166).

[3] J. Baez and J. Dolan. "Higher-Dimensional Algebra III. n-Categories and the Algebra of Opetopes". In: *Advances in Mathematics* 135.2 (1998), pp. 145–206. DOI: 10.1006/aima.1997.1695 (cit. on p. 160).

[4] J. Baez, J. Moeller, and T. Trimble. *Schur Functors and Categorified Plethysm*. 2021. DOI: 10.48550/arXiv.2106.00190 (cit. on p. 160).

[5] J. Beck. "Distributive laws". In: *Seminar on triples and categorical homology theory*. Springer. Germany: Springer, 1969, pp. 119–140 (cit. on p. 162).

[6] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial species and tree-like structures*. Vol. 67. Cambridge, UK: Cambridge University Press, 1998. DOI: 10.1017/CBO9781107325913 (cit. on pp. 159, 173).

[7] F. Bergeron and C. Reutenauer. "Combinatorial Resolution of Systems of Differential Equations III: a Special Class of Differentially Algebraic Series". In: *European Journal of Combinatorics* 11 (1990), pp. 501–512. DOI: 10.1016/S0195-6698(13)80035-2 (cit. on pp. 160, 161, 173).

[8] R. Blute, J. Cockett, and R. A. Seely. "Cartesian differential categories". In: *Theory and Applications of Categories* 22.23 (2009), pp. 622–672 (cit. on p. 160).

[9] N. Bourbaki. "Algèbres tensorielles, algèbres extérieures, algèbres symétriques". In: *Algèbre: Chapitres 1 à 3*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 379–596 (cit. on p. 159).

[10] J. Bourke. "Codescent objects in 2-dimensional universal algebra". PhD thesis. School of Mathematics and Statistics, University of Sydney, 2010 (cit. on p. 172).

[11] F. Chyzak and B. Salvy. "Non-commutative Elimination in Ore Algebras Proves Multivariate Identities". In: *Journal of Symbolic Computation* 26.2 (1998), pp. 187–227 (cit. on p. 159).

[12] B. Day. "On Closed Categories of Functors". In: *Reports of the Midwest Category Seminar, IV*. Lecture Notes in Mathematics, Vol. 137. Berlin: Springer, Berlin, 1970, pp. 1–38. DOI: 10.1007/BFb0060438 (cit. on p. 161).

[13] H. Décoste, G. Labelle, and P. Leroux. "Une approche combinatoire pour l'itération de Newton–Raphson". In: *Advances in Applied Mathematics* 3.4 (1982), pp. 407–416. DOI: 10.1016/S0196-8858(82)80013-4 (cit. on p. 173).

[14] J. Elgueta. "The groupoid of finite sets is biinitial in the 2-category of rig categories". In: *Journal of Pure and Applied Algebra* 225.11 (2021), p. 106738. DOI: 10.1016/j.jpaa.2021.106738 (cit. on p. 163).

[15] M. Fiore. "Mathematical Models of Computational and Combinatorial Structures". In: *Foundations of Software Science and Computational Structures*. Ed. by V. Sassone. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 25–46. DOI: 10.1007/978-3-540-31982-5_2 (cit. on p. 161).

[16] M. Fiore. "Discrete Generalised Polynomial Functors". In: *Automata, Languages, and Programming*. Germany: Springer Berlin Heidelberg, 2012, pp. 214–226. DOI: 10.1007/978-3-642-31585-5_22 (cit. on p. 161).

[17] M. Fiore. *An Axiomatics and a Combinatorial Model of Creation/Annihilation Operators*. 2015. eprint: arXiv:1506.06402 (cit. on p. 161).

[18] I. L. Franco. "Pseudo-commutativity of KZ 2-monads". In: *Advances in Mathematics* 228.5 (2011), pp. 2557–2605. DOI: 10.1016/j.aim.2011.06.039 (cit. on p. 170).

[19] P. Gabriel and M. Zisman. *Calculus of fractions and homotopy theory*. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 35. Springer-Verlag New York, Inc., New York, 1967, pp. x+168 (cit. on p. 168).

[20] N. Gambino and J. Kock. "Polynomial functors and polynomial monads". In: *Mathematical proceedings of the cambridge philosophical society*. Vol. 154. Cambridge, UK: Cambridge University Press, 2013, pp. 153–192. DOI: 10.1017/S0305004112000394 (cit. on p. 160).

[21] E. Getzler and M. Kapranov. *Cyclic operads and cyclic homology*. 1995 (cit. on p. 161).

[22] J.-Y. Girard. "Linear logic". In: *Theoretical Computer Science* 50.1 (1987), pp. 1–101. DOI: 10.1016/0304-3975(87)90045-4 (cit. on p. 160).

[23] R. Hasegawa. "Two applications of analytic functors". In: *Theoretical Computer Science* 272.1 (2002). Theories of Types and Proofs 1997, pp. 113–175. DOI: 10.1016/S0304-3975(00)00349-2 (cit. on p. 159).

[24] G. Im and G. Kelly. "A universal property of the convolution monoidal structure". In: *Journal of Pure and Applied Algebra* 43.1 (1986), pp. 75–88. DOI: 10.1016/0022-4049(86)90005-8 (cit. on pp. 161, 163).

[25] B. Jacobs. *Introduction to Coalgebra*. Vol. 59. Cambridge, UK: Cambridge University Press, 2017. DOI: 10.1017/CBO9781316823187 (cit. on p. 173).

[26] G. Janelidze and G. Kelly. "A note on actions of a monoidal category". In: *Theory Appl. Categ* 9.61-91 (2001), p. 02 (cit. on p. 164).

[27] A. Joyal. "Une théorie combinatoire des séries formelles". In: *Adv. Math. (NY)* 42.1 (1981), pp. 1–82 (cit. on pp. 159, 169).

[28] A. Joyal. "Foncteurs analytiques et especes de structures". In: *Combinatoire énumérative*. Vol. 1234. Lecture Notes in Math. Germany: Springer, Berlin, 1986, pp. 126–159. DOI: 10.1007/BFb0072514 (cit. on pp. 159, 173).

[29] G. Kelly. "Elementary observations on 2-categorical limits". In: *Bulletin of the Australian Mathematical Society* 39 (1989), pp. 301–317. DOI: 10.1017/S0004972700002781 (cit. on pp. 169, 171, 172).

[30] A. Kock. "Strong functors and monoidal monads". In: *Archiv der Mathematik* 23.1 (1972), pp. 113–120 (cit. on p. 165).

[31] J. Kock. "Polynomial functors and combinatorial Dyson–Schwinger equations". In: *Journal of Mathematical Physics* 58.4 (2017), p. 041703. DOI: 10.1063/1.4977012 (cit. on p. 160).

[32] G. Labelle. "Une combinatoire sous-jacente au théoreme des fonctions implicites". In: *Journal of Combinatorial Theory, Series A* 40.2 (1985), pp. 377–393. DOI: 10.1016/0097-3165(85)90097-4 (cit. on p. 173).

[33] G. Labelle. "On combinatorial differential equations". In: *Journal of Mathematical Analysis and Applications* 113.2 (1986), pp. 344–381. DOI: 10.1016/0022-247X(86)90310-0 (cit. on pp. 160, 161).

[34] G. Labelle. "Combinatorial Directional Derivatives and Taylor Expansions". In: *Discrete Math.* 79.3 (1990), 279–297. DOI: 10.1016/0012-365X(90)90336-G (cit. on pp. 160, 161).

[35] S. Lack. "A 2-categories companion". In: *Towards higher categories*. Vol. 152. IMA Vol. Math. Appl. New York: Springer, 2010, pp. 105–191. DOI: 10.1007/978-1-4419-1524-5_4 (cit. on p. 162).

[36] M. Laplaza. "Coherence for distributivity". In: *Coherence in categories*. Germany: Springer, 1972, pp. 29–65. DOI: 10.1007/BFb0059555 (cit. on pp. 160, 163).

[37] P. Leroux and G. X. Viennot. "Combinatorial resolution of systems of differential equations, I. Ordinary differential equations". In: *Combinatoire énumérative*. Ed. by G. Labelle and P. Leroux. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 210–245. DOI: 10.1007/BFb0072518 (cit. on pp. 160, 161).

[38] P. Leroux and G. X. Viennot. "Combinatorial resolution of systems of differential equations. IV. separation of variables". In: *Discrete Mathematics* 72.1 (1988), pp. 237–250. DOI: 10.1016/0012-365X(88)90213-0 (cit. on pp. 160, 161, 173).

[39] P. Leroux and G. X. Viennot. "Combinatorial resolution of systems of differential equations. IV. separation of variables". In: *Discrete Mathematics* 72.1 (1988), pp. 237–250. DOI: 10.1016/0012-365X(88)90213-0 (cit. on p. 173).

[40] C. McBride and R. Paterson. "Applicative programming with effects". In: *J. Funct. Program.* 18 (2008), pp. 1–13. DOI: 10.1017/S0956796807006326 (cit. on p. 173).

[41] M. Méndez and O. Nava. "Colored species, c-monoids, and plethysm, I". In: *Journal of combinatorial theory, Series A* 64.1 (1993), pp. 102–129. DOI: 10.1016/0097-3165(93)90090-U (cit. on pp. 161, 171, 173).

[42] M. Menni. "Combinatorial functional and differential equations applied to differential posets". In: *Discrete Mathematics* 308.10 (2008), pp. 1864–1888. DOI: 10.1016/j.disc.2007.04.035 (cit. on pp. 160, 173).

[43] J. Obradović. "Cyclic operads: syntactic, algebraic and categorified aspects". 2017USPCC191. PhD thesis. École doctorale Sciences mathématiques de Paris centre, 2017 (cit. on p. 160).

[44] R. Paterson. "Constructing applicative functors". In: *International Conference on Mathematics of Program Construction*. Germany: Springer, 2012, pp. 300–323 (cit. on p. 173).

[45] V. Poénaru. "What is... an infinite swindle". In: *Notices of the AMS* 54.5 (2007), pp. 619–622 (cit. on p. 162).

[46] J.-F. Pommaret. "Differential Galois Theory". In: *Partial Differential Equations and Group Theory*. Germany: Springer, 1994, pp. 259–318. DOI: 10.1007/978-94-017-2539-2_7 (cit. on pp. 159, 165).

[47] M. Van der Put and M. Singer. *Galois theory of linear differential equations*. Vol. 328. Germany: Springer Science & Business Media, 2012. DOI: 10.1007/978-3-642-55750-7 (cit. on pp. 165, 172).

[48] D. S. Rajan. "The equations $D^k Y = X^n$ in combinatorial species". In: *Discrete Mathematics* 118.1 (1993), pp. 197–206. DOI: 10.1016/0012-365X(93)90061-W (cit. on p. 160).

[49] D. Spivak. Poly*: An abundant categorical setting for mode-dependent dynamics*. 2020. DOI: https://doi.org/10.48550/arXiv.2005.01894 (cit. on p. 160).

[50] D. Spivak and D. Myers. "Dirichlet Functors are Contravariant Polynomial Functors". In: *arXiv:2004.04183 preprint* (2020), p. 11. DOI: https://doi.org/10.48550/arXiv.2004.04183 (cit. on p. 160).

[51] D. Spivak and D. Myers. "Dirichlet Polynomials form a Topos". In: *arXiv:2003.04827 preprint* (2020), p. 11. DOI: https://doi.org/10.48550/arXiv.2003.04827 (cit. on p. 160).
[52] K. Szlachányi. "Monoidal Morita equivalence". In: *Contemporary Mathematics* 391 (2005), pp. 353–370. DOI: 10.1090/conm/391 (cit. on p. 163).
[53] C. T. and H. Z. *Algebraic Groups and Differential Galois Theory*. Graduate Studies in Mathematics. American Mathematical Society, 2011. DOI: 10.1090/gsm/122 (cit. on p. 159).
[54] A. Toumi, R. Yeung, and G. de Felice. "Diagrammatic Differentiation for Quantum Machine Learning". In: *EPTCS* 343 (2021), pp. 132–144. DOI: 10.4204/EPTCS.343.7. eprint: arXiv:2103.07960 (cit. on p. 160).
[55] G. Viennot. "Une interprétation combinatoire des coefficients des développements en série entière des fonctions elliptiques de Jacobi". In: *Journal of Combinatorial Theory, Series A* 29.2 (1980), pp. 121–133. DOI: 10.1016/0097-3165(80)90001-1 (cit. on p. 173).
[56] C. Walker. "Distributive laws via admissibility". In: *Applied Categorical Structures* 27.6 (2019), pp. 567–617. DOI: 10.1007/s10485-019-09567-9 (cit. on p. 163).
[57] B. Yorgey. "Combinatorial Species and Labelled Structures". available at https://repository.upenn.edu/edissertations/1512/. PhD thesis. University of Pennsylvania, 2014 (cit. on p. 166).

A Coherence conditions for strengths

Definition A.1 (Morphism of $\mathcal{R}$-modules)**.** Given a monoidal 2-category, let $\mathcal{R}$ be a pseudomonoid, and let $\mathcal{M}, \mathcal{N}$ be two $\mathcal{R}$-bimodules. We denote the left and right unit constraints by j and k, and left and right associativity constraints by α and β. A (lax) *morphism of $\mathcal{R}$-bimodules* $F : \mathcal{M} \to \mathcal{N}$ is a 1-cell $\mathcal{M} \to \mathcal{N}$, together with, 2-naturally in objects $\mathcal{A}$, maps

$$\begin{array}{c} C \obslash FM \xrightarrow{\xi} F(C \obslash M) \\ FM \oslash C' \xrightarrow{\xi^{R}} F(M \oslash C') \end{array} \tag{21}$$

for every $C, C' : \mathcal{A} \to \mathcal{R}$ and $M : \mathcal{A} \to \mathcal{M}$.

These maps must satisfy the following coherence conditions (we give only the ones pertaining to the left constraints λ, α):

- naturality in both components; the diagrams

$$\begin{array}{ccc} F(C \obslash M) & \xleftarrow{\xi} & C \obslash FM \\ {\scriptstyle F(f \obslash u)}\downarrow & & \downarrow{\scriptstyle f \obslash Fu} \\ F(C' \obslash M') & \xleftarrow[\xi]{} & C' \obslash FM' \end{array} \tag{22}$$

  are commutative, for every pair of morphisms $f : C \to C'$ and $u : M \to M'$.

- compatibility with the monoidality of the action maps, in the form of compatibility with the isomorphisms $C \obslash (C' \obslash M) \cong (C \otimes C') \obslash M$ witnessing the strong monoidality of the action functor and $I \oslash M \cong M$: the diagram

$$\begin{array}{ccc} & F(I \oslash M) & \\ {}^{\xi}\swarrow & & \searrow^{Fj} \\ I \oslash FM & \xrightarrow[j]{} & FM \end{array}$$

$$\begin{array}{ccc} F((C \otimes C') \oslash M) & \xrightarrow{F\alpha} & F(C \oslash (C' \oslash M)) \\ {\scriptstyle \xi}\uparrow & & \uparrow{\scriptstyle \xi} \\ (C \otimes C') \oslash FM & & C \oslash F(C' \oslash M) \\ {\scriptstyle \alpha}\downarrow & & \uparrow{\scriptstyle \xi} \\ C \oslash (C' \oslash FM) & = & C \oslash (C' \oslash FM) \end{array}$$

are commutative, for $C, C' \in \mathcal{R}$, $M \in \mathcal{M}$.

B Proofs

Proof of 4.8. If X is an object of $\mathcal{R}$, then we have a map $\partial(!) : \partial(X) \to \partial(1) = 0$. But for any object A that admits a map $f : A \to 0$, we must have $A \cong 0$, because the composite $\pi_2 \circ (f, 1_A) : A \to 0 \times A \to A$ is the identity of A, and $0 \times A \cong 0$ by distributivity. □

Proof of 4.9. Let $[A, -]$ be the right adjoint of $A \otimes - : \mathcal{R} \to \mathcal{R}$. Then left strengths on T are in natural bijection with enrichment structures on T, i.e. maps $t_{AB} : [A, B] \to [TA, TB]$, and by application of the faithful functor $\mathcal{R}(I, -) : \mathcal{R} \to \mathsf{Set}$, such enrichment structures map one-to-one (not onto necessarily) to Set-enrichment structures $\mathcal{R}(A, B) \to \mathcal{R}(TA, TB)$. However, there is only one of these. □

Proof of 4.13. Let $m : M \otimes M \to M$ be the multiplication of M; the map ∂m is of the following form

$$\partial M \otimes M + M \otimes \partial M \xrightarrow{\partial m} \partial M \tag{23}$$

and by the universal property of coproducts, it can be written as the map $\left[\begin{smallmatrix} i_R \\ i_L \end{smallmatrix}\right]$, where

$$i_R : \partial M \otimes M \to \partial M \qquad i_L : M \otimes \partial M \to \partial M. \tag{24}$$

Evidently, these maps are our candidate right and left actions of M over ∂M.

Now, the fact that m is associative is witnessed by the commutative square

$$\begin{array}{ccc} M \otimes M \otimes M & \xrightarrow{M \otimes m} & M \otimes M \\ {\scriptstyle m \otimes M}\downarrow & & \downarrow{\scriptstyle m} \\ M \otimes M & \xrightarrow[m]{} & M \end{array} \tag{25}$$

If we derive it, applying ∂ to each map, we get the commutative square

$$\begin{array}{ccccc} & & \partial M\otimes M\otimes M+M\otimes\partial M\otimes M+M\otimes M\otimes\partial M & & \\ & \swarrow^{\partial m\otimes M+m\otimes\partial M} & & \searrow^{\partial M\otimes m+M\otimes\partial m} & \\ \partial M\otimes M+M\otimes\partial M & & & & \partial M\otimes M+M\otimes\partial M \\ & \searrow_{\left[\begin{smallmatrix} i_R \\ i_L \end{smallmatrix}\right]} & & \swarrow_{\left[\begin{smallmatrix} i_R \\ i_L \end{smallmatrix}\right]} & \\ & & \partial M & & \end{array}$$

which, thanks to the Leibniz action of ∂ on morphisms, can be seen as the object- and morphism-wise sum of two diagrams

$$\begin{array}{ccc} \cdot & \xrightarrow{\partial M\otimes m} & \cdot \\ {\scriptstyle \partial m\otimes M}\downarrow & & \downarrow{\scriptstyle i_R} \\ \cdot & \xrightarrow[i_R]{} & \cdot \end{array} \qquad\qquad \begin{array}{ccc} \cdot & \xrightarrow{M\otimes\partial m} & \cdot \\ {\scriptstyle m\otimes\partial M}\downarrow & & \downarrow{\scriptstyle i_L} \\ \cdot & \xrightarrow[i_L]{} & \cdot \end{array} \tag{26}$$

witnessing precisely that i_R is a right action, and i_L is a left M-action on ∂M. □

Proof of 5.14. The proof is divided into two parts: first, we show that every object of $\Sigma[Y]$ can be written as a formal sum $\sum E_i \cdot Y^i$, where E_i is a set and Y^i is the ith convolution power of the monoidal unit for Day convolution; then, we show that similarly, every object of $\mathcal{R}[Y]$ can be written as $\sum A_i \otimes Y^i$.

As for the first claim, it follows from the fact that $\Sigma[Y]$ is the closure under coproducts of representables; as for the second claim, we shall show that $\mathcal{R}[Y]$ has the universal property of the coproduct $\mathcal{R}\odot\Sigma[Y]$, or more clearly, the pushout of the span $\Sigma[Y] \leftarrow \mathsf{Fin} \to \mathcal{R}$.[9]

Inspecting the universal property: first of all, there is an obvious cospan of 2-rig morphisms $\mathcal{R} \to \mathcal{R}[Y] \leftarrow \Sigma[Y]$ sending C to $C\otimes Y^0$ and $[n]$ to Y^n; and given a diagram

$$\begin{array}{ccccc} & & \mathcal{R} & & \\ & & \downarrow & \searrow^{G} & \\ \Sigma[Y] & \longrightarrow & \mathcal{R}[Y] & & \\ & \searrow_{F} & & \overset{\left[\begin{smallmatrix} F \\ G \end{smallmatrix}\right]}{\searrow} & \\ & & & & \mathcal{B} \end{array} \tag{28}$$

we can define a unique dotted functor $\left[\begin{smallmatrix} F \\ G \end{smallmatrix}\right] : \mathcal{R}[Y] \to \mathcal{B}$ as

$$\sum_{i=0}^{d} A_i \otimes Y^i \mapsto \sum_{i=0}^{d} GA_i \otimes (FY)^{\otimes n}, \tag{29}$$

since a 2-rig morphism $F : \Sigma[Y] \to \mathcal{B}$ is completely determined by the image of $Y = y(1)$. □

[9] Something analogous happens in commutative algebra, where rings of polynomials with coefficients in R can be defines from free $\mathbf{Z}$-algebras $\mathbf{Z}[X]$ via a universal property, that the diagram

$$\begin{array}{ccc} \mathbf{Z} & \longrightarrow & R \\ \downarrow & \ulcorner & \downarrow \\ \mathbf{Z}[Y] & \longrightarrow & R[Y] \end{array} \tag{27}$$

is a pushout.

Proof of 4.11. Let's examine the left closed case: this means that every $A \otimes -$ has a right adjoint. The right closed case is analogous, mutatis mutandis.

The condition of having a right tensorial strength amounts to the presence of maps $t_{AB} : A \otimes DB \to D(A \otimes B)$ satisfying suitable conditions.

The maps t_{AB} now transpose to

$$\hat{t}_{AB} : DB \longrightarrow [A, D(A \otimes B)] \tag{30}$$

and the $\hat{t}_{AB}$'s are natural in B, and a wedge in A: this means that there is a unique map

$$\hat{t}_B : DB \longrightarrow \int_A [A, D(A \otimes B)]; \tag{31}$$

we now claim that

M1) the correspondence $\lambda B. \int_A [A, D(A \otimes B)]$ is an endofunctor of $\mathcal{A}$;

M2) the correspondence $\Theta : D \mapsto \lambda B. \int_A [A, D(A \otimes B)]$ is an endofunctor of $[\mathcal{A}, \mathcal{A}]$; moreover, it is a comonad;

M3) a Θ-coalgebra is exactly an endofunctor equipped with a right tensorial strength, whose components are obtained from the coalgebra map by reverse-engineering the construction of Θ.

The last part of the third claim is obvious; what remains of the third claim is an exercise on diagram chasing. Functoriality is evident from the canonical way in which we built Θ, and $DB \to \int_A [A, D(A \otimes B)]$ attach to the components of a natural transformation $D \Rightarrow \Theta(D)$.

It remains to show that Θ is a comonad:

- the counit is obtained from the terminal wedge of $\Theta(D)$, taking the component on the monoidal unit (say, I):
$$\int_A [A, D(A \otimes B)] \xrightarrow{\epsilon_B = \pi_I} [I, D(I \otimes B)] \cong DB \tag{32}$$
- the comultiplication is obtained from the following computation:

$$\begin{aligned}
\Theta\Theta(D)(A) &= \int_B [B, \Theta(D)(A \otimes B)] \\
&\cong \int_B [B, \int_C [C, D(A \otimes B \otimes C)]] \\
&\cong \int_B \int_C [B, [C, D(A \otimes B \otimes C)]] \\
&\cong \int_B \int_C [B \otimes C, D(A \otimes B \otimes C)]
\end{aligned}$$

It is evident, now, that the projections $\pi_{B \otimes C}$ of the terminal wedge of $\Theta(D)$ assemble into a morphism $\sigma : \Theta \Rightarrow \Theta\Theta$ of the right type; moreover, this choice of ϵ and σ is the unique that satisfies the counit equations of a comonad; showing that $\sigma : \Theta \Rightarrow \Theta^2$ is coassociative is a matter of diagram chasing. □

Proof of 5.9. Let $\boldsymbol{D}[Y] \to \mathcal{R} \ltimes \mathcal{R}$ be the essentially unique $\boldsymbol{D}$-rig map that takes Y to $(0, I)$, and let $\mathcal{R} \to \mathcal{R} \ltimes \mathcal{R}$ be the map taking C to $(C, 0)$. By pairing these maps, we get a map $\varepsilon : \mathcal{R}[Y] \to \mathcal{R} \ltimes \mathcal{R}$ out of the coproduct $\mathcal{R}[Y] = \mathcal{R} \odot \boldsymbol{D}[Y]$. It is clear that ε coinverts the 2-cell $0 \Rightarrow Y^2$. Given a $\boldsymbol{D}$-rig map $F : \mathcal{R}[Y] \to \mathcal{S}$ that coinverts this 2-cell, define a map $\overline{F} : \mathcal{R} \ltimes \mathcal{R} \to \mathcal{S}$ that takes $(R, 0)$ to $F(R)$, and $(0, I)$ to $F(Y)$. One may check that $\overline{F}$ is a $\boldsymbol{D}$-rig map. □

Proof of 5.18. Let ∂ denote the standard derivative, and denote the $\boldsymbol{D}$-rig map $-\circ G$ by φ. Then the left side corresponds to the value at an object F of the composite $\boldsymbol{D}$-rig map

$$\boldsymbol{D}[1] \xrightarrow{\varphi} \boldsymbol{D}[1] \xrightarrow{\langle 1,\partial\rangle} \boldsymbol{D}[1][\varepsilon], \tag{33}$$

taking F to $(F\circ G, (F\circ G)')$ and taking X to (G, G'). On the other hand, $\varphi\circ\partial : \boldsymbol{D}[1]\to\boldsymbol{D}[1]$ is a φ-augmented derivation, and so is $(\varphi\partial)\otimes G'$. By 5.2, it corresponds to the $\boldsymbol{D}$-rig map $\boldsymbol{D}[1]\to\boldsymbol{D}[1][\varepsilon]$ taking F to

$$(\varphi(F), (\varphi\partial(F))\otimes G') = (F\circ G, (F'\circ G)\otimes G'). \tag{34}$$

This map is uniquely determined by where it sends the generator X, but this value on X is the same as before,

$$(X\circ G, (X'\circ G)\otimes G') = (G, G'). \tag{35}$$

This means the $\boldsymbol{D}$-rig maps

$$F\mapsto (F\circ G, (F\circ G)'), \qquad F\mapsto (F\circ G, (F'\circ G)\otimes G') \tag{36}$$

coincide, and this completes the proof. □

Generalized Leibniz rule and Taylor formula

Proof of 4.4. Expand $(F*G)^{(U)}[C] = (F*G)[C+U]$ using the fact that

$$(F*G)[C+U] = \sum_{A+B=C+U} FA\times GB. \tag{37}$$

For each indexing pair (A,B), put $A' = A\cap C$, $B' = B\cap C$, $S = A\cap U$, $T = B\cap U$. Then $A = A'+S$ and $B = B'+T$ and $S+T = U$. It follows that

$$\begin{aligned}(F*G)[C+U] &\cong \sum_{A+B=C+U} FA\times GB\\ &\cong \sum_{S+T=U}\sum_{A'+B'=C} F[A'+S]\times G[B'+T]\\ &\cong \sum_{S+T=U}\sum_{A'+B'=C} F^{(S)}[A']\times G^{(T)}[B']\\ &\cong \sum_{S+T=U} (F^{(S)}*G^{(T)})[C]\end{aligned}$$

This concludes the proof. □

Proof of 4.5. Let's first observe that we have the analytic functor formula

$$F(X) = \int^n F[n]\times X^n \tag{38}$$

which mimics the Maclaurin series expansion; this is obtained from the fact that $F(-)\cong \mathrm{Lan}_J F$, and the integral on the right-hand side is exactly that Kan extension.

Now given an n-element set U, let's interpret $\partial^n F(A) = \partial^{(U)} F(A) = F(U+A)$ as a species in the variable n but as analytic in the set-variable A. We have then the formula

$$\partial^n F(A) = \int^m F[m+n] \times A^m. \tag{39}$$

And thus we can categorify $\sum_{n=0}^{\infty} \frac{\partial^n f(a)}{n!} x^n$ as the double coend

$$\begin{aligned}
&\int^{nm} F[m+n] \times A^m \times X^n \\
&\qquad \cong \int^{nmj} F[j] \times \Sigma(j, m+n) \times A^m \times X^n \\
&\qquad\qquad \cong \int^j F[j] \times \left(\int^{m,n} \Sigma(j, m+n) \times A^m \times X^n \right).
\end{aligned}$$

Now, we have an isomorphism

$$\int^{mn} \Sigma(j, m+n) \times A^m \times X^n \cong (A+X)^j \tag{40}$$

which ultimately comes out of the fact that Set is an extensive category: there exists an equivalence of categories $\mathsf{Set}/A \times \mathsf{Set}/X \cong \mathsf{Set}/(A+X)$. We conclude that

$$\int^j F[j] \times (A+X)^j \tag{41}$$

is the value $F(A+X)$ of the analytic functor $F(-)$. □

Proof that (15) *is a coinverter.* The universal property of the coinverter amounts to the following:

- C1) for each morphism of 2-rigs $p : \mathcal{C}[Y] \to \mathcal{X}$ such that $0 \to p(Y^2 \otimes R(Y))$ is invertible in $\mathcal{X}$, there exists a unique (up to isomorphism) $\bar{p} : \mathcal{C}[Y]_{<2} \to \mathcal{X}$ such that $q \circ \bar{p} = p$;
- C2) for each natural transformation $\alpha : p \Rightarrow p'$ of 2-rig morphisms with the property that the horizontal composition $\alpha \boxminus u$ is an isomorphism, there exists a unique $\bar{\alpha} : \bar{q} \Rightarrow \bar{q}'$ such that $q * \bar{\alpha} = \alpha$.

Both properties descend from the fact that p, being a 2-rig morphism, preserves coproducts; if $p(A + BY + RY^2) \cong p(A+BY) + p(RY^2)$, and the initial arrow $0 \to p(RY^2)$ is an isomorphism, the vertical right arrow in the commutative diagram

$$\begin{array}{ccc}
p(A+BY)+0 & \longrightarrow & p(A+BY) \\
\downarrow & & \downarrow \\
p(A+BY)+p(RY^2) & \longrightarrow & p(A+BY+RY^2)
\end{array} \tag{42}$$

is an isomorphism; thus, p is uniquely determined by its action on $\mathcal{C}[Y]_{<2}$, and $\bar{p}(A+BY)$ can be defined just as $p(A+BY)$. For what concerns 2-cells $\alpha : p \Rightarrow p'$, a similar diagram

$$\begin{array}{ccc}
p(A+BY+RY^2) & \longrightarrow & p'(A+BY+RY^2) \\
\wr\downarrow & & \downarrow\wr \\
p(A+BY) & \xrightarrow[\alpha_{A+BY}]{} & p'(A+BY) \\
\Vert & & \Vert \\
p(A+BY) & \xrightarrow[\bar{\alpha}_{A+BY}]{} & p'(A+BY)
\end{array} \tag{43}$$

is commutative, so α is uniquely determined by its components at objects $A + BY$ of $C[Y]_{<2}$. □

Dynamic Operads, Dynamic Categories: From Deep Learning to Prediction Markets

Brandon T. Shapiro David I. Spivak

Natural organized systems adapt to internal and external pressures and this happens at all levels of the abstraction hierarchy. Wanting to think clearly about this idea motivates our paper, and so the idea is elaborated extensively in the introduction, which should be broadly accessible to a philosophically-interested audience.

In the remaining sections, we turn to more compressed category theory. We define the monoidal double category $\mathbb{O}\mathbf{rg}$ of dynamic organizations, we provide definitions of $\mathbb{O}\mathbf{rg}$-enriched, or *dynamic*, categorical structures—e.g. dynamic categories, operads, and monoidal categories—and we show how they instantiate the motivating philosophical ideas. We give two examples of dynamic categorical structures: prediction markets as a dynamic operad and deep learning as a dynamic monoidal category.

1 Introduction

Intuitively, an *open dynamical system* is a machine or worker with an interface by which to interact with whatever else is out there. Open dynamical systems can be organized as circuits or control loops, so that they affect each other by their outward expressions of internal work, and thereby possibly form a more complex worker. The framework here is fractal—or more precisely *operadic*—in its structure: organizations of workers can be nested into arbitrary hierarchies of abstraction.

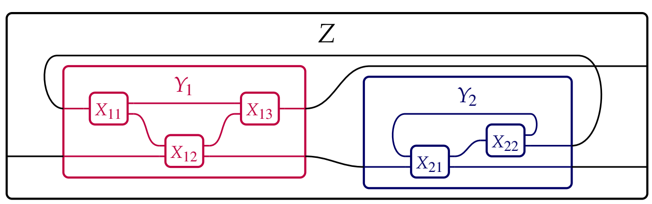


Figure 1: A nesting of interacting open dynamical systems: the $X_{i,j}$ are wired together to form the Y_i, which are wired together to form Z; typically these groupings are chosen to create new abstractions, e.g. in logical circuits or control systems. The permanence of the above-displayed wiring pattern is exactly what is relaxed in this paper; a dynamic organization is one in which interactions may change dynamically based on what flows within the system.

But if we think about some things that interact to do work in the real world, we notice that often the organization itself—the connections themselves—change. Unlike what we see in Fig. 1, the way we connect this hour may be different from the way we connect next hour; in particular, our interfaces go in

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 183–202, doi:10.4204/EPTCS.380.11

and out of contact. At the end of this paragraph, look away from the page for a few seconds, think about some things you know that interact together or influence each other, and ask yourself three questions about them: Do these things ever stop interacting? If so, do they ever start interacting again? And how is it decided?

1.1 Accounting for organizational change

We propose that the metaphysical nature and scope of these questions should be complemented by some sort of guard rails to keep our contemplation on track. This is the role of mathematics in our work. It provides a symbolic *accounting system* which is articulate enough to facilitate one person in explicating an example and asking questions about it.

The category **Poly** of polynomial functors in one variable is an ergonomic mathematical structure with many applications and spin-off categorical gadgets. We will begin in Section 2 by recalling one such gadget from [4]: a category-enriched multicategory $\mathbb{O}\mathbf{rg}$ that will be the conceptual centerpiece of our accounting system. Its objects are polynomial functors in one variable, and its morphisms are polynomial coalgebras related to a certain monoidal closed structure on **Poly**. We will see that the morphisms in $\mathbb{O}\mathbf{rg}$ are intuitively "collective organizational patterns that change dynamically".

Leaving the mathematics aside until Section 2—at which point we will have almost nothing more to say about the background philosophy—let's return to the question "how is the organizational pattern between various systems decided, moment-by-moment?" Let's mesh this question with the idea that the so-organized systems can be nested into arbitrary hierarchies of abstraction. And let's think about all this in the frame of a certain worldview which we invite you the reader to engage with like a fictional movie, not intended to convince you of fact but instead simply to convey an experience. Here goes.

In this worldview, we notice that everything that makes any sense to us happens to be a collective. A cell body, a human body, an antibody, Topos Institute, an idea, an airport, a sentence, a mathematical definition, a grain of sand, ... each is a collective of interacting parts that may themselves be collectives.

It's quite often the case that these collectives, like the ship of Theseus, are not permanent organizations that are fixed for all time; they are adapting to forces from within and without the system. Even a grain of sand can break or melt; even a mathematical definition can be refactored. So then what's outside the system, generating these forces that influence it? We imagine that what's outside is in fact more of the same kind of stuff as what's inside, just not as cohesive perhaps. Let's go full-on woo: if the universe is a big system, then maybe the sort of thing that happens in our head is—in some way—just like what happens outside of it. Maybe the motives that organize Brandon and David into a collaborative thinking and paper-writing unit are, in the some reasonable account, of the same nature as the motives that organize each one of them into a body.

But is this right? How could you check such a claim? One would need to give a reasonable account of it, and since we as authors can't currently give such an account, we don't make this claim. Instead, what we present here is an *accounting system* in which the woo-person, (or would it instead be the reductive materialist?) who thought that what went on inside the head was somehow the same as what went on outside, could endeavor to provide such an account of their thinking.

1.2 Dynamic categorical structures

Our main definition in this paper is what we call an *dynamic categorical structure*. We might poetically say that a *dynamic* category is one where the morphisms between two objects change in response to what flows between those objects. To define it, we first refactor the definition of $\mathbb{O}$**rg** from [4] from an operad to a monoidal double category; we then define a dynamic *thing* to be a *thing* enriched in $\mathbb{O}$**rg**. Once these are defined, we give a couple examples: a *prediction market* operad and a *deep learning* monoidal category. In the prediction market, a population Y predicts a distribution based on the predictions of its member populations X_i weighted by their reputations, and the reputations change dynamically based on the returned outcome. A similar story holds with deep learning.

We thank you the reader for having postponed your counterpoints and counterexamples, and we ask you to reengage both skepticism and interest as you see fit. We invite you to ask openly: what's not a collective of interacting parts that are themselves collectives? Nature, love, or experience perhaps? It all depends on how you look. What we present here is an accounting system for making sense of a certain sort of experiential pattern; the matter itself is whatever it is.

1.3 Acknowledgments

The influences on this paper are too numerous and unranked to name, but in particular we thank Sophie Libkind for stimulating conversations, and we thank Scott Garrabrant for teaching us about Kelly betting, which partially inspired Section 4.1. Thanks also to David Jaz Myers and Samantha Jarvis for catching typos.

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0348.

2 The Monoidal Double Category $\mathbb{O}$rg

In [4], the second author defined a category-enriched multicategory $\mathbb{O}$**rg**, whose objects are polynomials and whose morphisms are polynomial coalgebras. In this section, we describe how $\mathbb{O}$**rg** in fact more naturally takes the form of a monoidal double category, with coalgebras as horizontal morphisms, maps of polynomials as vertical morphisms, and the Dirichlet tensor product $\otimes$ (see (2) below) providing the monoidal structure.[1]

Before we begin, recall that a polynomial is a functor $p\colon \mathbf{Set} \to \mathbf{Set}$ which is isomorphic to a sum of representables; following [4], we denote $p, q \in \mathbf{Poly}$ by

$$p = \sum_{I \in p(1)} y^{p[I]} \qquad \text{and} \qquad q = \sum_{J \in q(1)} y^{q[J]} \tag{1}$$

and refer to each $I \in p(1)$ as a p*-position* and to each $i \in p[I]$ as a p*-direction at* I. A map $\phi\colon p \to q$ of polynomials is a natural transformation. Combinatorially, ϕ provides: for each $I \in p(1)$ a choice of $\phi(I) \in q(1)$ and for each $j \in q[\phi(I)]$ a choice of $\phi(I, j) \in p[I]$.[2]

[1] In fact, $\mathbb{O}$**rg** is a duoidal double category, with a second monoidal structure $\triangleleft$, but we will not use that here.

[2] In [4], what we denote $\phi(I)$ is denoted $\phi_1(I)$ and what we denote $\phi(I, j)$ is denoted $\phi_I^\sharp(j)$.

For polynomials p, q, their Dirichlet tensor product is the polynomial

$$p \otimes q = \sum_{(I,J) \in p(1) \times q(1)} y^{p[I] \times q[J]} \tag{2}$$

2.1 $[p, q]$-coalgebras

We first recall the definitions of the internal-hom polynomials $[p, q]$ and concretely describe the category of $[p, q]$-coalgebras, which will form the category of morphisms from p to q in the underlying bicategory of $\mathbb{O}$**rg**.

Definition 2.1. For polynomials $p, q \in \mathbf{Poly}$ as in (1), their *internal hom* with respect to the tensor product $\otimes$ is the polynomial

$$[p, q] \coloneqq \sum_{\phi\colon p \to q} y^{\sum\limits_{I \in p(1)} q[\phi(I)]} \tag{3}$$

It can also be written $[p, q] \cong \prod_{I \in p(1)} \sum_{J \in q(1)} \prod_{j \in q[J]} \sum_{i \in p[I]} y$. ◊

For intuition, a $[p, q]$-coalgebra (denoted $p \twoheadrightarrow q$) is a machine that outputs maps $\phi\colon p \to q$ and that inputs what *flows* between them: pairs (I, j) where $I \in p(1)$ is a position of p, which "flows" to q as $J \coloneqq \phi(I) \in q(1)$, and $j \in q[J]$ is a direction of q, which "flows" backward to p as $\phi(I, j) \in p[I]$. More precisely, using [4, Definition 2.10], we define $[p, q]$-coalgebras as follows.

Definition 2.2. The category $[p, q]$-**Coalg** has as objects pairs $\mathbf{S} = (S, \beta)$ where S is a set and $\beta\colon S \to [p, q](S)$ is a function, and where a morphism from $\mathbf{S}$ to $\mathbf{S}'$ is a function $f\colon S \to S'$ making (4) commute.

$$\begin{array}{ccc} S & \xrightarrow{\beta} & [p,q](S) \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle [p,q](f)} \\ S' & \xrightarrow[\beta']{} & [p,q](S') \end{array} \tag{4}$$

We refer to S as the *state set* and to each element $s \in S$ as a *state*. ◊

Unwinding this definition, it is useful to break β into two functions $\beta \coloneqq (\mathrm{act}^\beta, \mathrm{upd}^\beta)$, an *action* function

$$\mathrm{act}^\beta\colon S \to \mathbf{Poly}(p, q) = [p, q](1)$$

and, for each state $s \in S$, an *update* function

$$\mathrm{upd}^\beta_s\colon \sum_{I \in p(1)} q\left[\mathrm{act}^\beta_s(I)\right] \to S.$$

For a state $s \in S$ and position $I \in p(1)$ we often write $\mathrm{act}^\beta_s\colon p \to q$ and $\mathrm{upd}^\beta_s(I)\colon q[\mathrm{act}^\beta_s(I)] \to S$. We may suppress the β when it is clear from context, writing act_s and upd_s. A coalgebra map $\mathbf{S} \to \mathbf{S}'$ is a function $S \to S'$ between the state sets that preserves actions and updates.

When, for each $s \in S$, the update upd_s is the constant function sending everything to s, we say the coalgebra $\mathbf{S}$ is *static*, as it remains constantly at s regardless of the inputs $I \in p(1)$ and $j \in q[\mathrm{act}_s(I)]$ flowing between p and q.

Example 2.3. A special case of a static $[p,q]$-coalgebra is given by a map $\phi \in \mathbf{Poly}(p,q)$. For each such ϕ, there is a coalgebra $\{\phi\}$ with a singleton state set and with act^β sending the point to ϕ; we call it a *singleton* coalgebra.

A coalgebra is static iff it is the coproduct of singleton coalgebras. ◊

More examples and intuition for $[p,q]$-coalgebras can be found in [4].

2.2 Composition of hom-coalgebras

We now describe how $[p,q]$-coalgebras behave like morphisms from p to q.

Proposition 2.4. *The categories $[p,q]$-**Coalg** form the hom-categories in a bicategory $\mathbb{O}$**rg**, which has polynomials as objects.*

We use $\mathbb{O}$**rg** to denote both the bicategory from Proposition 2.4 and the categorical operad in [4, Definition 2.19], as both are derived from the monoidal double category $\mathbb{O}$**rg** described in the following sections. For now, we merely present the identities and composites in this bicategory. Identities are easy: the identity object in $\mathbb{O}\mathbf{rg}(p,p) = [p,p]$-**Coalg** is given by the one-state coalgebra $\{\text{id}_p\}$.

The composition functor $\mathbb{O}\mathbf{rg}(p,q) \times \mathbb{O}\mathbf{rg}(q,r) \longrightarrow \mathbb{O}\mathbf{rg}(p,r)$ is defined as the composite:

$$[p,q]\text{-}\mathbf{Coalg} \times [q,r]\text{-}\mathbf{Coalg} \to ([p,q] \otimes [q,r])\text{-}\mathbf{Coalg} \longrightarrow [p,r]\text{-}\mathbf{Coalg},$$

where the first functor is the lax monoidality of $(-)$-**Coalg**: **Poly** → **Cat**, as described in [4, Proposition 2.13], and the second is given by applying $(-)$-**Coalg** to the usual "composition" map of internal-homs $[p,q] \otimes [q,r] \to [p,r]$ in **Poly**. Using (3) we see that on positions, this map takes the form

$$([p,q] \otimes [q,r])(1) = \mathbf{Poly}(p,q) \times \mathbf{Poly}(q,r) \xrightarrow{\fatsemi} \mathbf{Poly}(p,r) = [p,r](1)$$

and on directions it is given for $\phi\colon p \to q$ and $\psi\colon q \to r$ by the function

$$\left(\sum_{I \in p(1)} q[\phi(I)]\right) \times \left(\sum_{J \in q(1)} r[\psi(J)]\right) \leftarrow \sum_{I \in p(1)} r[\psi(\phi(I))]$$

which sends (I,k) to $\big((I, \psi(\phi(I),k)), (\phi(I), j)\big)$.

Concretely, the composite of a $[p,q]$-coalgebra $\mathbf{S}$ and a $[q,r]$-coalgebra $\mathbf{S}'$ is a $[p,r]$-coalgebra which we denote $\mathbf{S} \fatsemi \mathbf{S}'$ and define as follows:

- its state set is given by $S \times S'$
- the action of the pair (s,s') is given by the composite

$$\text{act}^{\beta \fatsemi \beta'}_{s,s'} := (\text{act}^\beta_s \fatsemi \text{act}^{\beta'}_{s'})\colon p \to q \to r$$

- the update function of (s,s') is induced by the functions

$$\sum_{I \in p(1)} r\left[\text{act}^{\beta\fatsemi\beta'}_{s,s'}(I)\right] \xrightarrow{(I,k) \mapsto \left(I, \text{act}^{\beta'}_{s'}\left(\text{act}^\beta_s(I), k\right)\right)} \sum_{I \in p(1)} q\left[\text{act}^\beta_s(I)\right] \xrightarrow{\text{upd}^\beta_s} S,$$

$$\sum_{I \in p(1)} r\left[\text{act}^{\beta\fatsemi\beta'}_{s,s'}(I)\right] \xrightarrow{(I,k) \mapsto \left(\text{act}^\beta_s(I), k\right)} \sum_{J \in q(1)} r\left[\text{act}^{\beta'}_{s'}(J)\right] \xrightarrow{\text{upd}^{\beta'}_{s'}} S'.$$

Horizontal composition of coalgebra-morphisms—i.e. of the 2-cells in the bicategory $\mathbb{O}$**rg**—is given simply by the cartesian product. The coherence isomorphisms and axioms for a bicategory then follow from the essential uniqueness of finite products of sets, and the unitality and associativity of composition for polynomial maps.

2.3 Monoidal product of coalgebras

It is shown in [4, Proposition 2.13] that the tensor product $\otimes$ of polynomials extends to make $\mathbb{O}$**rg** a monoidal bicategory. That is, for polynomials p,q,p',q' there is a functor

$$[p,q]\text{-}\mathbf{Coalg}\times[p',q']\text{-}\mathbf{Coalg}\to([p,q]\otimes[p',q'])\text{-}\mathbf{Coalg}\to[p\otimes p',q\otimes q']\text{-}\mathbf{Coalg}$$

derived from the map of polynomials $[p,q]\otimes[p',q']\to[p\otimes p',q\otimes q']$ given on positions by

$$\mathbf{Poly}(p,q)\times\mathbf{Poly}(p',q')\xrightarrow{\otimes}\mathbf{Poly}(p\otimes p',q\otimes q')$$

and on directions by, for $\phi\colon p\to q$ and $\phi'\colon p'\to q'$,

$$\left(\sum_{I\in p(1)}q[\phi_1(I)]\right)\times\left(\sum_{I'\in p'(1)}q'[\phi'_1(I')]\right)\longleftarrow\sum_{(I,I')\in p(1)\times p'(1)}q[\phi_1(I)]\times q'[\phi'_1(I')]$$

sending (I,I',j,j') to (I,j,I',j').

Concretely, this tensor product takes a $[p,q]$-coalgebra $\mathbf{S}$ and a $[p',q']$-coalgebra $\mathbf{S}'$ to the $[p\otimes p',q\otimes q']$-coalgebra with states $S\times S'$, action

$$S\times S'\to\mathbf{Poly}(p,q)\times\mathbf{Poly}(p',q')\to\mathbf{Poly}(p\otimes p',q\otimes q'),$$

and update described similarly componentwise. The tensor product of coalgebra morphisms is also given by the cartesian product of functions, and it is (very) tedious but ultimately straightforward to check that the essential uniqueness of products guarantees that $\otimes$ gives a monoidal structure on $\mathbb{O}$**rg**.

2.4 $\mathbb{O}$rg as a double category

Defining $\mathbb{O}$**rg** as a monoidal bicategory is sufficient for most of the constructions of $\mathbb{O}$**rg**-enriched structures in Section 3. However, using a double category structure casting singleton coalgebras $\mathbf{S}_\phi\in[p,q]\text{-}\mathbf{Coalg}$ (see Example 2.3) as morphisms $\phi\colon p\to q$ in **Poly** facilitates our eventual definition of maps between dynamic structures.

Specifically, the definition of $\mathbb{O}$**rg** as a monoidal bicategory extends to a monoidal pseudo-double category with coalgebras as horizontal morphisms, maps in **Poly** as vertical morphisms, and squares as in (5) given by maps of coalgebras from $\mathbf{S}\fatsemi\{\psi\}$ to $\{\phi\}\fatsemi\mathbf{S}'$.

$$\begin{array}{ccc} p & \xrightarrow{\mathbf{S}} & q \\ {\scriptstyle\phi}\big\downarrow & \Downarrow & \big\downarrow{\scriptstyle\psi} \\ p' & \xrightarrow[\mathbf{S}']{} & q' \end{array} \tag{5}$$

The symbol $\nrightarrow$ is intended to indicate that the map is "dynamic", changing in response to what flows between p and q.

As $\{\phi\}$ and $\{\psi\}$ have only one state, and composition of coalgebras acts as the cartesian product on states, such a square amounts to a function $S \to S'$ making (6) commute:

$$\begin{array}{ccccc} S & \xrightarrow{\beta} & [p,q](S) & \xrightarrow{\psi_*} & [p,q'](S) \\ {\scriptstyle f}\big\downarrow & & & & \big\downarrow{\scriptstyle [p,q'](f)} \\ S' & \xrightarrow[\beta']{} & [p',q'](S') & \xrightarrow[\phi^*]{} & [p,q'](S') \end{array} \tag{6}$$

Identities and composites for these squares are determined by the bicategory structure, as this double category is a restriction in the vertical direction of the double category of lax-commuting squares in a bicategory.[3]

We now proceed to discuss various categorical structures enriched in $\mathbb{O}$**rg**, which describe dynamical systems equipped with algebraic structure that lets us remove abstraction barriers when considering nested layers and complex arrangements of the components of the system.

3 Dynamic structures via $\mathbb{O}$rg-Enrichment

A monoidal double category is a viable setting for enriching various categorical structures. Intuitively, enrichment in $\mathbb{O}$**rg** replaces the usual set of arrows between two objects in a categorical structure with a $[p,q]$-coalgebra for some choice of polynomials p,q. Therefore not only can each arrow be realized as a map of polynomials $p \to q$, but this map carries dynamics that encode how a position in p and a direction in q determine a transition from one arrow to another. The morphism "reacts" to what's flowing between p and q.

Different situations call for different categorical structures to model their dynamics: some systems primarily involve many-to-one arrangements such as the wiring diagrams in Fig. 1, others such as gradient descent fit naturally into a many-to-many arrow framework, and we expect in future work to consider evolving systems in which different components operate at differing time scales. Rather than choose one such categorical form to favor, and then go through the tedious exercise of forcing all of the others to conform to it, we describe how to add dynamics to the definitions of many different structures.

*A dynamic *thing* is a *thing* enriched in $\mathbb{O}$**rg**.*

This slogan is intentionally imprecise, so as to be maximally inclusive of different notions of categorical structures (*things*) and notions of enrichment, and also to allow the reader who has an intuitive understanding and no need for precision to skip the remainder of this paragraph. Our intuition and examples come from the theories of enrichment described in [1] and [3]. In the former, a *thing* can be any suitable type of generalized multicategory, while in the latter a *thing* can be any structure defined as an algebra for a familial monad on a presheaf category equipped with a choice of "higher" and "lower"

[3] It should be noted however that the vertical arrows in $\mathbb{O}$**rg** are regarded as polynomial maps rather than coalgebras, so that they compose strictly unitally and associatively.

dimensional cell shapes. In both cases, *things* are algebras for a particular cartesian monad T and admit an "enriched" analogue with respect to any T-multicategory. To define T-algebras enriched in $\mathbb{O}\mathbf{rg}$ is then to identify $\mathbb{O}\mathbf{rg}$ with a T-multicategory, and in all of our examples this identification arises naturally from the observation that monoidal double categories give rise to T-multicategories in a natural way.

We now give specific instances of $\mathbb{O}\mathbf{rg}$-enrichment: in Section 3.1 for dynamical categories, in Section 3.2 for dynamical multicategories and operads, and in Section 3.3 for dynamical monoidal categories and PRO(P)s. We are also interested in using dynamic duoidal categories to describe dynamical systems in which different contributors to a system operate at different rates, using the duoidal structure on $\mathbb{O}\mathbf{rg}$ based on $\triangleleft$, but that is beyond the scope of this paper.

3.1 Dynamic categories

Enrichment of categories only uses $\mathbb{O}\mathbf{rg}$'s double category structure—not its monoidal structure—as any double category forms an fc-multicategory (also known as a virtual double category) in the sense of [1, Section 2.1]. The following definition of enrichment in $\mathbb{O}\mathbf{rg}$ is an unwound version of the more general definition in [1, Section 2.2].

Definition 3.1. An $\mathbb{O}\mathbf{rg}$-enriched (henceforth *dynamic*) category A consists of

- a set A_0 of objects;
- for each $a \in A_0$, a polynomial p_a;
- for each $a, b \in A_0$, a $[p_a, p_b]$-coalgebra $\mathbf{S}_{a,b}$;
- for each $a \in A_0$, an "identitor" square in $\mathbb{O}\mathbf{rg}$ as in (7) left; and
- for each $a, b, c \in A_0$, a "compositor" square in $\mathbb{O}\mathbf{rg}$ as in (7) right:

$$\begin{array}{ccc} p_a & \xrightarrow{\{\mathrm{id}_{p_a}\}} & p_a \\ \| & \Downarrow & \| \\ p_a & \xrightarrow[\mathbf{S}_{a,a}]{} & p_a \end{array} \qquad\qquad \begin{array}{ccccc} p_a & \xrightarrow{\mathbf{S}_{a,b}} & p_b & \xrightarrow{\mathbf{S}_{b,c}} & p_c \\ \| & & \Downarrow & & \| \\ p_a & & \xrightarrow[\mathbf{S}_{a,c}]{\qquad\qquad} & & p_c \end{array} \tag{7}$$

such that these squares satisfy unit and associativity equations (Definition A.1). ◊

The sets $S_{a,b}$ form an ordinary category which we say *underlies* A. In fact, a dynamic category could be equivalently defined as an ordinary category such that each object a is assigned a polynomial p_a and each set of arrows $\mathrm{Hom}(a,b)$ is equipped with a $[p_a, p_b]$-coalgebra structure, with composition and identities respecting the coalgebra structure. This means that the arrow id_a in $\mathrm{Hom}(a,a)$ acts as the identity map on p_a and is unchanged by updates, while for f in $\mathrm{Hom}(a,b)$ and g in $\mathrm{Hom}(b,c)$ the composite $f \mathbin{⨟} g$ acts as the composite $p_a \to p_b \to p_c$ of the actions of f and g, and the update of their composite equals the composite of their updates.

3.2 Dynamic multicategories

A monoidal double category also gives rise to an fm-multicategory in the sense of [1, Section 3.1], so we can talk about multicategories enriched in $\mathbb{O}\mathbf{rg}$ as in [1, Section 3.2].

Definition 3.2. An $\mathbb{O}\mathbf{rg}$-enriched (henceforth *dynamic*) multicategory A consists of

- a set A_0 of objects;
- for each $a \in A_0$, a polynomial p_a;
- for each $a_1,...,a_n,b \in A_0$, a $[p_{a_1} \otimes \cdots \otimes p_{a_n}, p_b]$-coalgebra $\mathbf{S}_{a_1,...,a_n;b}$;
- for each $a \in A_0$, an "identitor" square in $\mathbb{O}\mathbf{rg}$ as in (8) left; and
- for each $a_{1,1},\ldots,a_{1,m_1}, \ldots, a_{n,1},\ldots,a_{n,m_n}, b_1,\ldots,b_n$, and $c \in A_0$, a "compositor" square in $\mathbb{O}\mathbf{rg}$ as in (8) right

$$\begin{array}{ccc} p_a & \xrightarrow{\{\mathrm{id}_{p_a}\}} & p_a \\ \| & \Downarrow & \| \\ p_a & \xrightarrow[\mathbf{S}_{a;a}]{} & p_a \end{array} \qquad \begin{array}{ccccc} p_{a_{1,1}} \otimes \cdots \otimes p_{a_{n,m_n}} & \xrightarrow{\bigotimes_i \mathbf{S}_{a_{i,1},...,a_{i,m_i};b_i}} & p_{b_1} \otimes \cdots \otimes p_{b_n} & \xrightarrow{\mathbf{S}_{b_1,...,b_n;c}} & p_c \\ \| & & \Downarrow & & \| \\ p_{a_{1,1}} \otimes \cdots \otimes p_{a_{n,m_n}} & & \xrightarrow[\mathbf{S}_{a_{1,1},...,a_{n,m_n};c}]{} & & p_c \end{array} \tag{8}$$

satisfying unit and associativity equations (see Definition A.2 for the one-object case). ◊

The sets $S_{a,b}$ form an ordinary (set-enriched) multicategory, which underlies A and has a description similar to the underlying category we described below Definition 3.1.

We will mostly be interested in what we call a *dynamic operad*, the case when a dynamic multicategory A has only one object, assigned the polynomial "interface" p. It consists simply of a $[p^{\otimes n}, p]$-coalgebra $\mathbf{S}_n$ for each $n \in \mathbb{N}$, equipped with coalgebra maps

$$\{\mathrm{id}_p\} \to \mathbf{S}_1 \qquad \text{and} \qquad \bigotimes_{i \in I} \mathbf{S}_{n_i} \to \mathbf{S}_N \tag{9}$$

where I is any finite set and $N := \sum_{i \in I} n_i$, which together satisfy the usual equations.

Example 3.3. A *collective* (as defined in [2]) is a $\otimes$-monoid in **Poly**, meaning a polynomial p equipped with a monoid structure on its positions $p(1)$ and co-unital co-associative "distribution" functions $p[I \cdot J] \to p[I] \times p[J]$ for each $I, J \in p(1)$. This can be viewed as a dynamic operad where $\mathbf{S}_n$ is given by $\{\cdot_n\}$, the singleton coalgebra on the n-ary monoidal product $(\cdot_n)\colon p^{\otimes n} \to p$, and where the maps of coalgebras in (9) are isomorphisms deduced from the equations for a monoid. ◊

Example 3.4. In Example 3.3, the coalgebras $\mathbf{S}_n$ are determined by a single map of polynomials, with trivial updates since the state sets are singletons. This can be generalized to an intermediate notion between collectives and dynamic multicategories, where the coalgebras are still static but may have multiple states.

Given any multicategory M and multifunctor $F\colon M \to \mathbf{Poly}$, where **Poly** here denotes the multicategory underlying $(\mathbf{Poly}, y, \otimes)$, there is a dynamic multicategory A_F with

- object set $\mathrm{Ob}(M)$;
- for each $a \in \mathrm{Ob}(M)$, the polynomial interface $p_a := F(a)$;
- for each tuple $(a_1,...,a_n;b)$ in $\mathrm{Ob}(M)$, state set $S_{a_1,...,a_n;b} := M(a_1,...,a_n;b)$;
- the action $\mathrm{act}^\beta\colon M(a_1,...,a_n;b) \to \mathbf{Poly}(p_{a_1} \otimes \cdots \otimes p_{a_n}, p_b)$ is given by F; and
- for any state s in $M(a_1,...,a_n;b)$, the update function upd_s^β is the constant function at s. ◊

Example 3.5. Let $\mathbf{S}$ be any p-coalgebra for a polynomial p. There is a dynamic operad on p with $\mathbf{S}_0 := \mathbf{S}$, with $\mathbf{S}_1 := \{\mathrm{id}_p\}$, and with all other $\mathbf{S}_n := \varnothing$ assigned the empty coalgebra. ◊

Example 3.6. Consider a dynamic operad with interface $y \in \mathbf{Poly}$. The internal hom polynomial $[y^{\otimes n}, y]$ is simply y, so this structure amounts to an operad S with a function $S_n \to S_n$ for each n, commuting with the operad structure. A dynamic operad on y can thus be identified with an operad $\mathcal{S}$ equipped with an operad map $\mathcal{S} \to \mathcal{S}$ to itself. ◊

3.3 Dynamic monoidal categories

A monoidal double category is precisely a representable fmc-multicategory as in [3, Section 2], so we can also enrich strict monoidal categories in $\mathbb{O}\mathbf{rg}$.[4] These are similar to $\mathbb{O}\mathbf{rg}$-enriched multicategories, but include many-to-many coalgebras rather than just many-to-one.

Definition 3.7. An $\mathbb{O}\mathbf{rg}$-enriched (henceforth *dynamic*) strict monoidal category A consists of

- a monoid $(A_0, e, *)$ of objects;
- for each $a \in A_0$, a polynomial p_a;
- an isomorphism of polynomials $y \cong p_e$;
- for each $a, a' \in A_0$, an isomorphism of polynomials $p_a \otimes p_{a'} \cong p_{a*a'}$;
- for each $a, b \in A_0$, a $[p_a, p_b]$-coalgebra $\mathbf{S}_{a,b}$;
- for each $a \in A_0$, an "identitor" square in $\mathbb{O}\mathbf{rg}$ as in Eq. (10) left;
- for each $a, b, c \in A_0$, a "compositor" square in $\mathbb{O}\mathbf{rg}$ as in Eq. (10) center; and
- for each $a, a', b, b' \in A_0$, a "productor" square in $\mathbb{O}\mathbf{rg}$ as in Eq. (10) right:

$$
\begin{array}{ccc}
p_a & \xrightarrow{\{\mathrm{id}_{p_a}\}} & p_a \\
\| & \Downarrow & \| \\
p_a & \xrightarrow[\mathbf{S}_{a,a}]{} & p_a
\end{array}
\qquad
\begin{array}{ccccc}
p_a & \xrightarrow{\mathbf{S}_{a,b}} & p_b & \xrightarrow{\mathbf{S}_{b,c}} & p_c \\
\| & & \Downarrow & & \| \\
p_a & & \xrightarrow[\mathbf{S}_{a,c}]{} & & p_c
\end{array}
\qquad
\begin{array}{ccc}
p_a \otimes p_{a'} & \xrightarrow{\mathbf{S}_{a,b} \otimes \mathbf{S}_{a',b'}} & p_b \otimes p_{b'} \\
\wr\| & \Downarrow & \wr\| \\
p_{a*a'} & \xrightarrow[\mathbf{S}_{a*a',b*b'}]{} & p_{b*b'}
\end{array}
\tag{10}
$$

satisfying unit, associativity, and interchange equations (see Definition A.3 for the one-object case). ◊

Similar to Sections 3.1 and 3.2, the sets $S_{a,b}$ form the arrows in an ordinary strict monoidal category underlying A.

For the rest of this paper, we will only be interested in the restricted case of a dynamic monoidal category with object monoid $(\mathbb{N}, 0, +)$, which we call a dynamic PRO.[5] Concretely, this consists of a polynomial interface p (so that in the notation above $p_n := p^{\otimes n}$ for $n \in \mathbb{N}$) along with a $[p^{\otimes m}, p^{\otimes n}]$-coalgebra $S_{m,n}$ for each $m, n \in \mathbb{N}$, equipped with the maps of coalgebras as in (10). The identitors, compositors, productors, and their equations amount to the ability to compose any string diagram of the usual sort for monoidal categories, with each m-to-n box given by a state in $S_{m,n}$, into a new box (i.e. state) with the appropriate sources and targets. We denote a dynamic PRO as $(p, \mathbf{S})$, where $\mathbf{S}$ encodes all of the coalgebras $\mathbf{S}_{m,n}$ that constitute the $\mathbb{O}\mathbf{rg}$-enrichment and the structure maps are implicit.

We now turn to morphisms between dynamic PROs; the interested reader can hopefully find analogous definitions for dynamic categories and operads.

Definition 3.8. A *morphism* of dynamic PROs from $(p, \mathbf{S})$ to $(p', \mathbf{S}')$ is given by a map of polynomials $\phi \colon p \to p'$ and, for each $m, n \in \mathbb{N}$, "commutor" squares as in (11) in $\mathbb{O}\mathbf{rg}$ which commute with the

[4] We use throughout the notion of *strong* enrichment in a monoidal double category from [3, Section 3].

[5] A PRO is the non-symmetric version of a PROP. While all of our examples are in fact symmetric, to keep the paper short we do not describe their symmetry operations.

identitor, compositor, and productor squares.

$$\begin{array}{ccc} p^{\otimes m} & \xrightarrow[\sim]{\mathbf{S}_{m,n}} & p^{\otimes n} \\ {\scriptstyle \phi^{\otimes m}}\downarrow & \Downarrow & \downarrow{\scriptstyle \phi^{\otimes n}} \\ p'^{\otimes m} & \xrightarrow[\mathbf{S}'_{m,n}]{\sim} & p'^{\otimes n} \end{array} \tag{11}$$

◊

This definition of morphism (taken from [3, Section 3]) is the direct theoretical benefit of treating $\mathbb{O}\mathbf{rg}$ as a monoidal double category rather than as a monoidal bicategory (closer to its description in [4]). Otherwise morphisms could either only be easily defined between dynamic PROs with the same interface polynomial, which is needlessly restrictive, or take the form of a $[p,p']$-coalgebra, which seems to be too general to be of much use.

A morphism $(p,\mathbf{S}) \to (p',\mathbf{S}')$ can be interpreted as a way of telling the codomain how to run the domain. The map of polynomials $p \to p'$ specifies how the positions of p can be modeled by those of p' and how the directions of p' are returned as directions of p, while the commutor squares describe how the states of $\mathbf{S}_{m,n}$ can be modeled by those of $\mathbf{S}'_{m,n}$ in a way that respects this change of interface. A type of theorem that we hope to instantiate in future work is of the form "this dynamic structure that we're interested in can be run by (has a map to) this other dynamic structure that we already understand well."

Example 3.9. For a fixed polynomial p, there is a terminal dynamic PRO with interface p, which we denote $\mathbf{S}^{p!}$; here $\mathbf{S}^{p!}_{m,n}$ is the terminal $[p^{\otimes m}, p^{\otimes n}]$-coalgebra for each $m,n \in \mathbb{N}$.

A state in $\mathbf{S}^{p!}$ is a (not necessarily finite) $[p^{\otimes m}, p^{\otimes n}]$-tree. By this we mean a tree co-inductively defined by a root node labeled with a polynomial map $\phi\colon p^{\otimes m} \to p^{\otimes n}$ together with an arrow—whose source is the root and whose target is another $[p^{\otimes m}, p^{\otimes n}]$-tree—assigned to each tuple

$$\big((I_1,\ldots,I_m), i_1,\ldots,i_n\big) \in p^{\otimes m}(1) \times p^{\otimes n}[\phi(I_1,\ldots,I_m)] \tag{12}$$

The action of such a tree is simply the map ϕ labeling its root, and the update sends a tuple as in (12) to the target of its assigned arrow.

The idea is that the state-set of the terminal dynamic PRO encodes all possible trajectories along different actions, and this coalgebra is terminal because from any other coalgebra there is a map to $\mathbf{S}^{p!}_{m,n}$ sending each state to the tree whose root is labeled by the action of the state and whose edges from the root go to the trees for each of the state's possible updates.

To define a dynamic PRO structure on the terminal coalgebra $\mathbf{S}^{p!}$, it only remains to define maps of coalgebras as in Eq. (10), and these are all taken to be the unique map to the terminal $[p^{\otimes m}, p^{\otimes n}]$-coalgebra; the equations hold automatically. This is the terminal dynamic PRO with interface p because for any other such dynamic PRO there is a morphism given by the identity map on p and with commutor squares to $\mathbf{S}^{p!}_{m,n}$ the unique map to the terminal $[p^{\otimes m}, p^{\otimes n}]$-coalgebra. In other words, $\mathbf{S}^{p!}$ *uniquely runs* any other dynamic PRO with interface p. ◊

4 Dynamic Structures in Nature

Our main results are that dynamic structures describe phenomena we see instantiated around us. In this paper, we focus on deep learning and a prediction market in which the reputations of various guess-makers evolve based on how successful they are.

4.1 The prediction market dynamic operad

Fix a finite set X, elements of which we call *outcomes* and intuit to be "all equally likely", define the set Δ^+_X of *guesses on* X as[6]

$$\Delta^+_X := \left\{ \gamma\colon X \to (0,1] \;\middle|\; 1 = \sum_x \gamma(x) \right\}$$

Let Δ^+ denote the operad of finite nowhere-zero probability distributions, where Δ^+_N is defined as above with the natural number N regarded as the N-element set. Then Δ^+_X is an algebra for it: for any $\mu \in \Delta_N$ and $\gamma \in (\Delta^+_X)^N$, we define

$$\mu \cdot \gamma := \left(x \mapsto \sum_{i \in N} \mu_i \cdot \gamma_i(x) \right)$$

and it is easy to check that $(\mu \cdot \gamma) \in \Delta^+_X$, i.e. its components are in bounds $(\mu \cdot \gamma)(x) \in (0,1]$ and it is normalized $\sum_x (\mu \cdot \gamma)(x) = 1$.

We now construct a dynamic operad with interface $p_X \in \mathbf{Poly}$ defined as:

$$p_X := \Delta^+_X y^X$$

and use the Δ^+_N as our state spaces. The idea is that a state $\mu \in \Delta^+_N$ says how much the organization trusts each of its N members (guess-makers) relative to each other. A member's position at a given moment is a report of how much confidence it has in each of the X-many possibilities, represented by its probability distribution.

The action of a trust distribution $\mu \in \Delta^+_N$ is the map of polynomials $p_X^{\otimes N} \to p_X$ which on positions sends $\gamma \in (\Delta^+_X)^N$ to $\mu \cdot \gamma$ and on directions sends $x \in X$ to $(x, \ldots, x) \in X^N$. The idea is that the organization aggregates its members' predictions according to its current trust-distribution, and the outcome is accurately communicated back to each member.

The most interesting part of the dynamic structure is how the trust distribution is updated once predictions are made and a result $x \in X$ is returned. When $N = 0$, there's nothing to do: $\Delta^+_0 = \varnothing$. For membership $N \geq 1$, trust distribution $\mu \in \Delta^+_N$, guesses $\gamma \in (\Delta^+_X)^N$, and outcome $x \in X$, we define the updated trust distribution $\gamma(x) * \mu \in \Delta^+_N$ as

$$\gamma(x) * \mu := \left(i \mapsto \frac{\gamma_i(x)\mu_i}{\sum_j \gamma_j(x)\mu_j} \right).$$

Finally, we describe the operadic structure maps. As Δ^+_1 is a singleton set whose action is the identity on p_X, the identitor $\{\mathrm{id}_{p_X}\} \to \Delta^+_1$ is an isomorphism. The operadic compositor is given by the usual operad structure on (nowhere-zero) distributions:

$$\Delta^+_N \times \Delta^+_{M_1} \times \cdots \times \Delta^+_{M_N} \to \Delta^+_{\sum_i M_i} \qquad (\mu, \nu_1, \ldots, \nu_N) \mapsto \mu \circ \nu := \big((i,j) \mapsto \mu_i \nu_j\big).$$

[6] We assume that each guess assigns a nonzero probability to each possible outcome, which avoids the issues of dividing by zero when updating or permanent loss of a guess-maker's reputation. This should be interpreted as both humility and good strategy on the part of the guess-makers.

Theorem 4.1. *The maps defined above are maps of coalgebras and satisfy the coherence equations of a dynamic operad described in Definition A.2.*

This is proven in Appendix B.

4.2 The gradient descent dynamic PRO

Deep learning uses the algorithm of gradient descent to optimize a choice of function, based on external feedback on its output. This naturally fits into the paradigm of dynamic structures, since functions $\mathbb{R}^m \to \mathbb{R}^n$ can form the states of a polynomial coalgebra, with the feedback providing the information needed to update the choice of function. These functions can be composed and juxtaposed in a way that preserves the updates. That is, the composite of gradient descenders is a gradient descender.

Definition 4.2. For the rest of this section, we will use the state sets

$$S_{m,n} := \left\{(M \in \mathbb{N}, f : \mathbb{R}^{M+m} \to \mathbb{R}^n, p \in \mathbb{R}^M) \,\middle|\, f \text{ is differentiable}\right\}. \qquad \lozenge$$

The idea is that these states are the possible parameters among which a gradient descender is meant to find the optimal choice, while f dictates how the parameter affects the resulting function $f(p,-)$. In the dynamics of these states described below, only the value of the parameter p will be updated; the parameter-space dimension M and the parameterized function f will remain fixed, though network composition of gradient descenders will involve combining these data in nontrivial ways. Fix $\epsilon > 0$.

For every $x \in \mathbb{R}$, let $T_x\mathbb{R}$ denote the tangent space at x; for all practical purposes $T_x\mathbb{R}$ can be regarded as simply $\mathbb{R}$, but in both the description of polynomials as bundles and the intuition for this example it makes sense to use the tangent space. We proceed to define a dynamic PRO with interface $t := \sum_{x\in\mathbb{R}} y^{T_x\mathbb{R}}$ and coalgebras $\mathbf{S}_{m,n}$ which update the state sets $S_{m,n}$ from Definition 4.2 using gradient descent. The PRO structure maps encode how networks of gradient descenders can be composed into a single gradient descender with a larger parameter space.

Definition 4.3. The $[t^{\otimes m}, t^{\otimes n}]$-coalgebra structure on $S_{m,n}$ is given by

- On positions, the action $\mathrm{act}^{\beta}_{M,f,p} : \mathbb{R}^m \to \mathbb{R}^n$ is given by $f(p,-)$.
- For $x \in \mathbb{R}^m$, the action $\mathrm{act}^{\beta}_{M,f,p}(x,-) \colon T_{f(p,x)}\mathbb{R}^n \to T_x\mathbb{R}^m$ on directions sends $y \in T_{f(p,x)}$ to $\pi_m(Df)^\top \cdot y$.
- The update function $\mathrm{upd}^{\beta}_{M,f,p}$ sends $x \in \mathbb{R}^m$ and $y \in T_{f(p,x)}$ to $(M, f, p + \epsilon\pi_M(Df)^\top \cdot y)$ for our fixed ϵ. $\lozenge$

The action of a state as a map $t^{\otimes m} \to t^{\otimes n}$ is given by applying the parameterized function f with the parameter p, resulting in a function $\mathbb{R}^m \to \mathbb{R}^n$ as desired. The transpose $(Df)^\top$ of the derivative of f sends a feedback vector $y \in T_{f(p,x)}\mathbb{R}^n$, which can be interpreted as the difference in $\mathbb{R}^n$ between the "correct" result for x and the current approximation $f(p,x)$, to the corresponding "correction" to (p,x) in $\mathbb{R}^{M+m}$. The projection of this correction to $T_x\mathbb{R}^m$ provides the action of the state on directions, which in a network will then be further propagated back to the gradient descender which had output x. The projection to $T_p\mathbb{R}^M$ provides the direction and magnitude in which to update the parameters (scaled by the "learning rate" ϵ).

Thus far, we have provided the data of the polynomial t and the $[t^{\otimes m}, t^{\otimes n}]$-coalgebras $\mathbf{S}_{m,n}$ needed to define a dynamic PRO. We now define the identitor, compositor, and productor morphisms of coalgebras presented by the squares in Definition 3.7.

- The identitors $\{\mathrm{id}_{t^{\otimes n}}\} \to \mathbf{S}_{n,n}$ send the unique state in the domain to the state

$$(0, \mathrm{id}_{\mathbb{R}^n}, 0) \in S_{n,n}.$$

- The compositors $\mathbf{S}_{\ell,m} \mathbin{⨟} \mathbf{S}_{m,n} \to \mathbf{S}_{\ell,n}$ send the pair $((L, f, p), (M, g, q))$ to

$$\left(M + L, g(-, f(-,-)) \colon \mathbb{R}^{M+L+\ell} \xrightarrow{\mathrm{id}\times f} \mathbb{R}^{M+m} \xrightarrow{g} \mathbb{R}^n, (q, p) \in \mathbb{R}^{M+L}\right).$$

- The productors $\mathbf{S}_{m,n} \otimes \mathbf{S}_{m',n'} \to \mathbf{S}_{m+m',n+n'}$ send the pair $((M, f, p), (M', f', p'))$ to

$$(M + M', (f, f'), (p, p')).$$

These structure maps ensure that whenever two gradient descenders are combined in series or parallel, the resulting composite descender retains the parameter spaces of both. Likewise when the input or output of a descender is wired past some other descender in a network, it does not contribute any new parameters and merely preserves its input/output until plugged into a descender. The following is proven in Appendix B.

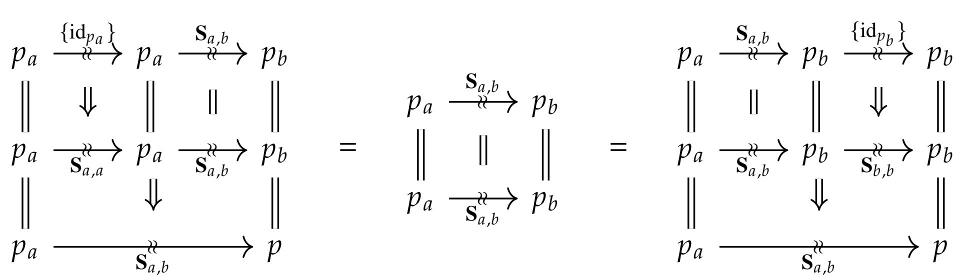
Theorem 4.4. *The maps defined above are maps of coalgebras and satisfy the coherence equations of a dynamic PRO described in Definition A.3.*

A Coherence Equations

We now present the equations that must be satisfied by the structure maps in dynamic categories, operads and PROs. While we only provide the equations for the single-object variant of dynamic multicategories and monoidal categories, respectively, the equations in the general case are entirely analogous.

Definition A.1. The equations between the identitors and compositors in a dynamic category are as follows:

- The left and right unit laws



(13)

- The associativity law

$$
\begin{array}{ccccccc}
p_a & \xrightarrow[\sim]{\mathbf{S}_{a,b}} & p_b & \xrightarrow[\sim]{\mathbf{S}_{b,c}} & p_c & \xrightarrow[\sim]{\mathbf{S}_{c,d}} & p_d \\
\| & & \Downarrow & & \| & \| & \| \\
p_a & & \xrightarrow[\mathbf{S}_{a,c}]{\sim} & & p_c & \xrightarrow[\mathbf{S}_{c,d}]{\sim} & p_d \\
\| & & & \Downarrow & & & \| \\
p_a & & & \xrightarrow[\mathbf{S}_{a,d}]{\sim} & & & p_d
\end{array}
\quad = \quad
\begin{array}{ccccccc}
p_a & \xrightarrow[\sim]{\mathbf{S}_{a,b}} & p_b & \xrightarrow[\sim]{\mathbf{S}_{b,c}} & p_c & \xrightarrow[\sim]{\mathbf{S}_{c,d}} & p_d \\
\| & \| & \| & & \Downarrow & & \| \\
p_a & \xrightarrow[\mathbf{S}_{a,b}]{\sim} & p_b & & \xrightarrow[\mathbf{S}_{b,d}]{\sim} & & p_d \\
\| & & & \Downarrow & & & \| \\
p_a & & & \xrightarrow[\mathbf{S}_{a,d}]{\sim} & & & p_d
\end{array}
\tag{14}
$$

◊

We now present the equations for dynamic operads. These equations derive directly from the definition of operads, namely the associativity and unitality of operadic composition, but unlike the equations above only involve a single polynomial p.

Definition A.2. The equations between the identitors and compositors in a dynamic operad are as follows:

- The left and right unit laws

$$
\begin{array}{ccccc}
p^{\otimes n} & \xrightarrow[\sim]{\{\mathrm{id}_p\}^{\otimes n}} & p^{\otimes n} & \xrightarrow[\sim]{\mathbf{S}_n} & p \\
\| & \Downarrow & \| & \| & \| \\
p^{\otimes n} & \xrightarrow[\mathbf{S}_1^{\otimes n}]{\sim} & p^{\otimes n} & \xrightarrow[\mathbf{S}_n]{\sim} & p \\
\| & & \Downarrow & & \| \\
p^{\otimes n} & & \xrightarrow[\mathbf{S}_n]{\sim} & & p
\end{array}
\quad = \quad
\begin{array}{ccc}
p^{\otimes n} & \xrightarrow[\sim]{\mathbf{S}_n} & p \\
\| & \| & \| \\
p^{\otimes n} & \xrightarrow[\mathbf{S}_n]{\sim} & p
\end{array}
\quad = \quad
\begin{array}{ccccc}
p^{\otimes n} & \xrightarrow[\sim]{\mathbf{S}_n} & p & \xrightarrow[\sim]{\{\mathrm{id}_p\}} & p \\
\| & \| & \| & \Downarrow & \| \\
p^{\otimes n} & \xrightarrow[\mathbf{S}_n]{\sim} & p & \xrightarrow[\mathbf{S}_1]{\sim} & p \\
\| & & \Downarrow & & \| \\
p^{\otimes n} & & \xrightarrow[\mathbf{S}_n]{\sim} & & p
\end{array}
\tag{15}
$$

- The associativity law

$$
\begin{array}{ccccccc}
p^{\otimes \ell_{1,1}} \otimes \cdots \otimes p^{\otimes \ell_{n,m_n}} & \xrightarrow[\sim]{\bigotimes_{i,j} \mathbf{S}_{\ell_{i,j}}} & p^{\otimes m_1} \otimes \cdots \otimes p^{\otimes m_n} & \xrightarrow[\sim]{\bigotimes_i \mathbf{S}_{m_i}} & p^{\otimes n} & \xrightarrow[\sim]{\mathbf{S}_n} & p \\
\wr\| & & \Downarrow & & \| & \| & \| \\
p^{\otimes(\sum_j \ell_{1,j})} \otimes \cdots \otimes p^{\otimes(\sum_j \ell_{n,j})} & & \xrightarrow{\bigotimes_i \mathbf{S}_{\sum_j \ell_{1,j}}} & & p^{\otimes n} & \xrightarrow[\mathbf{S}_n]{\sim} & p \\
\wr\| & & & \Downarrow & & & \| \\
p^{\otimes(\sum_{i,j} \ell_{i,j})} & & & \xrightarrow[\mathbf{S}_{\sum_{i,j} \ell_{i,j}}]{\sim} & & & p
\end{array}
$$

$$
= \tag{16}
$$

$$
\begin{array}{ccccccc}
p^{\otimes \ell_{1,1}} \otimes \cdots \otimes p^{\otimes \ell_{n,m_n}} & \xrightarrow[\wr]{\bigotimes_{i,j} \mathbf{S}_{\ell_{i,j}}} & p^{\otimes m_1} \otimes \cdots \otimes p^{\otimes m_n} & \xrightarrow[\wr]{\bigotimes_i \mathbf{S}_{m_i}} & p^{\otimes n} & \xrightarrow[\wr]{\mathbf{S}_n} & p \\
\| & \| & \wr\| & & \Downarrow & & \| \\
p^{\otimes \ell_{1,1}} \otimes \cdots \otimes p^{\otimes \ell_{n,m_n}} & \xrightarrow[\bigotimes_{i,j} \mathbf{S}_{\ell_{i,j}}]{\wr} & p^{\otimes (\sum_i m_i)} & & \xrightarrow[\mathbf{S}_{\sum_i m_i}]{\wr} & & p \\
\wr\| & & \Downarrow & & & & \| \\
p^{\otimes (\sum_{i,j} \ell_{i,j})} & & & \xrightarrow[\mathbf{S}_{\sum_{i,j} \ell_{i,j}}]{\wr} & & & p
\end{array}
$$

◊

The equations for dynamic PROs below are similarly derived from the definition of monoidal categories, namely that composition and products of arrows are associative and unital (giving the associativity and unitality equations for compositors and productors) and products are functorial (giving the interchange equations).

Definition A.3. The equations between the identitors, compositors, and productors in a dynamic PRO are as follows:

- The identitor interchange law

$$
\begin{array}{ccc}
p^{\otimes n} \otimes p^{\otimes n'} & \xrightarrow[\wr]{\{\mathrm{id}_{p^{\otimes n}}\} \otimes \{\mathrm{id}_{p^{\otimes n'}}\}} & p^{\otimes n} \otimes p^{\otimes n'} \\
\| & \Downarrow & \| \\
p^{\otimes n} \otimes p^{\otimes n'} & \xrightarrow{\mathbf{S}_{n,n} \otimes \mathbf{S}_{n',n'}} & p^{\otimes n} \otimes p^{\otimes n'} \\
\wr\| & \Downarrow & \wr\| \\
p^{\otimes (n+n')} & \xrightarrow[\mathbf{S}_{n+n',n+n'}]{\wr} & p^{\otimes (n+n')}
\end{array}
\quad = \quad
\begin{array}{ccc}
p^{\otimes n} \otimes p^{\otimes n'} & \xrightarrow[\wr]{\{\mathrm{id}_{p^{\otimes n}}\} \otimes \{\mathrm{id}_{p^{\otimes n'}}\}} & p^{\otimes n} \otimes p^{\otimes n'} \\
\wr\| & \wr\| & \wr\| \\
p^{\otimes (n+n')} & \xrightarrow{\{\mathrm{id}_{p^{\otimes (n+n')}}\}} & p^{\otimes (n+n')} \\
\| & \Downarrow & \| \\
p^{\otimes (n+n')} & \xrightarrow[\mathbf{S}_{n+n',n+n'}]{\wr} & p^{\otimes (n+n')}
\end{array}
\qquad (17)
$$

- The compositor interchange law

$$
\begin{array}{ccccc}
p^{\otimes \ell} \otimes p^{\otimes \ell'} & \xrightarrow[\wr]{\mathbf{S}_{\ell,m} \otimes \mathbf{S}_{\ell',m'}} & p^{\otimes m} \otimes p^{\otimes m'} & \xrightarrow[\wr]{\mathbf{S}_{m,n} \otimes \mathbf{S}_{m',n'}} & p^{\otimes n} \otimes p^{\otimes n'} \\
\wr\| & \Downarrow & \wr\| & \Downarrow & \wr\| \\
p^{\otimes (\ell+\ell')} & \xrightarrow[\mathbf{S}_{\ell+\ell',m+m'}]{\wr} & p^{\otimes (m+m')} & \xrightarrow[\mathbf{S}_{m+m',n+n'}]{\wr} & p^{\otimes (n+n')} \\
\| & & \Downarrow & & \| \\
p^{\otimes (\ell+\ell')} & & \xrightarrow[\mathbf{S}_{\ell+\ell',n+n'}]{\wr} & & p^{\otimes (n+n')}
\end{array}
$$

$$= \qquad (18)$$

$$\begin{array}{ccccc}
p^{\otimes \ell}\otimes p^{\otimes \ell'} & \xrightarrow[\sim]{\mathbf{S}_{\ell,m}\otimes \mathbf{S}_{\ell',m'}} & p^{\otimes m}\otimes p^{\otimes m'} & \xrightarrow[\sim]{\mathbf{S}_{m,n}\otimes \mathbf{S}_{m',n'}} & p^{\otimes n}\otimes p^{\otimes n'}\\
\| & & \Downarrow & & \|\\
p^{\otimes \ell}\otimes p^{\otimes \ell'} & & \xrightarrow{\mathbf{S}_{\ell,n}\otimes \mathbf{S}_{\ell',n'}} & & p^{\otimes n}\otimes p^{\otimes n'}\\
\wr\| & & \Downarrow & & \wr\|\\
p^{\otimes(\ell+\ell')} & & \xrightarrow[\mathbf{S}_{\ell+\ell',n+n'}]{\sim} & & p^{\otimes(n+n')}
\end{array}$$

- The compositor associativity law

$$\begin{array}{ccccccc}
p^{\otimes k} & \xrightarrow[\sim]{\mathbf{S}_{k,\ell}} & p^{\otimes \ell} & \xrightarrow[\sim]{\mathbf{S}_{\ell,m}} & p^{\otimes m} & \xrightarrow[\sim]{\mathbf{S}_{m,n}} & p^{\otimes n}\\
\| & & \Downarrow & & \| & \| & \|\\
p^{\otimes k} & & \xrightarrow[\mathbf{S}_{k,m}]{\sim} & & p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,n}]{\sim} & p^{\otimes n}\\
\| & & & \Downarrow & & & \|\\
p^{\otimes k} & & & \xrightarrow[\mathbf{S}_{k,n}]{\sim} & & & p^{\otimes n}
\end{array}
\quad = \quad
\begin{array}{ccccccc}
p^{\otimes k} & \xrightarrow[\sim]{\mathbf{S}_{k,\ell}} & p^{\otimes \ell} & \xrightarrow[\sim]{\mathbf{S}_{\ell,m}} & p^{\otimes m} & \xrightarrow[\sim]{\mathbf{S}_{m,n}} & p^{\otimes n}\\
\| & \| & \| & & \Downarrow & & \|\\
p^{\otimes k} & \xrightarrow[\mathbf{S}_{k,\ell}]{\sim} & p^{\otimes \ell} & & \xrightarrow[\mathbf{S}_{\ell,n}]{\sim} & & p^{\otimes n}\\
\| & & & \Downarrow & & & \|\\
p^{\otimes k} & & & \xrightarrow[\mathbf{S}_{k,n}]{\sim} & & & p^{\otimes n}
\end{array}
\qquad (19)$$

- The compositor unit laws

$$\begin{array}{ccccc}
p^{\otimes m} & \xrightarrow[\sim]{\{\mathrm{id}_{p^{\otimes m}}\}} & p^{\otimes m} & \xrightarrow[\sim]{\mathbf{S}_{m,n}} & p^{\otimes n}\\
\| & \Downarrow & \| & \| & \|\\
p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,m}]{\sim} & p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,n}]{\sim} & p^{\otimes n}\\
\| & & \Downarrow & & \|\\
p^{\otimes m} & & \xrightarrow[\mathbf{S}_{m,n}]{\sim} & & p^{\otimes n}
\end{array}
\quad = \quad
\begin{array}{ccc}
p^{\otimes m} & \xrightarrow[\sim]{\mathbf{S}_{m,n}} & p^{\otimes n}\\
\| & \| & \|\\
p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,n}]{\sim} & p^{\otimes n}
\end{array}
\quad = \quad
\begin{array}{ccccc}
p^{\otimes m} & \xrightarrow[\sim]{\mathbf{S}_{m,n}} & p^{\otimes n} & \xrightarrow[\sim]{\{\mathrm{id}_{p^{\otimes m}}\}} & p^{\otimes n}\\
\| & \| & \| & \Downarrow & \|\\
p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,n}]{\sim} & p^{\otimes n} & \xrightarrow[\mathbf{S}_{n,n}]{\sim} & p^{\otimes n}\\
\| & & \Downarrow & & \|\\
p^{\otimes m} & & \xrightarrow[\mathbf{S}_{m,n}]{\sim} & & p^{\otimes n}
\end{array}
\qquad (20)$$

- The productor associativity law

$$\begin{array}{ccc}
p^{\otimes m}\otimes p^{\otimes m'}\otimes p^{\otimes m''} & \xrightarrow[\sim]{\mathbf{S}_{m,n}\otimes \mathbf{S}_{m',n'}\otimes \mathbf{S}_{m'',n''}} & p^{\otimes n}\otimes p^{\otimes n'}\otimes p^{\otimes n''}\\
\wr\| & \Downarrow & \wr\|\\
p^{\otimes(m+m')}\otimes p^{\otimes m''} & \xrightarrow{\mathbf{S}_{m+m',n+n'}\otimes \mathbf{S}_{m'',n''}} & p^{\otimes(n+n')}\otimes p^{\otimes n''}\\
\wr\| & \Downarrow & \wr\|\\
p^{\otimes(m+m'+m'')} & \xrightarrow[\mathbf{S}_{m+m'+m'',n+n'+n''}]{\sim} & p^{\otimes(n+n'+n'')}
\end{array}$$

$$= \qquad (21)$$

$$
\begin{array}{ccc}
p^{\otimes m} \otimes p^{\otimes m'} \otimes p^{\otimes m''} & \xrightarrow[\wr\wr]{\mathbf{S}_{m,n} \otimes \mathbf{S}_{m',n'} \otimes \mathbf{S}_{m'',n''}} & p^{\otimes n} \otimes p^{\otimes n'} \otimes p^{\otimes n''} \\
\wr\Vert & \Downarrow & \wr\Vert \\
p^{\otimes m} \otimes p^{\otimes (m+m'')} & \xrightarrow{\mathbf{S}_{m,n} \otimes \mathbf{S}_{m'+m'',n'+n''}} & p^{\otimes n} \otimes p^{\otimes (n'+n'')} \\
\wr\Vert & \Downarrow & \wr\Vert \\
p^{\otimes (m+m'+m'')} & \xrightarrow[\mathbf{S}_{m+m'+m'',n+n'+n''}]{\wr\wr} & p^{\otimes (n+n'+n'')}
\end{array}
$$

- The productor unit laws

$$
\begin{array}{ccc}
p^{\otimes m} & \xrightarrow[\wr\wr]{\mathbf{S}_{m,n}} & p^{\otimes n} \\
\wr\Vert & \wr\Vert & \wr\Vert \\
p^{\otimes 0} \otimes p^{\otimes m} & \xrightarrow{\{\mathrm{id}_{p^{\otimes 0}}\} \otimes \mathbf{S}_{m,n}} & p^{\otimes 0} \otimes p^{\otimes n} \\
\Vert & \Downarrow & \Vert \\
p^{\otimes 0} \otimes p^{\otimes m} & \xrightarrow{\mathbf{S}_{0,0} \otimes \mathbf{S}_{m,n}} & p^{\otimes 0} \otimes p^{\otimes n} \\
\wr\Vert & \Downarrow & \wr\Vert \\
p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,n}]{\wr\wr} & p^{\otimes n}
\end{array}
\; = \;
\begin{array}{ccc}
p^{\otimes m} & \xrightarrow[\wr\wr]{\mathbf{S}_{m,n}} & p^{\otimes n} \\
\Vert & \Vert & \Vert \\
p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,n}]{\wr\wr} & p^{\otimes n}
\end{array}
\; = \;
\begin{array}{ccc}
p^{\otimes m} & \xrightarrow[\wr\wr]{\mathbf{S}_{m,n}} & p^{\otimes n} \\
\wr\Vert & \wr\Vert & \wr\Vert \\
p^{\otimes m} \otimes p^{\otimes 0} & \xrightarrow{\mathbf{S}_{m,n} \otimes \{\mathrm{id}_{p^{\otimes 0}}\}} & p^{\otimes n} \otimes p^{\otimes 0} \\
\Vert & \Downarrow & \Vert \\
p^{\otimes m} \otimes p^{\otimes 0} & \xrightarrow{\mathbf{S}_{m,n} \otimes \mathbf{S}_{0,0}} & p^{\otimes n} \otimes p^{\otimes 0} \\
\wr\Vert & \Downarrow & \wr\Vert \\
p^{\otimes m} & \xrightarrow[\mathbf{S}_{m,n}]{\wr\wr} & p^{\otimes n}
\end{array}
\tag{22}
$$

◊

B Proofs of Dynamic Structure

We now proceed to prove that the coalgebras and structure maps defined above for organized predictions and gradient descent form dynamic structures. In each case, it suffices to show that the structure maps on states preserve coalgebra structure, and that the equations in Definition A.2 or Definition A.3, respectively, are satisfied.

Proof of Theorem 4.1. The operad equations are all satisfied as Δ^+ is known to be an operad, and morphisms of coalgebras are entirely determined by a function between the state sets. It then remains only to show that the identitor and compositor as defined in Section 4.1 commute with actions and updates. This is clearly true for the identitor as it is an isomorphism, so we focus on the compositor.

For the compositor to commute with actions on positions is the claim that Δ^+_X is an algebra for the operad Δ^+; it means that for $\mu \in \Delta^+_N$, $\nu_1 \in \Delta^+_{M_1}$, ..., $\nu_N \in \Delta^+_{M_N}$, and $\gamma_{i,j} \in \Delta^+_X$ for $i = 1, \ldots, N$ and $j = 1, \ldots, M_i$, we have

$$\sum_i \mu_i \left(\sum_j \nu_j \gamma_{i,j} \right) = \sum_{i,j} (\mu_i \nu_{i,j}) \gamma_{i,j},$$

which is clearly the case.

The compositor commutes with actions on directions because in $(\mathbf{S}_{M_1} \otimes \cdots \otimes \mathbf{S}_{M_N}) \mathbin{⨟} \mathbf{S}_N$ the action of $(\nu_1, \ldots, \nu_N, \mu)$ sends an outcome

$$x \in X = p_X[\mu \cdot (\nu \cdot \gamma)]$$

to

$$(x,\dots,x)\in X^N = p_X^{\otimes N}[\nu_1\cdot\gamma_1,\dots,\nu_N\cdot\gamma_N]$$

and then to

$$(x,\dots,x)\in X^{\sum_i M_i} = p_X^{\otimes \sum_i M_i}[\gamma_{1,1},\dots,\gamma_{N,M_N}],$$

while in $\mathbf{S}_{\sum_i M_i}$ the action of $\mu\circ\nu$ sends $x\in X$ to $(x,\dots,x)\in X^{\sum_i M_i}$ directly.

It then only remains to show that the compositor commutes with updates. Using the shorthand notation $\nu=(\nu_1,\dots,\nu_N)$ and $\gamma=(\gamma_1,\dots,\gamma_N)=(\gamma_{1,1},\dots,\gamma_{N,M_N})$ already employed above, to show that the composite of the updates of μ,ν agrees with the update of the composite $\mu\circ\nu$ amounts to the equation

$$\gamma(x)*(\mu\circ\nu)=\big((\nu\cdot\gamma)(x)*\mu\big)\circ\big(\gamma(x)*\nu\big) \tag{23}$$

for any $x\in X$. Here $\nu\cdot\gamma$ denotes $(\nu_1\cdot\gamma_1,\dots,\nu_N\cdot\gamma_N)$ and $\gamma(x)*\nu$ denotes $(\gamma_1(x)*\nu_1,\dots,\gamma_N(x)*\nu_N)$. On the (i,j)-component of these distributions, (23) unwinds to

$$\frac{\gamma_{i,j}(x)(\mu_i\nu_{i,j})}{\sum_{i',j'}\gamma_{i',j'}(x)(\mu_{i'}\nu_{i',j'})}=\left(\frac{\sum_{j'}(\nu_{i,j'}\gamma_{i,j'}(x))\mu_i}{\sum_{i'}\sum_{j'}(\nu_{i',j'}\gamma_{i',j'}(x))\mu_{i'}}\right)\left(\frac{\gamma_{i,j}(x)\nu_{i,j}}{\sum_{j'}\nu_{i,j'}\gamma_{i,j'}(x)}\right),$$

which is easily seen to hold by extracting μ_i from the first fraction on the right hand side and cancelling the sums over j'. □

Proof of Theorem 4.4. The unit and associativity equations follow immediately from associativity and unitality of addition, cartesian products, and function composition. The interchange equations follow from the preservation of 0 under addition and identity functions under cartesian products, the analogous interchange property of function composition and cartesian products of functions, and the fact that the compositors and productors act the same way on the parameters and their dimension.

It then remains only to show that the identitors, compositors, and productors are morphisms of coalgebras. This is immediate for the productors, as each component of the action and update functions respects the cartesian products of functions and parameters that define them, so we proceed only for the identitors and compositors.

For the identitors, the state $(0,\mathrm{id}_{\mathbb{R}^n},0)$ acts as the identity function on $\mathbb{R}^n$ and on directions by the transpose of its derivative, which is also the identity. The updates in the coalgebras $\mathbf{S}_{n,n}$ only modify the parameter p, so as the parameter here is 0-dimensional this state is never changed by the update function, as is the case in the coalgebra $\{\mathrm{id}_{t^{\otimes n}}\}$. Therefore this function is a map of coalgebras.

The compositors preserve the component of the action on positions as, for states

$$(L\in\mathbb{N}, f\colon\mathbb{R}^{L+\ell}, p\in\mathbb{R}^L)\quad\text{and}\quad(M\in\mathbb{N}, g\colon\mathbb{R}^{M+m}, q\in\mathbb{R}^M),$$

we have

$$g(q,-)\circ f(p,-)=g(-,f(-,-))(q,p,-).$$

This may seem like a trivial rewriting, but it illustrates how the compositor was defined in order for the action to be preserved, as on the left we have the composite of the actions on positions as in $\mathbf{S}_{\ell,m}\mathbin{⨟}\mathbf{S}_{m,n}$, and on the right we apply the compositor and take the action of the resulting state in $\mathbf{S}_{\ell,n}$.

To show that the compositor preserves both the action on directions and the update we note that by the chain rule, for $x \in \mathbb{R}^\ell$ and $z \in T_{g(q,f(p,x))}$,

$$D\left(g(-,f(-,-))\right)^\top z = Df^\top \cdot \pi_m(Dg^\top \cdot z) \in T_{(p,x)}\mathbb{R}^{L+\ell}.$$

Applying π_ℓ to both sides above shows that the compositor preserves the action on directions, as on the left we have the action on directions after applying the compositor and on the right we have the composition of the actions of (L,f,p) and (M,g,q) on directions as in $\mathbf{S}_{\ell,m} \fatsemi \mathbf{S}_{m,n}$.

Finally for updates, we observe by the chain rule that the update rule in $\mathbf{S}_{\ell,n}$ agrees with that in $\mathbf{S}_{\ell,m} \fatsemi \mathbf{S}_{m,n}$ under the compositor, as either way for x,z as above the composite state of (L,f,p) and (M,g,q) updates to

$$\left(M+L, g(-,f(-,-)), \left(p+\epsilon\pi_L(Df^\top \cdot \pi_m(Dg^\top \cdot z)), q+\epsilon\pi_M(Dg^\top \cdot z)\right)\right). \qquad \square$$

References

[1] Tom Leinster (1999): *Generalized enrichment for categories and multicategories*, doi:10.48550/arXiv.math/9901139. Available at `https://arxiv.org/abs/math/9901139`.

[2] Nelson Niu & David I.Spivak (2021): *Collectives: Compositional protocols for contributions and returns*, doi:10.48550/arXiv.2112.11518. Available at `https://arxiv.org/abs/2112.11518`.

[3] Brandon T. Shapiro (2022): *Enrichment of Algebraic Higher Categories*, doi:10.48550/arXiv.2205.12235. Available at `https://arxiv.org/abs/2205.12235`.

[4] David I.Spivak (2021): *Learners' Languages*. In *Proceedings of the 4th Annual Conference on Applied Category Theory*, EPTCS, Cambridge, UK, doi:10.1145/2815072.2815075.

Extending Resource Monotones using Kan Extensions

Robin Cockett
Department of Computer Science, University of Calgary, Alberta, Canada
Institute for Quantum Science and Technology, University of Calgary, Alberta, Canada

Isabelle Jianing Geng   Carlo Maria Scandolo
Department of Mathematics & Statistics, University of Calgary, Alberta, Canada
Institute for Quantum Science and Technology, University of Calgary, Alberta, Canada

Priyaa Varshinee Srinivasan
Department of Computer Science, University of Calgary, Alberta, Canada
National Institute of Standards and Technology, Maryland, USA

In this paper we generalize the framework proposed by Gour and Tomamichel regarding extensions of monotones for resource theories. A monotone for a resource theory assigns a real number to each resource in the theory signifying the utility or the value of the resource. Gour and Tomamichel studied the problem of extending monotones using set-theoretical framework when a resource theory embeds fully and faithfully into the larger theory. One can generalize the problem of computing monotone extensions to scenarios when there exists a functorial transformation of one resource theory to another instead of just a full and faithful inclusion. In this article, we show that (point-wise) Kan extensions provide a precise categorical framework to describe and compute such extensions of monotones. To set up monontone extensions using Kan extensions, we introduce partitioned categories (pCat) as a framework for resource theories and pCat functors to formalize relationship between resource theories. We describe monotones as pCat functors into $([0,\infty],\leq)$, and describe extending monotones along any pCat functor using Kan extensions. We show how our framework works by applying it to extend entanglement monotones for bipartite pure states to bipartite mixed states, to extend classical divergences to the quantum setting, and to extend a non-uniformity monotone from classical probabilistic theory to quantum theory.

1 Introduction

Resource theories [19, 6, 9] in physics model systems in which certain operations considered to be 'free of cost' among of the set of all operations. For example, placing a glass of chilled water at room temperature warms up the water to the ambient temperature. In this context, operations that change the temperature of the water to be in equilibrium with the ambient temperature are considered to be free. In order to produce a "resourceful state" — for example, a glass of chilled water — one requires non-free transformations, such as a fridge, which consumes electricity. Resource theories have been successfully used to study, among other examples, thermodynamical systems [13, 25], entanglement [20, 16], and coherence [36].

A central question in the resource-theoretic modelling of systems is: *given two resources, is there a free transformation to convert one resource into the other?* The answer to this question imposes a preorder on resources which captures their value or usefulness. Intuitively, a resource is more valuable than another if, by possessing the former, we are given access to a larger set of resources including the latter through free transformations. This not a partial order, because their may be different resources

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 203–223, doi:10.4204/EPTCS.380.12

that can be converted freely into each other. Such resources are considered equivalent. In this way, we can set up a partial order on the equivalence classes of resources. One way to define such an order is to quantify resources by introducing monotones, which are a order-preserving maps from the set of all resources into $[0,\infty]$ [9]. Monotones assign a value to resources that is compatible with the preorder, viz. with their usefulness. Monotones often have a physical meaning, such as in the resource theories of quantum thermodynamics [25], where, for systems at a fixed temperature, free energy is a monotone, and for isolated systems, entropy is the natural monotone.

Given a monotone M for a resource theory which embeds in a larger theory, a natural question to ask is whether the monotone M for the smaller theory can be used to quantify the resources in the larger resource theory. This question arises from the observation that that resources exclusive to the larger theory can possibly be converted to resources contained in the smaller theory, and vice versa. It turns out that one can always compute the optimal upper and lower bound for the value of every resource in the larger theory. In other words, it is possible to extend the monotone M to give optimal upper and lower bounds respectively on the value of resources in the larger theory.

In [17], Gour and Tomamichel presented a set-theoretical framework for extending monotones from a subset of resources to the entire set of resources. A similar construction was also introduced by Gonda and Spekkens in [12]. Given a monotone M over a subset of states, they compute 'minimal' and 'maximal' extensions of the monotone to the entire set of states. In this article, we show that these extensions are special cases of more general categorical concepts, called (point-wise) left and right Kan extensions [21, 5, 26, 30]. Kan extensions deal with optimally extending a functor $F:\mathbb{X}\longrightarrow\mathbb{A}$ along another functor $K:\mathbb{Y}\longrightarrow\mathbb{A}$ to give two functors: $\overline{F}_K:\mathbb{Y}\longrightarrow A$ called the left Kan extension of F along K, and $\underline{F}_K:\mathbb{Y}\longrightarrow\mathbb{A}$ called the right Kan extension of F along K. The right Kan extension can be interpreted as the most conservative extension of F along K and the left Kan extension as the most liberal extension of F along K.

We first introduce partitioned Categories (pCats) as a framework for resource theories. Partitioned categories are categories with a chosen subcategory of free transformations. The subcategory includes all the objects of the parent category, in other words, the inclusion of the subcategory into the parent category is bijective on objects. Relationships between resource theories are set up as pCat functors. In this article, since we consider monotones which are not necessarily additive, thus we do not demand symmetric monoidal structure on pCats.

Given a resource theory, the necessary and sufficient conditions for transformations of resources can be encoded as a pCat functor from the resource theory into a preorder. We call such a pCat functor as a preorder collapse. A resource monotone is a preorder collapse into $([0,\infty],\leq)$. In resource theories, contravariant rather than covariant monotones are encountered more frequently, the reason being if resource A can be transformed to resource B using only free transformation(s), then the value of A is considered to be at least as high as the value of B. We refer to a contravariant resource monotone as an op-monotone. The distinction between monotones and op-monotones is important in the computation of monotone extensions. The categorical descriptions of pCats, pCat functors, preorder collapse, monotones are discussed with various running examples in Section 3.1. Table 1, we briefly summarizes the functors of resource theories introduced in this article.

Having set up monotones as pCat functors, optimal extensions of monotones along any pCat functor are given by their left and right Kan extensions. Lemma 3.35 examines the properties of the monotone extensions thus computed, and prove that such extensions are optimal and monotonic. In Lemma 3.37, we show that extending a monotone along a full and faithful functor recovers the case described in [17] by Gour and Tomamichel and by Gonda and Spekkens in [12] (therein called yield and cost constructions). We apply the Kan extension framework for monotones to extend classical divergences to the quantum

pCat functors	functors which preserve free transformations
Preorder collapse	a pCat functor whose codomain category is a preorder
Monotone	a pCat functor whose codomain category is $([0,\infty],\leq)$
Op-monotone	a pCat functor whose codomain category is $([0,\infty],\geq)$

,

Table 1: Functors for resource theories

setting, to extend bipartite pure states entanglement monotone to mixed states, and to extend Shannon entropy as a measure of non-uniformity from classical probabilistic theory to quantum theory. Section 3.4 is dedicated to setting up the Kan extension framework for monotones, and to studying the extension properties and its applications.

Notation: *In this paper, we use bold letters $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{D}$ to denote categories. We use uppercase letters to denote both objects in the categories and functors between categories, whose meaning will be clear from the context. Lowercase letters f,g,h,π are reserved for maps in the categories. Let X,Y,Z be objects, and let $X \xrightarrow{f} Y$, $Y \xrightarrow{g} Z$ be two arrows, we denote the composition of the two arrows $X \xrightarrow{f} Y \xrightarrow{g} Z$ as fg, and similar notations apply for the composition of functors.*

2 An introduction to Kan extensions

Kan extensions [21, 5, 26, 30] are a broadly applicable notion which is quite central to category theory. Indeed, Mac Lane in his book 'Categories for working Mathematician' [26] gave the chapter on Kan extensions the title "All concepts are Kan extensions". In this section, we provide the definition of Kan extensions and discuss limits and colimits as an example of Kan extensions.

2.1 Left and right Kan extensions

We first provide the definition Kan extensions of a functor along another functor, and explain the universal properties.

Definition 2.1. Let $F:\mathbb{X}\longrightarrow\mathbb{D}$ and $K:\mathbb{X}\longrightarrow\mathbb{Y}$ be any two functors.

(i) **Right Kan (minimal) extension** of F along K is a functor $\underline{F}_K:\mathbb{Y}\longrightarrow\mathbb{D}$ with a natural transformation $\underline{\psi}:K\underline{F}_K\Rightarrow F$ which is universal , see Fig. 2-(a). The right Kan extension is written as $(\underline{F}_K,\underline{\psi})$.

(ii) **Left Kan (maximal) extension** of F along K is a functor $\overline{F}_K:\mathbb{Y}\longrightarrow\mathbb{D}$ with a natural transformation $\overline{\psi}:F\Rightarrow K\overline{F}_K$ which is couniversal, see Fig. 2-(b). The left Kan extension is written as $(\overline{F}_K,\overline{\psi})$.

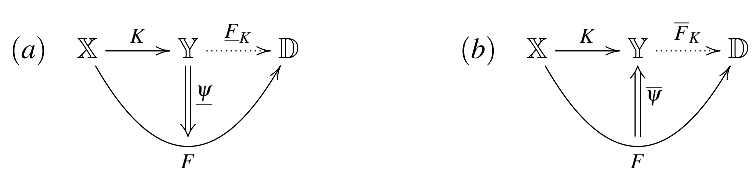


Figure 1: (a) Right Kan Extension (b) Left Kan Extension

Fig. 1 shows the Kan extensions of F along K. We refer to the category $\mathbb{D}$ as the **target**, the category $\mathbb{X}$ as the **source categories**. Functor F is extended from its source $\mathbb{X}$ along K. Right and Left Kan extensions of F along K are usually written as $\mathrm{Ran}_K(F)$ and $\mathrm{Lan}_K(F)$. However, we use the notation introduced in [17] for resource monotone extensions for uniformity.

Let us examine the universal properties of the Kan extensions. The universal property of right Kan extension assures that for any other functor $H : \mathbb{Y} \longrightarrow \mathbb{D}$ with a natural transformation $\gamma : KH \Rightarrow F$, there exists a $\gamma' : H \longrightarrow \underline{F}_K$ such that γ factors through $\underline{\psi}$ via γ', that is, $\gamma = (K \otimes \gamma')\underline{\psi}$ (See Fig. 2-(a)). Informally, this means the right Kan extension of F along K is the most conservative extension and that any other extension H can be transformed to $\underline{F}_K$. In this sense, $\underline{F}_K$ is the minimal extension of F along K.

Similarly the couniversal property of left Kan extension assures that for any other functor $H : \mathbb{Y} \longrightarrow \mathbb{D}$ with a natural transformation $\delta : F \Rightarrow KH$, there exists a $\delta' : \overline{F}_K \longrightarrow H$ such that δ factors through $\overline{\psi}$ via δ', that is, $\gamma = \overline{\psi}(K \otimes \gamma')$ (See Fig. 2-(b)). Informally, this means that $\overline{F}_K$ can be naturally transformed to any other such H. In this sense, $\overline{F}_K$ is the maximal extension of F along K.

The universal properties of Kan extensions assure that the extensions are optimal.

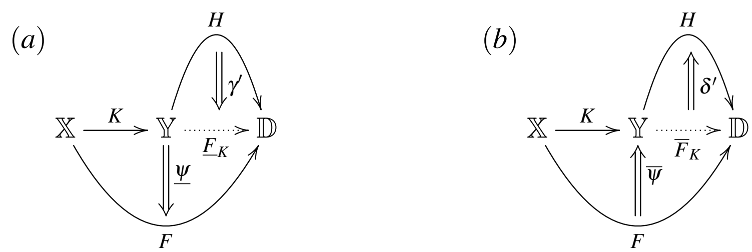
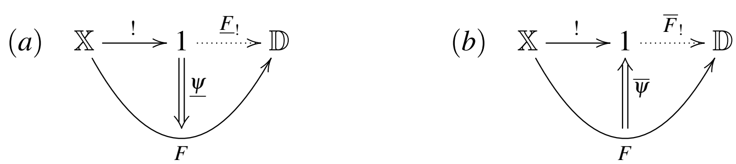


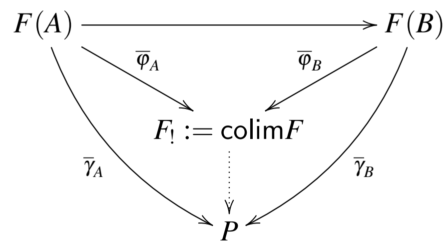
Figure 2: (a) Right Kan Extension is universal (b) Left Kan Extension is couniversal

Example 2.2. The left and right Kan extensions of a functor F along a terminal functor $(!)$ gives precisely the limit and the colimit of F. The terminal functor maps all the objects and the maps of the domain category to the single object and the single map in the terminal category (1) respectively. Any functor proceeding from the terminal category chooses precisely one object and its identity morphism in the codomain category.



The **left Kan extension** of $F : \mathbb{X} \longrightarrow 1$ along the unique functor into 1 gives a **colimiting cocone**. The functor $\underline{F}_!$ chooses precisely one object in $\mathbb{D}$ (hence we write the object as $\underline{F}_!$) which is the apex of the cocone. The natural transformation $\overline{\psi}$ has components, $\overline{\psi}_X : !F(X) \Rightarrow \underline{F}_!$ for each $X \in \mathbb{X}$.

Due to the couniversal property of $\underline{\psi}$, for any other functor $P : 1 \longrightarrow \mathbb{D}$ with a natural transformation $\gamma : !P \Rightarrow F$, there exists a unique natural transformation $\gamma' : P \Rightarrow !\underline{F}_!$ such that $\gamma'\underline{\psi} = \gamma$. Hence, $\underline{F}_!$ is **the limit of diagram** F.



Suppose $\mathbb{D}$ is the poset $(\mathbb{R}, \leq)$, then in the above diagram, $\underline{F}_!$ is precisely the **greatest lower bound** of $\{F(A), F(B)\}$.

Similarly, **the right Kan extension** gives a **limiting cone**. $\overline{F}_!$ is referred to as the **limit of diagram** F. When $\mathbb{D}$ is a poset $\overline{F}_!$ is the **least upper bound** of the subset of $\mathbb{R}$ chosen by F.

2.2 How to compute Kan extensions?

In Example 2.2, it was shown that the left and right Kan extensions of a functor F along the unique functor into the terminal category are respectively the colimit and the limit of diagram F. In this section, we show how one can compute Kan extensions of a functor when the target category is complete (has all small limits) and cocomplete (has all small colimits) and the intermediate category is locally small (the arrows between any two objects in the category is a small set).

Theorem 2.3. *[30, Thoerem 6.2.1] Given functors $F : \mathbb{X} \longrightarrow \mathbb{D}$ and $K : \mathbb{X} \longrightarrow \mathbb{Y}$, if the category $\mathbb{D}$ is cocomplete, then the left Kan extension $\overline{F}_K$ exists and is defined to be:*

$$\forall Y \in \mathbb{Y}, \ \ \overline{F}_K(Y) := \mathsf{colim}(K \downarrow Y \xrightarrow{\pi_{K \downarrow Y}} \mathbb{X} \xrightarrow{F} \mathbb{D}) \tag{2.1}$$

with the natural transformation $\overline{\psi}$ extracted from colimiting cocones in $\mathbb{D}$.

If $\mathbb{C}$ is complete, then the right Kan extension $\underline{F}_K$ exists and is defined to be:

$$\forall Y \in \mathbb{Y}, \ \ \underline{F}_K(Y) := \mathsf{lim}(Y \downarrow K \xrightarrow{\pi_{Y \downarrow K}} \mathbb{X} \xrightarrow{F} \mathbb{D}) \tag{2.2}$$

with the natural transformation $\underline{\psi}$ extracted from limiting cones in $\mathbb{D}$.

Proof. (Sketch)

Suppose $F : \mathbb{X} \longrightarrow \mathbb{D}$ is any functor and $\mathbb{D}$ is cocomplete. Then, one can compute the left Kan extension $(\underline{F}_K, \overline{\psi})$ of F along any functor $K : \mathbb{X} \longrightarrow \mathbb{Y}$ as follows:

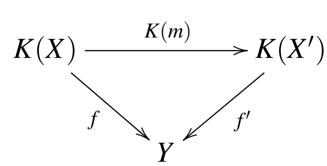
Defining functor $\underline{F}_K : \mathbb{Y} \longrightarrow \mathbb{D}$:

The left Kan extension is computed on each point (object) in $\mathbb{Y}$.

For each object Y in $\mathbb{Y}$, consider the slice category $(K \downarrow Y)$. The objects in the slice category are pairs (X, f) where,

$$f : K(X) \longrightarrow Y \in \mathbb{Y}$$

and a map $m : (X, f) \longrightarrow (X, f')$ in the slice category is a map $m \in \mathbb{X}$ such that the following triangle commutes:



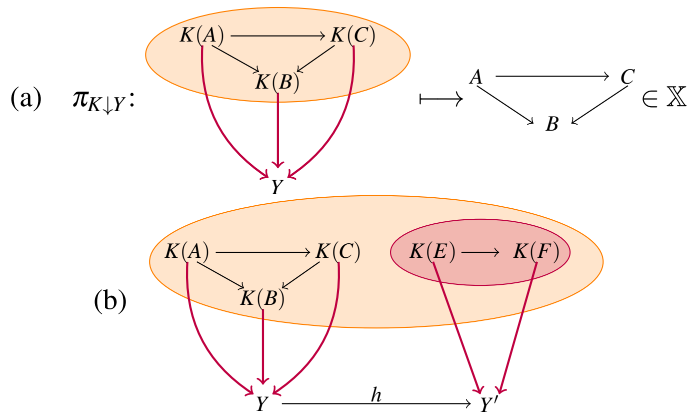


Figure 3: (a) $\pi_{K\downarrow Y}$ projects the shaded region of $(K \downarrow Y)$ into $\mathbb{X}$ (in general $K \downarrow Y$ is not a subcategory of $\mathbb{X}$); (b) An arrow $h : Y \to Y' \in \mathbb{Y}$ leads to the (shaded) base of above Y' to be included in the (shaded) base above Y. Hence the base of the colimiting cocone of $\pi_{K\downarrow Y}F$ includes the shaded base of the colimiting cone of $\pi_{K\downarrow Y}F$ inducing a unique map $\mathsf{colim}(\pi_{K\downarrow Y}F) \to \mathsf{colim}(\pi_{K\downarrow Y'}F)$

Stated informally, the slice category contains complete information on how to arrive at an object $Y \in \mathbb{Y}$ using objects and transformations of $\mathbb{X}$. The projection functor $\pi_{K\downarrow Y}$ chooses precisely the subcategory of $\mathbb{X}$ relevant to Y, see Figure 3-(a). The left Kan extension on point Y is the colimit of the diagram F applied to this sub-category. The couniversal cocone of the diagram $\pi_{K\downarrow Y}F$ has a natural transformation $\lambda : \mathsf{Lim}(\pi_{K\downarrow Y}F) \Rightarrow F$, with a component λ_X for each object $\pi_{K\downarrow Y}(f, K(X)) := X \in \mathbb{X}$.

The left extension $\overline{F}_K$ is then defined as follows:

- For all objects $Y \in \mathbb{Y}$, $\underline{F}(Y) := \mathsf{colim}(\pi_{K\downarrow Y}F)$.
- For all maps $h : Y \longrightarrow Y'$, $\mathsf{colim}(\pi_{K\downarrow Y}F) \longrightarrow \mathsf{colim}(\pi_{K\downarrow Y'}F)$ is the unique arrow induced by h, see Figure 3-(b).

Defining the natural transformation $\overline{\psi} : F \Rightarrow K\overline{F}_K$:

For all $X \in \mathbb{X}$, $\overline{\psi}_X$ is the component $\mathsf{lim}(\pi_{K\downarrow KX}F) \longrightarrow F(X)$ of the colimiting cocone corresponding to the initial object $(1_K X, KX) \in (K \downarrow KX)$.

Computing right Kan extension is dual to computing left Kan extensions. If $F : \mathbb{X} \longrightarrow \mathbb{D}$ is any functor and $\mathbb{D}$ is complete (contains all small limits), then one can compute the right Kan extension $\underline{F}_K, \underline{\psi})$ of F along any functor $K : \mathbb{X} \longrightarrow \mathbb{Y}$ by replacing the slice construction by the coslice construction, and colimits by limits in the above procedure. □

Corollary 2.4. If $(\underline{F}_K, \underline{\psi})$ is the right Kan extension of a functor F along any **full and faithful** functor K, then the natural transformation $\underline{\psi}$ is an isomorphism.

Similarly, if $(\overline{F}_K, \overline{\psi})$ is the left Kan extension of a functor F along any **full and faithful** functor K, then the natural transformation $\overline{\psi}$ is an isomorphism.

Proof. Note that for all $X \in \mathbb{X}$, $K\underline{F}(X) = \underline{F}(K(X)) := \mathsf{lim}(\pi_{K(X)\downarrow K})$

Since K is full and faithful, every $K(f) : K(X) \longrightarrow K(X') \in \mathbb{Y}$ corresponds to a unique $f : X \longrightarrow X' \in \mathbb{X}$. Then for all $X \in \mathbb{X}$, $(KX, 1_{KX})$ is an initial object in the coslice category $(KX \downarrow K)$. Thereby, the diagram $\pi_{KX\downarrow K}$ contains all the maps radiating from X. Hence, $\mathsf{lim}(\pi_{KX\downarrow K}F) = F(X)$, thereby, $\underline{\psi}$ is an isomorphism.

The argument for the left Kan extension is dual to the above proof. □

We use this procedure to compute extensions of resource monotones which are functors into a posetal category (a poset considered as a category), see Section 3.4.

3 Kan extensions of Resource Measures

3.1 Resource Theories as partitioned Categories

We introduce partitioned Categories as a framework for resource theories, and functors for partitioned Categories to describe relationships between resource theories.

Definition 3.1. A **partitioned category (pCat)** $(\mathbb{X}, \mathbb{X}_{\mathsf{f}})$ consists of a category $\mathbb{X}$ and a chosen subcategory $\mathbb{X}_{\mathsf{f}}$ of free transformations with the inclusion being bijective on objects.

The objects of the category are interpreted as **resources** and the maps to be **resource transformations**. The subcategory includes all objects and those transformations which are designated to be **free**.

The following are a few examples of resource theories as pCats:

Randomness

Cryptographic protocols use randomness as an essential resource for establishing secure communication of devices by generating random keys. The degree of randomness determines how secure the communication channel is. Randomness is also used in computer algorithms to solve certain problems. In other words, randomness is an essential computational resource of practical use. Entropy is used as measure of randomness: in particular, Shannon entropy quantifies randomness in that it expresses the average surprisal on the outcome of a random experiment. Entropy has been studied in the context of randomness using the category $\mathsf{FinProb}$ (renamed below as Detmn) [1] and [10, Example 2.5]. The following is a resource theory of randomness:

Example 3.2. $(\mathsf{Rand}, \mathsf{Detmn})$:

(Detmn is the chosen sub-category of free transformations in Rand)

Resources: (X, p) where X is a finite set and p is a probability distribution over X.

X can be interpreted of as a set of possible states of a system and p be the probability distribution over the states.

Resource Transformations: $M : (X, p) \longrightarrow (Y, q)$ is a real $|X| \times |Y|$ row stochastic matrix (rows sum to 1) such that $pM = q$.

A resource transformation $M : (X, p) \longrightarrow (Y, q)$ is row stochastic if and only if for all $x \in X$, M_x is a probability distribution: suppose the system is in state x, then the stochastic process produces states $y \in Y$ with probability M_{xy}.The requirement that $pM = q$ means that under the stochastic process M, the probability of Y being in state y after process M on X is given by $\sum_{x \in X} M_{xy} p_x$.

Identity transformations: Identity matrices

Composition: Suppose $(X, p) \xrightarrow{M} (Y, q) \xrightarrow{N} (Z, s)$, then $(X, p) \xrightarrow{MN} (Z, s)$ is defined as the matrix multiplication

Free transformations: A resource transformation $(X, p) \xrightarrow{M} (Y, q)$ is free if it is deterministic, that is, M is simply a function $X \longrightarrow Y$. Hence, for all $x \in X, y \in Y$, $M_{xy} \in \{0, 1\}$

Non-uniformity

Pure states represent states on which the experimenter has maximum information. These conditions are often very hard to achieve in concrete settings due to the presence of external noise. In such cases, the state is called mixed, and can be expressed as a convex combination of pure states. From this perspective, it is clear that pure states represent the maximal resource and the closer a state is to a pure state, the more resourceful it is. Therefore, the least resourceful state of any system is the *maximally mixed state*, which can be expressed as a uniform probability distribution over the states of the system. [15]

Example 3.3. $(\mathsf{Rand}, \mathsf{Uniform})$:

Resources, transformations, identity and composition: Same as example 3.2

Free transformation: A map $(X,p) \xrightarrow{U} (Y,q)$ is free if U is a uniform matrix. A row stochastic matrix $(X,p) \xrightarrow{M} (Y,q)$ is uniform if for all $y \in Y$,

$$\sum_{x \in X} M_{xy} = 1$$

The columns of M sum to $|X|/|Y|$. When U is a square matrix, it is doubly stochastic.

Note that, a uniform probability distribution $u := (1/n, 1/n, 1/n, \cdots, 1/n)$ is simply the uniform matrix $(\{*\}, (1)) \longrightarrow (X, u)$, which is u itself.

$(\mathsf{Rand}, \mathsf{Uniform})$ consists of classical probabilistic states. A non-uniformity theory based on quantum states is as follows:

Example 3.4. $(\mathsf{qRand}, \mathsf{qUniform})$

Resources: (ρ, H) where $\rho : H \longrightarrow H \in L(H)$ is a quantum state, also known as density matrix (a positive semi-definite operator with trace 1), and H is a finite-dimensional Hilbert space.

Resource transformations: $(\rho, H) \xrightarrow{\mathscr{E}} (\sigma, K)$ is a quantum channel $\mathscr{E} : L(H) \longrightarrow L(K)$ such that $\mathscr{E}(\rho) = \sigma$

Composition and Identity transformations: Usual composition of quantum channels and identity channels

Free transformations: Unital quantum channels i.e., $\mathscr{E} : L(H) \longrightarrow L(K)$ such that $\mathscr{E}\left(\frac{1}{\mathsf{dim}(H)} 1_H\right) = \frac{1}{\mathsf{dim}(K)} 1_K$, where $\mathbb{1}_H \in L(H)$ is the identity matrix. In other words, unital channels preserve maximally mixed states.

Entanglement

Entanglement is one of the most important quantum resources, and it is used in several communication scenarios, such as quantum teleportation [3] or dense coding [4]. It is known that local operations and classical communication (LOCC) cannot increase the entanglement of a quantum state [20]. Hence, when entanglement is considered to be a resource, LOCC operations are precisely the free transformations of this resource theory. The basic setting in which entanglement is studied involves quantum states over two systems, which is referred to as "bipartite entanglement".

A resource theory of bipartite entanglement is constructed as follows. The following resource theory is obtained by applying the coslice (state) construction on [10, Example 3.7]:

Example 3.5. $(\mathsf{Bip}, \mathsf{LOCC})$:

Resources: $\rho \in L(H \otimes K)$ is a quantum state which is a positive semi-definite operator with trace 1, and H, K are finite-dimensional Hilbert spaces.

Resource transformations: $\rho \xrightarrow{\mathscr{E}} \sigma$ is a quantum channel (completely positive trace preserving map) such that $\mathscr{E}(\rho) = \sigma$

Free transformations: Local operations and classical communication

The composition is the usual composition of identity channels.

Distinguishability

In some situations it is important to consider pairs of quantum states and evaluate how different they are from each other. To this end, various quantifiers have been defined, such as the trace distance, the fidelity [38] or quantum divergences [17, 14]. These quantifiers all show that, whenever the same channel is applied to each element of a pair of quantum states, in general our ability to distinguish the resulting states is decreased. This suggests setting up a resource theory of the distinguishability, also known as quantum relative majorization [29, 7].

A resource theory of quantum distinguishability is given as follows [17, 14].

Example 3.6. (Distinguish, Processing):

Resources : $((\rho,\sigma),H)$ are pairs of quantum states, that is, $\rho,\sigma \in L(H)$ where H is a finite-dimensional Hilbert space.

Resource transformations : $(\mathscr{E}_1,\mathscr{E}_2) : ((\rho_H,\sigma_H),H) \longrightarrow ((\rho_K,\sigma_K),K)$ are pairs of quantum channels $\mathscr{E}_1,\mathscr{E}_2 : L(H) \longrightarrow L(K)$ such that $\mathscr{E}_1(\rho_H) = \rho_K$ and $\mathscr{E}_2(\sigma_H) = \sigma_K$

Composition and identity transformations : $(\mathscr{E}_1,\mathscr{E}_2)(\mathscr{E}_3,\mathscr{E}_4) := (\mathscr{E}_1\mathscr{E}_3,\mathscr{E}_2\mathscr{E}_4)$ and identity transformations are given by pairs of identity channels

Free transformations: $(\mathscr{E}_1,\mathscr{E}_2)$ such that $\mathscr{E}_1 = \mathscr{E}_2$

3.2 Relationships between Resource Theories as pCat functors

Now that we have formalized resource theories as pCats, we can formalize the relationship between resource theories as functors of pCats. For example, classical theories of the corresponding quantum resource theories. Physical theories defined on pure states are considered as subtheories of corresponding mixed state theories. Such relationships can be formalized as functors of pCats.

Definition 3.7. A **functor of partitioned categories (pCat)**, $F : (\mathbb{X},\mathbb{X}_\mathsf{f}) \longrightarrow (\mathbb{Y},\mathbb{Y}_\mathsf{f})$, is a functor $F : \mathbb{X} \longrightarrow \mathbb{Y}$ such that if $f \in \mathbb{X}_\mathsf{f}$ then $F(f) \in \mathbb{Y}_\mathsf{f}$ i.e., the functor preserves free transformations.

$F : (\mathbb{X},\mathbb{X}_\mathsf{f})$ being a functor means that it preserves the identity transformations: $F(1_A) = 1_{F(A)}$, and it preserves the composition in $\mathbb{X}$: $F(fg) = F(f)F(g)$.

Figure 4 is a schematic of a pCat functor. The triangles represent non-free transformations, and the hollow circles represent free transformations. As one can see, a pCat functor may or may not preserve a non-free transformation.

Definition 3.8. A pCat functor $F : (\mathbb{X},\mathbb{X}_\mathsf{f}) \longrightarrow (\mathbb{Y},\mathbb{Y}_\mathsf{f})$ is **full** if $F : \mathbb{X} \longrightarrow \mathbb{Y}$ is full, and F is **faithful** if $F : \mathbb{X} \longrightarrow \mathbb{Y}$ is faithful.

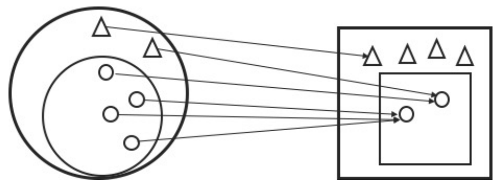


Figure 4: Schematic for functor of pCats

Let us look at a few examples of full and faithful pCat functors. For a quantum system, pure states are considered to be a subset of mixed states since mixed states are convex combination of pure states. The resource theory of pure states bipartite entanglement embeds in the general theory of bipartitle entanglement through an inclusion functor, see Example 3.9. Lemma 3.10 proves that this inclusion is a pCat functor.

Example 3.9. The resource theory of bipartite pure-state entanglement, $(\mathsf{PureBip}, \mathsf{LOCC_p})$ has pure quantum states ρ, where $\rho \in L(H \otimes K)$, as resources. A resource transformation is a quantum channel $\mathscr{E} : \rho \longrightarrow \sigma$ such that $\mathscr{E}(\rho) = \sigma$ where $\sigma \in L(H' \otimes K')$. The free transformations are LOCC operations between pure bipartite states, here denoted as $\mathsf{LOCC_p}$.

Lemma 3.10. *The inclusion* $i : \mathsf{PureBip} \hookrightarrow \mathsf{Bip}$ *defined to be identity on objects and maps is a full and faithful pCat functor* $i : (\mathsf{PureBip}, \mathsf{LOCC_p}) \hookrightarrow (\mathsf{Bip}, \mathsf{LOCC})$.

Proof. The inclusion is a pCat functor since $\mathsf{LOCC_p} \hookrightarrow \mathsf{LOCC}$. Moreover, $i : \mathsf{PureBip} \hookrightarrow \mathsf{Bip}$ is full and faithful inclusion. □

Classical theories are considered as sub-theories of quantum theories. This gives an inclusion functor classical distinguishability into quantum distinguishability. The following is the resource theory for classical distinguishability and is referred to as classical relative majorization in [31, 32, 33, 29, 7]:

Example 3.11. In the resource theory of classical distinguishability, $(\mathsf{cDistinguish}, \mathsf{cProcessing})$, a resource $((p,q),X)$ is a pair of probability distributions $p := (p_1, \cdots, p_{|X|})$ and $q := (q_1, \cdots, q_{|X|})$ over a finite set X. Resource transformations $(M, M') : ((p,q),X) \longrightarrow ((p',q'),Y)$ where M and M' are pairs of row stochastic matrices such that $pM = p'$ and $qM' = q'$. Free transformations are (M, M') such that $M = M'$.

Example 3.12. The inclusion $i : \mathsf{cDistinguish} \hookrightarrow \mathsf{Distinguish}$ is defined as follows:

- For all resources $((p,q),X) \in \mathsf{cDistinguish}$, $i((p,q),X) := ((\rho^p, \rho^q), \mathbb{C}^{|X|})$ where $[\rho^p]_{ij} = \delta_{ij} p_i$, $1 \le i \le |X|, 1 \le j \le |X|$. ρ^p and ρ^q are diagonal density matrices with the probability distributions p and q as their diagonals respectively.
- Given a transformation $(M, M') : ((p,q),X) \longrightarrow ((p',q'),Y)$, $i((M,M')) := (\mathscr{E}, \mathscr{E}')$ where $\mathscr{E}$ and $\mathscr{E}'$ are determined by M and M' respectively as follows.

  For any quantum state (positive semi-definite operator of trace 1 on a Hilbert Space H),

$$\mathscr{E}(\rho) = \sum_{i,j} B_{ij} \rho B_{ij}^{\dagger} \tag{3.1}$$

  where $B_{ij} = \sqrt{M_{ij}} |j\rangle\langle i|$ where $1 \le i, j \le |X|$ and $B_{ij}^{\dagger}$ is its adjoint (cf. [38]). ($|j\rangle$ is a column vector with 1 at position j and zero elsewhere.)

Lemma 3.13. *The inclusion* $i : \mathsf{cDistinguish} \hookrightarrow \mathsf{Distinguish}$ *defined in Example 3.12 is full and faithful (or fully faithful) pCat functor* $i : (\mathsf{cDistinguish}, \mathsf{cProcessing}) \hookrightarrow (\mathsf{Distinguish}, \mathsf{Processing})$.

Example 3.14. Closely, related to Example 3.12, is the inclusion of Rand into qRand. The inclusion $i : \mathsf{Rand} \hookrightarrow \mathsf{qRand}$ is defined as follows: for all $(p,X) \in \mathsf{Rand}$, $i((p,X)) := (\rho^p, \mathbb{C}^{|X|})$, and for row stochastic matrices $M \in \mathsf{Rand}$, $i(M)$ is defined as in eqn (3.1).

Lemma 3.15. *The inclusion* $i : \mathsf{Rand} \hookrightarrow \mathsf{qRand}$ *defined in Example 3.14 is a full and faithful pCat functor* $i : (\mathsf{Rand}, \mathsf{Uniform}) \hookrightarrow (\mathsf{qRand}, \mathsf{qUniform})$.

3.3 Preorder collapse and monotones

One of the major goals of resource theories is to the identify necessary and sufficient conditions for the existence of a free transformation between two resources. Once such conditions are identified, one can choose to 'forget' the different possible ways in which resource A can be converted to resource B freely, and only 'remember' if there exists a free transformation from A to B.

The necessary and sufficient conditions for the existence of a free transformation between pairs of resources define an equivalence class on the resource theory (freely inter-convertible resources are considered to be equivalent) and a preorder on the equivalence classes. Such necessary and sufficient conditions can be encoded into a pCat functor from the resource theory. On applying this functor, the resource theory collapses into a preorder:

Definition 3.16. Given a resource theory $(\mathbb{X}, \mathbb{X}_\mathsf{f})$ and a preorder $(\mathsf{ob}(\mathbb{X}), \mathsf{order})$ where $\mathsf{ob}(\mathbb{X})$ refers to the set of objects of $\mathbb{X}$, a **preorder collapse** of the resource theory $(\mathbb{X}, \mathbb{X}_\mathsf{f})$ is a pCat functor $(\mathbb{X}, \mathbb{X}_\mathsf{f}) \longrightarrow (\mathsf{chaos}_\mathbb{X}, \mathsf{order}_\mathbb{X})$, where $\mathsf{chaos}_\mathbb{X}$ is the indiscrete (chaotic) category with the same objects as $\mathbb{X}$, and for any two objects $A, B \in \mathsf{chaos}_\mathbb{X}$, the transformation $A \longrightarrow B \in \mathsf{order}_\mathbb{X}$ if "A order B" is true.

Let us look at an example of a preorder collapse of $(\mathsf{Rand}^{\mathsf{op}}, \mathsf{Uniform})$ determined by the majorization relation [27]. Suppose $p := (p_1, p_2, p_3, \cdots, p_n)^\uparrow$ and $q := (q_1, q_2, q_3, \cdots, q_m)^\uparrow$ such that the elements of the distribution are in increasing order. We say q is majorized by p written as $q \preceq p$ if the Lorenz curve [27, 24] of p lies either completely below the Lorenz curve of q (see Figure 5) or coincides with it. This means that q is more uniform than p.

Lorenz curve [24, 27, 15] $L(p)$ for a probability distribution $p := (p_1, p_2, \cdots, p_n)$ is characterized as the linear interpolation of points $(i/n, \sum_{k=1}^{i} p_k)$, where $i = 0, 1, \cdots, n$; see Figure 5.

Example 3.17. Define $P : \mathsf{Rand} \longrightarrow \mathsf{chaos}_\mathsf{Rand}$ as follows: for each probability distribution $p \in \mathsf{Rand}$, $P(p) := p$; each $M : p \longrightarrow q$ is mapped to the unique arrow $p \longrightarrow q$.

The functors into chaotic categories are determined by the objects. It is straightforward that P as defined above is a functor. The below theorem establishes that $P : (\mathsf{Rand}^{\mathsf{op}}, \mathsf{Uniform}) \to (\mathsf{chaos}_\mathsf{Rand}, \preceq_\mathsf{Rand})$ is pCat functor.

Theorem 3.18. *[18] Given two finite probability distributions p and q,* $q \preceq p$ *if and only if there exists a uniform matrix* $U : p \longrightarrow q$ *such that* $pU = q$.

Corollary 3.19. The map $P : (\mathsf{Rand}^{\mathsf{op}}, \mathsf{Uniform}) \to (\mathsf{chaos}_\mathsf{Rand}, \preceq)$ is a preorder collapse.

Proof. By Theorem 3.18, if $U : p \longrightarrow q \in \mathsf{Uniform}$, then $P(q) \preceq P(p)$. □

Given a resource theory, one can assign a real value to each resource such that the assignment respects the preorder defined by the theory. In this respect, a monotone is a function $f : R \longrightarrow [0, \infty]$ where R is a set of resources which preserves the preorder on the equivalence class of resources. To represent a monotone as a pCat functor, the poset $([0, \infty], \leq)$ is defined as a pCat as follows:-

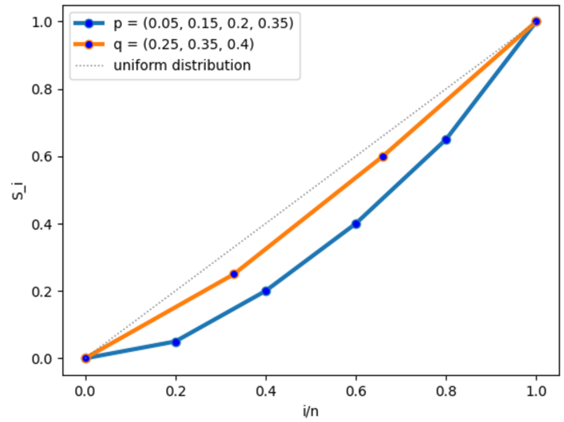


Figure 5: The Lorenz curve of q is majorized by the Lorenz curve of p

Definition 3.20. The poset $([0,\infty],\leq)$ is encoded as the pCat $(\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]})$ where $\mathsf{chaos}_{[0,\infty]}$ is the chaotic category with objects as $r\in[0,\infty]$ and the free transformations are those maps respecting the $\leq$ order ($m\longrightarrow n\in\leq_{[0,\infty]}$ if and only if $m\leq n$).

Definition 3.21. A **monotone** for a resource theory $(\mathbb{X},\mathbb{X}_{\mathsf{f}})$ is a pCat functor

$$F:(\mathbb{X},\mathbb{X}_{\mathsf{f}})\longrightarrow\left(\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]}\right).$$

An **op-monotone** is a contravariant monotone, that is,

$$F:(\mathbb{X},\mathbb{X}_{\mathsf{f}})\longrightarrow\left(\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]}\right)^{\mathsf{op}}:=\left(\mathsf{chaos}^{\mathsf{op}}_{[0,\infty]},\geq_{[0,\infty]}\right).$$

Even though op-monotones are more frequently used in resource theories, we defined the codomain of a monotone to be $(\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]})$ because, in general, an arrow $a\longrightarrow b$ in a posetal category refers to $a\leq b$, and such ordering becomes relevant when one computes inf and sup of a subset in the poset, see Section 3.4. Let us look at a few examples of monotones:

In information theory, Shannon entropy is a well-known measure of randomness or uncertainty in the outcome when a random experiment (experiment with multiple outcomes) is repeated one or more times. The value of Shannon entropy lies in [0,1] where 0 represents absolute certainty and 1 represents maximum uncertainty. In the following example, we construct a monotone for the resource theory of randomness, $(\mathsf{Rand},\mathsf{Detmn})$, and an op-monotone $(\mathsf{Rand},\mathsf{Uniform})$ based on the Shannon entropy:

Example 3.22. Define $\mathsf{Shannon}:\mathsf{Rand}\longrightarrow\mathsf{chaos}_{[0,\infty]}$ as follows:

- For all finite probability distributions, $(X,p)\in\mathsf{Rand}$, $\mathsf{Shannon}(p):=H(p)$ where $H(p)$ is the Shannon entropy of p:
$$H(p):=-\sum_{1\leq i\leq|X|}p_i\,\mathsf{log}\,p_i$$
- For all $(X,p)\xrightarrow{\mathsf{Shannon}}(X,q)\in\mathsf{Rand}$, then $F(M)$ is the unique arrow $H(p)\longrightarrow H(q)$

It is straightforward that $\mathsf{Shannon}:\mathsf{Rand}\longrightarrow\mathsf{chaos}_{[0,\infty]}$ is a functor. We note that the functor Shannon acts as a monotone for $(\mathsf{Rand},\mathsf{Detmn})$ and as an op-monotone for $(\mathsf{Rand},\mathsf{Uniform})$.

Lemma 3.23. *[11] Suppose* $\mathsf{Shannon}:(X,p)\longrightarrow(Y,q)\in\mathsf{Detmn}$*, then* $H(p)\geq H(q)$.

Corollary 3.24. The map Shannon : $(\mathsf{Rand},\mathsf{Detmn}) \to ([0,\infty],\geq)$ defined as in Example 3.22 is an op-monotone.

Lemma 3.25. *[15, 27] Suppose* Shannon : $(X,p) \longrightarrow (Y,q) \in$ Uniform, *then* $H(p) \leq H(q)$.

Corollary 3.26. The map Shannon : $(\mathsf{Rand},\mathsf{Uniform}) \to ([0,\infty],\leq)$ defined as in Example 3.22 is a monotone.

Next we describe a monotone for the resource theory, (cDistinguish, cProcessing):

Example 3.27. [17, 14, Definition 2] A **classical divergence** D : cDistinguish $\longrightarrow \mathsf{chaos}_{[0,\infty]}$ is any functor, that for any resource $((p,q),X) \in$ cDistinguish and a resource transformation $(M,M) \in$ cDistinguish, satisfies the data processing inequality:

$$D(p,q) \geq D(pM,qM)$$

Lemma 3.28. *Any classical divergence* D : (cDistinguish, cProcessing $\longrightarrow (\mathsf{chaos}_{[0,\infty]},\geq_{[0,\infty]})$ *is an op-monotone.*

The following is a monotone for PureBip as follows [28, 38]:

Example 3.29. Define Schmidt : PureBip $\to \mathsf{chaos}_{[0,\infty]}$ to be the following: for all resources $\rho^{HK} \in$ PureBip, where $\rho^{HK} \in L(H\otimes K)$ is a pure quantum state,

$$N\left(\rho^{HK}\right) := \mathrm{Rank}\left(\rho^{H}\right) \text{ where, } \rho^{H} := \mathrm{Tr}_K\left(\rho^{HK}\right)$$

Lemma 3.30. *[28, 38]* $N : (\mathsf{PureBip},\mathsf{LOCC_p}) \to (\mathsf{chaos}_{[0,\infty]},\geq_{[0,\infty]})$ *is an op-monotone.*

3.4 Kan Extensions of monotones

Now, we set up resource theories to apply Kan extensions for extending monotones from one resource theory to another when there exists a pCat functor between them.

Given a monotone $M : (\mathbb{X},\mathbb{X}_\mathsf{f}) \longrightarrow (\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]})$ and a pCat functor $K : (\mathbb{X},\mathbb{X}_\mathsf{f}) \longrightarrow (\mathbb{Y},\mathbb{Y}_\mathsf{f})$, one could desire to extend M to obtain monotones on $(\mathbb{Y},\mathbb{Y}_\mathsf{f})$. Observe that a monotone $M : (\mathbb{X},\mathbb{X}_\mathsf{f}) \longrightarrow \left(\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]}\right)$ is concerned only with the free transformations: if $f : A \longrightarrow B \in \mathbb{X}_\mathsf{f}$, then $M(A) \leq M(B)$; otherwise $M(A) \longrightarrow M(B)$ is the unique arrow signifying that there is no order between A and B. Since $\mathbb{X}_\mathsf{f}$ includes all the objects of $\mathbb{X}$, and the monotone M is concerned with only free transformations, in order to extend $M : (\mathbb{X},\mathbb{X}_\mathsf{f}) \longrightarrow \left(\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]}\right)$ it suffices to extend $M_\mathsf{f} : \mathbb{X}_\mathsf{f} \longrightarrow \leq_{[0,\infty]}$ which is defined as follows :-

$$M_\mathsf{f} : \mathbb{X}_\mathsf{f} \longrightarrow \leq_{[0,\infty]}; \quad A \xrightarrow{f} B \mapsto M(A) \leq M(B)$$

Definition 3.31. Let $(\mathbb{X},\mathbb{X}_\mathsf{f}) \xrightarrow{K} (\mathbb{Y},\mathbb{Y}_\mathsf{f})$ be a pCat functor. Let $M : (\mathbb{X},\mathbb{X}_\mathsf{f}) \to (\mathsf{chaos}_{[0,\infty]},\leq_{[0,\infty]})$ be a monotone for the resource theory $(\mathbb{X},\mathbb{X}_\mathsf{f})$.

(a) The **minimal extension**[1] $\underline{M}_K : \mathbb{Y}_\mathsf{f} \longrightarrow \leq_{[0,\infty]}$ of M along K is the right Kan extension of the functor $M_\mathsf{f} : \mathbb{X}_\mathsf{f} \longrightarrow \leq_{[0,\infty]}$ along the functor $K_\mathsf{f} : \mathbb{X}_\mathsf{f} \longrightarrow \mathbb{Y}_\mathsf{f}; K_\mathsf{f}(h) := K(h)$ (see Figure 6-(a)).

(b) The **maximal extension** $\overline{M}_K : \mathbb{Y}_\mathsf{f} \longrightarrow \leq_{[0,\infty]}$ of M along K is the left Kan extension of the functor $M_\mathsf{f} : \mathbb{X}_f \longrightarrow \leq_{[0,\infty]}$ along the functor $K_\mathsf{f} : \mathbb{X}_\mathsf{f} \longrightarrow \mathbb{Y}_\mathsf{f}; K_\mathsf{f}(h) := K(h)$ (See Figure 6-(b)).

Let us unpack the definition of minimal and maximal extensions of a monotone in Definition 3.31. Any category given by a poset with suprema and infima is both complete and cocomplete. Since $([0,\infty],\leq)$ is such a poset, by Theorem 2.3 one can compute the minimal extension ($\underline{M}_K$) and the maximal extension ($\overline{M}_K$) using equations (2.2) and (2.1) respectively:

[1] We follow the naming convention in [17] for monotone extensions.

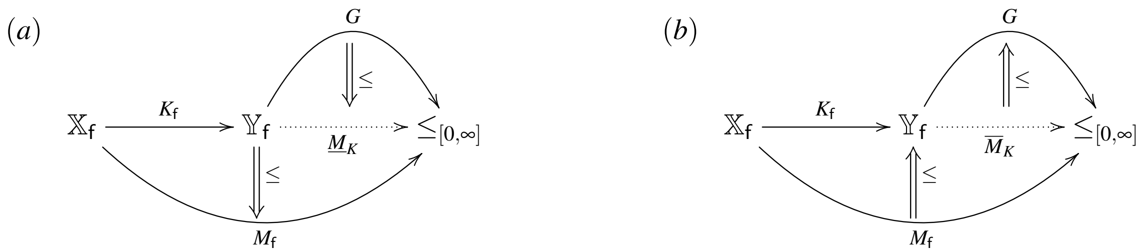


Figure 6: (a) Minimal (right Kan) extension (b) Maximal (left Kan) extension

Theorem 3.32. *(a) For all $Y \in \mathbb{Y}$, the minimal extension $\underline{M}_K(Y) : \mathbb{Y}_{\mathsf{f}} \longrightarrow \leq_{[0,\infty]}$ is given as:*

$$\underline{M}_K(Y) := \mathsf{lim}(\pi_{Y\downarrow K} M_{\mathsf{f}}) = \inf\{M(X) \mid Y \longrightarrow K(X) \in \mathbb{Y}_{\mathsf{f}}\} \tag{3.2}$$

(b)For all $Y \in \mathbb{Y}$, the maximal extension $\overline{M}_K(Y) : \mathbb{Y}_{\mathsf{f}} \longrightarrow \leq_{[0,\infty]}$ is given as:

$$\overline{M}_K(Y) := \mathsf{colim}(\pi_{K\downarrow Y} M_{\mathsf{f}}) = \sup\{M(X) \mid K(X) \longrightarrow Y \in \mathbb{Y}_{\mathsf{f}}\} \tag{3.3}$$

See Figure 7-(a) for a schematic of the minimal and maximal extensions of a monotone.

Usually for resource theories the codomain of the monotones is $([0,\infty]^{\mathsf{op}}, \leq^{\mathsf{op}}_{[0,\infty]}) = ([0,\infty]^{\mathsf{op}}, \geq_{[0,\infty]})$. In the computation of extensions of op-monotones, inf is flipped to sup in equation (3.2), and sup to be flipped to inf in equation (3.3):

Corollary 3.33. Suppose $M : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]})$ be a monotone and $K : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathbb{Y}, \mathbb{Y}_{\mathsf{f}})$. Then,

(a) For all $Y \in Y$, the minimal extension $\underline{M}_K(Y) : \mathbb{Y}_{\mathsf{f}} \longrightarrow \geq_{[0,\infty]}$ is given as:

$$\underline{M}_K(Y) := \mathsf{colim}(\pi_{Y\downarrow K} M_{\mathsf{f}}) = \sup\{M(X) \mid Y \longrightarrow K(X) \in \mathbb{Y}_{\mathsf{f}}\} \tag{3.4}$$

(b)For all $Y \in Y$, the maximal extension $\overline{M}_K(Y) : \mathbb{Y}_{\mathsf{f}} \longrightarrow \geq_{[0,\infty]}$ is given as:

$$\overline{M}_K(Y) := \mathsf{lim}(\pi_{K\downarrow Y} M_{\mathsf{f}}) = \inf\{M(X) \mid K(X) \longrightarrow Y \in \mathbb{Y}_{\mathsf{f}}\} \tag{3.5}$$

Proof. Note that $(\mathsf{chaos}_{[0,\infty}, \geq_{[0,\infty]}) = (\mathsf{chaos}^{\mathsf{op}}_{[0,\infty]}, \leq^{\mathsf{op}}_{[0,\infty]}) =: \left(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]}\right)^{\mathsf{op}}$. Hence, the limits in $\left(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]}\right)$ are the colimits in $\left(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]}\right)^{\mathsf{op}}$. □

Figure 7 - (b) and (c) visualizes the difference in computation of minimal extension of a (regular) monotone and an op-monotone.

Note that equation (3.4) is same as [17, Equation 2] and equation (3.5) is same as [17, Equation 3]. Let us have a closer look at equations (3.4) and (3.5). The minimum extension $\underline{M}_K$ assigns to any resource $Y \in \mathbb{Y}_{\mathsf{f}}$ the value of a resource $X \in \mathbb{X}_{\mathsf{f}}$ such that the value of X is lowest among the value of all those resources which can be transformed freely to Y under K $(KX \longrightarrow Y)$. If there exists no such $X \in \mathbb{X}_{\mathsf{f}}$ such that KX can be transformed to Y using a free transformation, then $\underline{M}_K(Y) = 0$ (colimit of the empty diagram is the initial object).

Similarly, the maximal extension $\overline{M}_K$ assigns to any resource $Y \in \mathbb{Y}_{\mathsf{f}}$ the value of a resource $X' \in \mathbb{X}_{\mathsf{f}}$ such that the value of X' is the highest among the value of all those resources which Y can be transformed to freely under K $(Y \longrightarrow KX)$. If there does not exist any such $X \in \mathbb{X}_{\mathsf{f}}$ which Y can be transformed to using a free transformation, then $\underline{M}_K(Y) = \infty$ (limit of the empty diagram is the terminal object).

We define what it means for the computed extensions to be optimal:

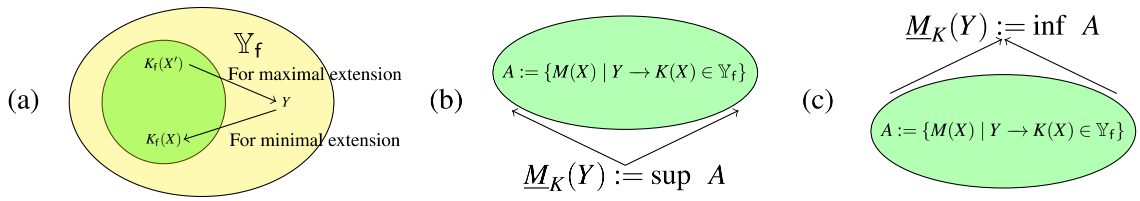


Figure 7: (a) Schematic of minimal and maximal extensions of any monotone along K; (b) Minimal extension of monotone M; (c) Minimal extension of an op-monotone M

Definition 3.34. The minimal extension of a monotone is **optimal** if for any other monotone $G:(\mathbb{Y},\mathbb{Y}_{\mathsf{f}}) \longrightarrow \left(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]}\right)$ such that for all $X \in \mathbb{X}$, $G(K(X)) \leq M(X)$, we have that

$$G(Y) \leq \underline{M}_K(Y)$$

The maximal extension of a monotone is **optimal** if for any other $G:(\mathbb{Y},\mathbb{Y}_{\mathsf{f}}) \longrightarrow \left(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]}\right)$ such that for all $X \in \mathbb{X}$, $M(X) \leq G(K(X))$, we have that

$$\overline{M}_K(Y) \leq G(Y)$$

For the extensions of an op-monotone to be optimal, " $\leq$ " is replaced by " $\geq$ " in definition 3.34.

Theorem 3.35. *Let $\underline{M}_K$ and $\overline{M}_K$ be minimal and maximal extensions of a monotone $M:(\mathbb{X},\mathbb{X}_{\mathsf{f}}) \longrightarrow \left(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]}\right)$ along a pCat functor $K:(\mathbb{X},\mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathbb{Y},\mathbb{Y}_{\mathsf{f}})$ as per definition 3.31. Then,*

(a) **Reduction:** *For all $X \in \mathbb{X}$,*
$$\underline{M}_K(K_{\mathsf{f}}(X)) \leq M(X) \leq \overline{M}_K(K_{\mathsf{f}}(X))$$

(b) **Monotonicity:** *For all $f:A \longrightarrow B \in \mathbb{Y}_{\mathsf{f}}$,*

$$\underline{M}_K(A) \leq \underline{M}_K(B) \text{ and } \overline{M}_K(A) \leq \overline{M}_K(B)$$

(c) **Optimality:** *$\underline{M}_K$ and $\overline{M}_K$ are optimal.*

Proof.

(a) Since $(\underline{M}_K, \leq)$ is the right Kan extension, for all $X \in \mathbb{X}$, $\underline{M}_K(K_{\mathsf{f}}(X)) \leq M_{\mathsf{f}}(X) = M(X)$ (see Figure 6-(a)).

Since $(\overline{M}_K, \leq)$ is the left Kan extension, for all $X \in \mathbb{X}$, $M(X) = M_{\mathsf{f}}(X) \leq \overline{M}_K(K_{\mathsf{f}}(X))$ (see Figure 6-(b)).

(b) Monotonicity follows from functoriality of $\underline{M}_K$ and $\overline{M}_K$

(c) The extensions are optimal by construction (see Figure 6 for the universal properties).

□

In the above lemma, Statement (a) tells us that the minimal and maximal extensions are respectively a lower and upper bound for M on $\mathbb{X}$. Statement (b) assures that the extensions are monotonic on free transformations. Statement (c) assures that the minimal and maximal extensions are respectively the greatest lower bound and the least upper bound for any other extension of M along K, hence are optimal.

Corollary 3.36. Let $\underline{M}_K$ and $\overline{M}_K$ be minimal and maximal extensions of an op-monotone $M : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]})$ along a pCat functor $K : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathbb{Y}, \mathbb{Y}_{\mathsf{f}})$ as per definition 3.31. Then, the following properties hold for the extensions:

(a) **Reduction:** For all $X \in \mathbb{X}$,

$$\underline{M}_K(K(X)) \geq M(X) \geq \overline{M}_K(K(X))$$

(b) **Monotonicity:** For all $f : A \longrightarrow B \in \mathbb{Y}_{\mathsf{f}}$,

$$\underline{M}_K(A) \geq \underline{M}_K(B) \text{ and } \overline{M}_K(A) \geq \overline{M}_K(B)$$

(c) **Optimality:** $\underline{M}_K$ and $\overline{M}_K$ are optimal.

Proof. $(\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]}) = (\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]})^{\mathsf{op}}$ □

Our Corollary 3.36 corresponds to [17, Theorem 1]. However, in contrast to the proof in [17], our proof uses only the structural properties of the extensions rather than the formula used to compute them. Moreover, Lemma 3.35 is more general since in [17, Theorem 1], $K : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathbb{Y}, \mathbb{Y}_{\mathsf{f}})$ is fixed to be a full and faithful inclusion.

[17, Theorem 1 - (a)] can be recovered precisely by fixing K to be full and faithful functor in Corollary 3.36:

Corollary 3.37. If $\underline{M}_F$ and $\overline{M}_F$ are minimal and maximal extensions respectively of an op-monotone $M : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]})$ along a full and faithful (ff) functor $F : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathbb{Y}, \mathbb{Y}_{\mathsf{f}})$, then:

(a) **ff-Reduction:** The extensions exactly preserves the value of the resources in $\mathbb{X}$ under the action of F:

$$\overline{M}_F(F(X)) = M(X) = \underline{M}_F(F(X))$$

(b) **ff-Optimality:** For any other monotone $G : (\mathbb{X}, \mathbb{X}_{\mathsf{f}}) \longrightarrow (\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]})$, which exactly preserves the value of the resources in $\mathbb{X}$ under the action of F, that is, $(G(F(X)) = M(X))$, then for all $Y \in \mathbb{Y}$:

$$\overline{M}_F(F(Y)) \geq G(Y) \geq \underline{M}_F(F(Y))$$

Proof. Statement (a) follows from Lemma 2.4.

For Statement (b), it is given that for all $X \in \mathbb{X}$, $G(F(X)) = F(X)$. From Lemma 3.35-(c), it follows that for all $Y \in \mathbb{Y}$,

$$\underline{M}_F(Y) \geq G(Y) \geq \overline{M}_F(Y)$$

□

3.4.1 Extending bipartite entanglement monotone from pure to mixed states:

Example 3.38. We introduced the op-monotone $\mathsf{Schmidt} : (\mathsf{PureBip}, \mathsf{LOCC_p}) \longrightarrow (\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]})$ in Example 3.29. Let us extend the monotone from pure bipartite states to mixed states along $i : (\mathsf{PureBip}, \mathsf{LOCC_p}) \hookrightarrow (\mathsf{Bip}, \mathsf{LOCC})$ (defined in Lemma 3.10), something that was already done in [37], however without the general machinery for computing extensions.

Figure 8 presents the diagrams corresponding to minimal and maximal extensions of the monotone Schmidt along the inclusion. The minimal and the maximal extensions are computed using equations (3.4) and (3.5) respectively.

It was pointed out in [17] that the definition for the Schmidt entanglement monotone on mixed bipartite states introduced in [37] coincides with equation (3.5) referring to the maximal extension of Schmidt.

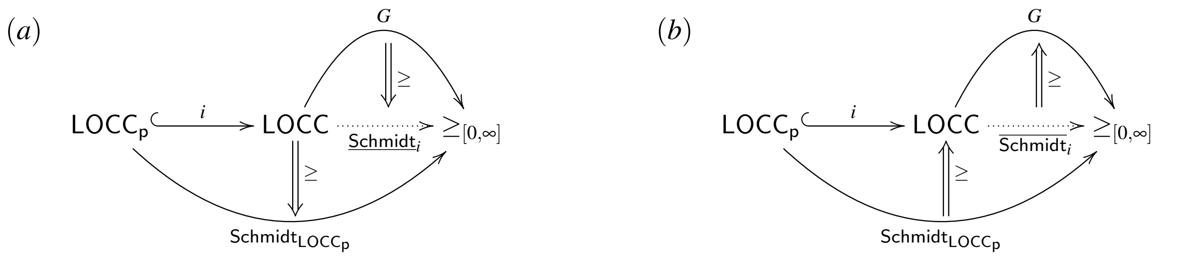


Figure 8: (a) Minimal extension of $\mathsf{Schmidt} : (\mathsf{PureBip}, \mathsf{LOCC_p})$ along $i : (\mathsf{PureBip}, \mathsf{LOCC_p}) \hookrightarrow (\mathsf{Bip}, \mathsf{LOCC})$; (b) Maximal extension of $\mathsf{Schmidt} : (\mathsf{PureBip}, \mathsf{LOCC_p})$ along $i : (\mathsf{PureBip}, \mathsf{LOCC_p}) \hookrightarrow (\mathsf{Bip}, \mathsf{LOCC})$

3.4.2 Extending classical divergences

Next we examine the properties of extensions of classical divergences to quantum setting:

Lemma 3.39. *Let* $D : (\mathsf{cDistinguish}, \mathsf{cProcessing}) \longrightarrow \left(\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]}\right)$ *be a classical divergence as defined in Definition 3.27. Let* $\underline{D}_i$ *and* $\overline{D}_i$ *be the minimal and maximal extensions respectively of* D *along* $i : (\mathsf{cDistinguish}, \mathsf{cProcessing}) \hookrightarrow (\mathsf{Distinguish}, \mathsf{Processing})$. *Then the extensions satisfy the following properties:*

(a) **Reduction***: For all* $((p,q),X) \in \mathsf{Distinguish}$, $\underline{D}_i(p||q) = D(p||q) = \overline{D}_i(p||q)$

(b) **Monotonicity***: For any* $M : ((p,q),X) \longrightarrow ((p',q'),Y)$, $\underline{D}_i(p||q) \geq \underline{D}_i(pM||qM)$ *and* $\overline{D}_i(p||q) \geq \overline{D}_i(pM||qM)$

(c) **Optimality***: Suppose* $D' : (\mathsf{Distinguish}, \mathsf{Processing}) \to \left(\mathsf{chaos}_{[0,\infty]}, \geq_{[0,\infty]}\right)$ *is any pCat functor such that for all* $((p,q),X) \in \mathsf{cDistinguish}$, $D'(i((p,q),X)) = D(((p,q),X))$. *Then for all* $((\rho,\sigma),H) \in \mathsf{Distinguish}$,

$$\underline{D}_i(\rho||\sigma) \geq G(\rho||\sigma) \geq \overline{D}_i(\rho||\sigma) \tag{3.6}$$

Proof. By Lemma 3.13, the inclusion $i : \mathsf{cDistinguish} \hookrightarrow \mathsf{Distinguish}$ is full and faithful. Hence, statement (a) and Statement (c) follows directly from ff-Reduction and ff-Optimality properties respectively in Lemma 3.35. Statement (b) follows from Monotonicity property in Lemma 3.37-(b). □

In the above statement, by the **Reduction** property, $\underline{D}_i$ and $\overline{D}_i$ reduces to classical divergence D on the classical states (pairs of density matrices with off diagonal elements to be zero). The **optimality** property ensures that, for any other quantum divergence that coincides with D on the classical states, must lie between the maximal and minimal extensions in the sense of Eqn. (3.6).

3.4.3 Extending Shannon entropy

Now we show that Kan extensions are related to some proposals of extending Shannon entropy from classical states to states of a general physical theory [2, 35, 22, 34, 8, 23]. Specifically, a measurement and a preparation extensions were proposed. Here, for simplicity, we will explain them in the context of quantum theory. In more detail, the measurement entropy H_{meas} of a quantum state ρ is defined as

$$H_{\text{meas}}(\rho) := \inf_{F} H(q), \tag{3.7}$$

where the infimum is taken over all rank-one POVMs $F := \{F_j\}$, and q is a probability distribution with $q_j := \operatorname{tr} F_j \rho$. Recall that a POVM is a collection of positive semi-definite operators $\{F_j\}$ that sum to the identity. On the other hand, the preparation entropy H_{prep} is defined as

$$H_{\text{prep}}(\rho) := \inf_{\sum_j \lambda_j \psi_j = \rho} H(\lambda), \tag{3.8}$$

where the infimum is over all convex decompositions $\sum_j \lambda_j \psi_i$ of the state ρ in terms of pure states ψ_j (recall that a quantum state ψ is pure if $\psi^2 = \psi$). In words, the measurement entropy H_{meas} is the smallest amount of randomness (as measured by Shannon entropy H) present in the probability distributions generated by rank-one POVMs on ρ. On the other hand, the preparation entropy H_{prep} is the smallest amount of randomness necessary to prepare ρ as an convex combination of pure states.

Let us consider the inclusion (Example 3.14) of resource theory of non-uniformity (given in Example 3.3) into quantum non-uniformity (given in Example 3.4). Consider extending the monotone $\mathsf{Shannon} : (\mathsf{Rand}, \mathsf{Uniform}) \longrightarrow (\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]})$ (given in Example 3.22) along the inclusion as shown in Figure 9. By the Kan extensions formula in equations (3.2) and (3.3), the minimal and maximal extension of Shannon are given as follows:

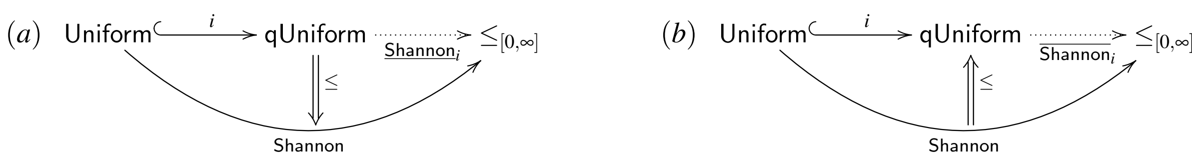


Figure 9: (a) Minimal extension of $\mathsf{Shannon} : (\mathsf{Rand}, \mathsf{Uniform})$ along $i : (\mathsf{Rand}, \mathsf{Uniform}) \hookrightarrow (\mathsf{qRand}, \mathsf{qUniform})$; (b) Maximal extension of $\mathsf{Shannon} : (\mathsf{Rand}, \mathsf{Uniform})$ along $i : (\mathsf{Rand}, \mathsf{Uniform}) \hookrightarrow (\mathsf{qRand}, \mathsf{qUniform})$

For all $\rho \in \mathsf{qUniform}$, the minimal extension $\underline{\mathsf{Shannon}}_i : \mathsf{qUniform} \longrightarrow \leq_{[0,\infty]}$ is given as:

$$\underline{\mathsf{Shannon}}_i(\rho) := \inf\{\mathsf{Shannon}(p) \mid \rho \longrightarrow i(p) \in \mathsf{qUniform}\} \tag{3.9}$$

For all $\rho \in \mathsf{qUniform}$, the maximal extension $\overline{\mathsf{Shannon}}_i : \mathsf{qUniform} \longrightarrow \leq_{[0,\infty]}$ is given as:

$$\overline{\mathsf{Shannon}}_i(\rho) := \sup\{\mathsf{Shannon}(p) \mid i(p) \longrightarrow \rho \in \mathsf{qUniform}\} \tag{3.10}$$

Let us have a closer look at equation (3.9) The only unital channels from a quantum to a classical system are given by rank 1 projective measurements $\{P_j\}$, where P_j are rank 1 orthogonal projectors. With this in mind, eqn. (3.9) can be rewritten as follows:

$$\underline{\mathsf{Shannon}}_i(\rho) := \inf_P H(q),$$

where the infimum is taken over all rank-one projective measurements $P := \{P_j\}$, and q is a probability distribution with $q_j := \operatorname{tr} P_j \rho$. Now we are going to show that $\underline{\mathsf{Shannon}}_i(\rho) = H_{\text{meas}}(\rho)$. To this end, notice that $\underline{\mathsf{Shannon}}_i(\rho) \geq H_{\text{meas}}(\rho)$ because the infimum in the definition of $\underline{\mathsf{Shannon}}_i(\rho)$ is over a smaller set. In theorem 5.4.15 of [34] it was shown that $H_{\text{meas}}(\rho)$ is achieved by considering the spectral POVM, which is a rank-1 projective measurement. Being $\underline{\mathsf{Shannon}}_i(\rho)$ defined as the infimum over rank-1 projective measurements, then we also have $\underline{\mathsf{Shannon}}_i(\rho) \leq H_{\text{meas}}(\rho)$, from which we conclude that $\underline{\mathsf{Shannon}}_i(\rho) = H_{\text{meas}}(\rho)$. Since $H_{\text{meas}}(\rho)$ is achieved by the spectral measurement, we know that $H_{\text{meas}}(\rho) = H(p)$, where p denotes the classical vector of the spectrum of ρ. This shows that H_{meas} as defined in equation (3.7) is indeed a monotone, as it coincides with the minimal Kan extension.

Let us now have a closer look at equation (3.10). The only unital channels from a classical to a quantum system are given by preparations of a convex combination of pure states $\{\psi_j\}$ associated with an orthonormal basis of the Hilbert Space corresponding to the quantum system, where the coefficients are the entries of the classical state on which the channel acts. With this in mind, eqn (3.10) can be rewritten as follows:

$$\overline{\mathsf{Shannon}}_i(\rho) := \sup_{\sum_j \lambda_j \psi_j = \rho} H(\lambda),$$

where the supremum is taken over all decompositions of ρ into *orthogonal* pure states. Now, we we observe that all such decompositions are diagonalizations of ρ (that is, $\rho = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$ with λ being a probability distribution), and therefore they have the same coefficients λ_j, which are the eigenvalues of ρ. In other words, $\overline{\mathsf{Shannon}}_i(\rho) = H(p)$, where p denotes the classical vector of the spectrum of ρ. Since there is only one vector λ (up to permutation) to optimize over, the supremum can be replaced with an infimum. With this in mind, we obtain an expression that is close the preparation entropy.

$$\overline{\mathsf{Shannon}}_i(\rho) := \inf_{\sum_j \lambda_j \psi_j = \rho} H(\lambda),$$

where the infimum is taken over all decompositions of ρ into *orthogonal* pure states. In Theorem 5.4.15 of [34] it was shown that $H_{\text{prep}}(\rho) = H(p)$, from which we have that $\overline{\mathsf{Shannon}}_i(\rho) = H_{\text{prep}}(\rho)$. This shows that H_{meas} as defined in equation (3.8) is indeed a monotone, as it coincides with the maximal Kan extension.

Notice that in this example, the minimal and maximal Kan extensions coincide.

4 Conclusion

In this article, we studied resource theories as partitioned categories (pCats) and relationship between resource theories as pCat functors thereof. A partitioned category (pCat) is a category with a chosen subcategory of free transformations. In this framework, a monotone for a resource theory can be viewed as a pCat functor from the theory into $(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]})$ where the pCat $(\mathsf{chaos}_{[0,\infty]}, \leq_{[0,\infty]})$ represents the partial order $([0,\infty], \leq)$.

We showed that a monotone can be extended from one theory to another using Kan extensions. We applied our framework to extend entanglement monotones for bipartite pure states to bipartite mixed states, to extend classical divergences to the quantum setting, and to extend non-uniformity monotone from classical probabilistic theory to quantum theory.

This project was inspired by Gour and Tomamichel's work [17] (see also [12]), which uses a set-based framework to provide formulae for the minimal and maximal extensions of a monotone for a resource theory that embeds (fully and faithfully) in a larger theory. The goal of our work was to present resource theories and monotones in a framework such that the extension formulae for monotones arise naturally. We found that they are precisely given by the well-studied notion of Kan extensions. On top of providing a natural ground to study extensions of monotones, we should also note that our categorical framework is also more general than the framework in [17], in that it can be used to compute monotone extensions when the pCat functor between resource theories is not a full and faithful embedding.

Acknowledgments

P. V. S. thanks Dr. Spencer Breiner, Dr. Joe Moeller, and Dr. Eswaran Subramanian for valuable discussions. C. M. S. acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant "The power of quantum resources" RGPIN-2022-03025 and the Discovery Launch Supplement DGECR-2022-00119.

References

[1] J. C. Baez, T. Fritz & T. Leinster (2011): *A characterization of entropy in terms of information loss*. Entropy 13(11), pp. 1945–1957, doi:10.3390/e13111945.

[2] H. Barnum, J. Barrett, L. Orloff Clark, M. Leifer, R. Spekkens, N. Stepanik, A. Wilce & R. Wilke (2010): *Entropy and information causality in general probabilistic theories*. New J. Phys. 12(3), p. 033024, doi:10.1088/1367-2630/12/3/033024.

[3] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres & W. K. Wootters (1993): *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*. Phys. Rev. Lett. 70, pp. 1895–1899, doi:10.1103/PhysRevLett.70.1895.

[4] C. H. Bennett & S. J. Wiesner (1992): *Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states*. Phys. Rev. Lett. 69, pp. 2881–2884, doi:10.1103/PhysRevLett.69.2881.

[5] F. Borceux (1994): *Handbook of Categorical Algebra*. 1, Cambridge University Press, Cambridge, doi:10.1017/CBO9780511525858.

[6] F. G. S. L. Brandão & G. Gour (2015): *Reversible Framework for Quantum Resource Theories*. Phys. Rev. Lett. 115, p. 070503, doi:10.1103/PhysRevLett.115.070503.

[7] F. Buscemi & G. Gour (2017): *Quantum relative Lorenz curves*. Phys. Rev. A 95, p. 012110, doi:10.1103/PhysRevA.95.012110.

[8] G. Chiribella & C. M. Scandolo (2016): *Entanglement as an axiomatic foundation for statistical mechanics*. arXiv:1608.04459 [quant-ph]. Available at `http://arxiv.org/abs/1608.04459`.

[9] E. Chitambar & G. Gour (2019): *Quantum resource theories*. Rev. Mod. Phys. 91, p. 025001, doi:10.1103/RevModPhys.91.025001.

[10] B. Coecke, T. Fritz & R. W Spekkens (2016): *A mathematical theory of resources*. Inf. Comput. 250, pp. 59–86, doi:10.1016/j.ic.2016.02.008.

[11] T. M. Cover & J. A. Thomas (2006): *Elements of information theory*, 2nd edition. Wiley, Hoboken, doi:10.1002/047174882X.

[12] T. Gonda & R. W. Spekkens (2019): *Monotones in General Resource Theories*. arXiv:1912.07085 [quant-ph]. Available at `https://arxiv.org/abs/1912.07085`.

[13] J. Goold, M. Huber, A. Riera, L. del Rio & P. Skrzypczyk (2016): *The role of quantum information in thermodynamics—a topical review*. J. Phys. A 49(14), p. 143001, doi:10.1088/1751-8113/49/14/143001.

[14] G. Gour (2021): *Uniqueness and Optimality of Dynamical Extensions of Divergences*. PRX Quantum 2, p. 010313, doi:10.1103/PRXQuantum.2.010313.

[15] G. Gour, M. P. Müller, V. Narasimhachar, R. W. Spekkens & N. Yunger Halpern (2015): *The resource theory of informational nonequilibrium in thermodynamics*. Phys. Rep. 583, pp. 1–58, doi:10.1016/j.physrep.2015.04.003.

[16] G. Gour & C. M. Scandolo (2020): *Dynamical Entanglement*. Phys. Rev. Lett. 125, p. 180505, doi:10.1103/PhysRevLett.125.180505.

[17] G. Gour & M. Tomamichel (2020): *Optimal extensions of resource measures and their applications*. Phys. Rev. A 102(6), doi:10.1103/physreva.102.062401.

[18] G. H. Hardy, J. E. Littlewood & G. Pólya (1929): *Some simple inequalities satisfied by convex functions*. Messenger Math. 58(145–152), p. 310.

[19] M. Horodecki & J. Oppenheim (2013): *(Quantumness in the context of) resource theories*. Int. J. Mod. Phys. B 27(01n03), p. 1345019, doi:10.1142/S0217979213450197.

[20] R. Horodecki, P. Horodecki, M. Horodecki & K. Horodecki (2009): *Quantum entanglement*. Rev. Mod. Phys. 81, pp. 865–942, doi:10.1103/RevModPhys.81.865.

[21] D. M Kan (1958): *Adjoint functors*. Trans. Am. Math. Soc. 87(2), pp. 294–329, doi:10.1090/S0002-9947-1958-0131451-0.

[22] G. Kimura, K. Nuida & H. Imai (2010): *Distinguishability measures and entropies for general probabilistic theories*. Rep. Math. Phys. 66(2), pp. 175–206, doi:10.1016/S0034-4877(10)00025-X.

[23] M. Krumm, H. Barnum, J. Barrett & M. P. Müller (2017): *Thermodynamics and the structure of quantum theory*. New J. Phys. 19(4), p. 043025, doi:10.1088/1367-2630/aa68ef.

[24] M. O. Lorenz (1905): *Methods of Measuring the Concentration of Wealth*. Publ. Am. Stat. Assoc. 9(70), pp. 209–219, doi:10.1080/15225437.1905.10503443.

[25] M. Lostaglio (2019): *An introductory review of the resource theory approach to thermodynamics*. Rep. Prog. Phys. 82(11), p. 114001, doi:10.1088/1361-6633/ab46e5.

[26] S. Mac Lane (1998): *Categories for the working mathematician*, 2nd edition. Graduate Texts in Mathematics, Springer Science & Business Media, New York, doi:10.1007/978-1-4757-4721-8.

[27] A. W. Marshall, I. Olkin & B. C. Arnold (2011): *Inequalities: Theory of Majorization and Its Applications*. Springer Series in Statistics, Springer, New York, doi:10.1007/978-0-387-68276-1.

[28] M. A. Nielsen & I. L. Chuang (2011): *Quantum Computation and Quantum Information*, 10th anniversary edition. Cambridge University Press, Cambridge, doi:10.1017/CBO9780511976667.

[29] J. M. Renes (2016): *Relative submajorization and its use in quantum resource theories*. J. Math. Phys. 57, p. 122202, doi:10.1063/1.4972295.

[30] E. Riehl (2017): *Category Theory in Context*. Aurora: Dover Modern Math Originals, Dover Publications, Mineola, NY.

[31] E. Ruch & A. Mead (1976): *The principle of increasing mixing character and some of its consequences*. Theor. Chim. Acta 41, pp. 95–117, doi:10.1007/BF01178071.

[32] E. Ruch, R. Schranner & T. H. Seligman (1978): *The mixing distance*. J. Chem. Phys. 69(1), pp. 386–392, doi:10.1063/1.436364.

[33] E. Ruch, R. Schranner & T. H. Seligman (1980): *Generalization of a theorem by Hardy, Littlewood, and Pólya*. J. Math. Anal. Appl. 76(1), pp. 222–229, doi:10.1016/0022-247X(80)90075-X.

[34] C. M. Scandolo (2018): *Information-theoretic foundations of thermodynamics in general probabilistic theories*. Ph.D. thesis, University of Oxford. Available at `https://arxiv.org/abs/1901.08054`.

[35] A. J. Short & S. Wehner (2010): *Entropy in general physical theories*. New J. Phys. 12(3), p. 033023, doi:10.1088/1367-2630/12/3/033023.

[36] A. Streltsov, G. Adesso & M. B. Plenio (2017): *Colloquium: Quantum coherence as a resource*. Rev. Mod. Phys. 89, p. 041003, doi:10.1103/RevModPhys.89.041003.

[37] B. M. Terhal & P. Horodecki (2000): *Schmidt number for density matrices*. Phys. Rev. A 61, p. 040301, doi:10.1103/PhysRevA.61.040301.

[38] M. M. Wilde (2017): *Quantum Information Theory*, 2nd edition. Cambridge University Press, Cambridge, doi:10.1017/9781316809976.

Fibrational Linguistics (FibLang): Language Acquisition

Fabrizio Genovese
0000-0001-7792-1375
20[]*
fabrizio.romano.genovese@gmail.com

Fosco Loregian
0000-0003-3052-465X
Tallinn University of Technology†
fosco.loregian@gmail.com

Caterina Puca
Quantinuum, 17 Beaumont Street, Oxford, OX1 2NA, United Kingdom
caterpuca@gmail.com

In this work we show how FibLang, a category-theoretic framework concerned with the interplay between language and meaning, can be used to describe *vocabulary acquisition*, that is the process with which a speaker p acquires new vocabulary (through experience or interaction).

We model two different kinds of vocabulary acquisition, which we call 'by example' and 'by paraphrasis'. The former captures the idea of acquiring the meaning of a word by being shown a witness representing that word, as in 'understanding what a cat is, by looking at a cat'. The latter captures the idea of acquiring meaning by listening to some other speaker rephrasing the word with others already known to the learner.

We provide a category-theoretic model for vocabulary acquisition by paraphrasis based on the construction of free promonads. We draw parallels between our work and Wittgenstein's dynamical approach to language, commonly known as 'language games'.

1 Introduction

Language has always been characterised as a distinctive, exclusive feature of human beings, yet children are not *born* fluent in any language at all. Along the history of human thought, this apparent discrepancy has promptly led to philosophical speculation on the innateness of language [24], which was subsequently replaced with a more cautious theory of innateness of syntactic structures [6]. On the other hand, empiricists such as J. Locke [23] argued that the human mind had to be thought of as a *tabula rasa*.

Either way, these heterogeneous philosophical stances share the necessity of formalising a common process: *language acquisition*, i.e. the process in which proficiency in a language increases with time or solicitation.[1] Although years of debate in linguistics have not come to a definite resolution yet, the transformationalist orientation of Chomsky has received severe criticism in the 20th century with the growing development of linguistic philosophy and with the renewed interest in L. Wittgenstein's ideas.

In particular, Chomsky's model of language acquisition has been accused of being overly reductionist and mechanical [29], as opposed to Wittgenstein's dynamic theory that embraces *context* as a fundamental aspect of meaning analysis [34, 33].

*Supported by the Independent Ethvestigator Program.

†Supported by the ESF funded Estonian IT Academy research measure (project 2014–2020.4.05.19–0001).

[1]Here and in the rest of paper we will be referring to first language acquisition only: although similarities have been pointed out concerning second language acquisition, such as *the silent period* [10], several substantial differences separate the two processes [16]. For instance, there is evidence that the learner's first language slows the development of acquisitional sequences predicted by the Natural Order Hypothesis [18, 27]. Additionally, according to the Critical Period Hypothesis, after puberty, lateralisation is accomplished, and reduced plasticity of the brain can compromise the fluent acquisition of a second language [3].

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 224–236, doi:10.4204/EPTCS.380.13

In this latter perspective, syntax and semantics must interact and reciprocally influence each other during communication, a process also referred to as a *language game* [5]. More precisely, language games can serve as a tool to untie the problems of context-dependency and ambiguity regarding words with multiple semantic interpretations.

The recently developed *FibLang* framework [14] takes a stab at tackling the enticing and deep problem of language, offering a category-theoretic framework concerned with describing the interplay between meaning and structure in natural language. As a theory, FibLang relies on fibered categories [17, 25, 32, 30]; the main idea underlying FibLang is characterizing linguistic meaning as *fibered over grammar*.

Here we argue that Wittgenstein's perspective hints at a fibrational formalisation of language acquisition. We will substantiate our hypothesis by formalising vocabulary acquisition in FibLang and show how the context-dependency and semantic ambiguity aspects underlying language games are organically embraced in our categorical description.

Since our fibrational approach to language acquisition naturally encodes agency, it is out-of-the-box compatible with applications to learning tasks in natural language processing (NLP). In this sense, it also enriches the static perspective of DisCoCat [7], which by the way, has also been recently revised using tools from categorical game theory applied to language games in [15].

In point of fact, in more recent years, a different framework, named DisCoCirc, has been adopted to allow for a dynamic flow between syntax and semantics [8]. Implementation aspects regarding quantum computers suggested this switch and, as a side-effect, it provides stronger foundations to the philosophical stance of FibLang.

Structure of the paper. In section 2, we will recall the basic definitions of FibLang. We will then provide our description of vocabulary acquisition by example in section 3. Subsequently, we will define the tool of explanations in section 4, and will use them to define vocabulary acquisition by paraphrasis in section 5. Finally, at the end of section 5, we provide a construction (cf. Construction 5.3) with which to show how our formalisation of vocabulary acquisition can be used to enrich grammar by appropriately acknowledging semantic interrelations.

2 Rappels of FibLang

FibLang was introduced in a prior installment of this series, [14]. Its main idea can be summarized as follows: whereas it is reasonable to believe that language has at least some degree of compositionality, especially when describing grammar, it becomes much more difficult to substantiate this position when it comes to describe meaning. Indeed, compositional –and especially cognitive– models of meaning such as Gärdenfors' [12] are prone to criticism on multiple fronts, not last the fact that a truly universal model of meaning is very difficult to define because of cultural and cognitive differences between speakers.

To circumvent these problems FibLang focuses on describing the interplay between meaning and structure in abstract terms in a way that is agnostic to the particular model one chooses to represent either. This vision is reified in the main idea being that a 'language' is a category $\mathcal{L}$ of some sort, and a speaker p of a language $\mathcal{L}$ is a *fibration* $\left[\begin{smallmatrix}\mathcal{E}^p\\ p^\sharp \downarrow\\ \mathcal{L}\end{smallmatrix}\right]$ over $\mathcal{L}$.[2] The defining property of a fibration is that its domain $\mathcal{E}^p$ is obtained gluing together all the various fibres $\mathcal{E}_L$ over objects of $\mathcal{L}$ in a coherent manner.

Borrowing from the work of Lambek [21], $\mathcal{L}$ represents a 'language category' while borrowing from fibration theory [**CLLT**, 2] $\mathcal{E}^p$, called the *total category* of the fibration, represents a 'semantics category'

[2] In [14] fibrations are denoted using the superscript $-^\sharp$ to distinguish them visually from bare functors.

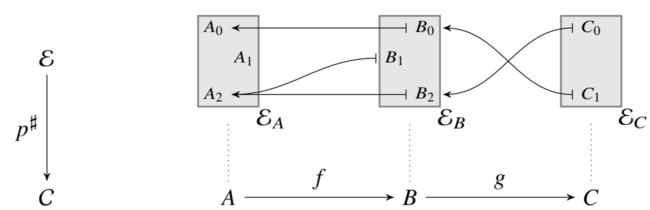
for the speaker p. This semantics category could be thought of as any sort of cognitive or distributional model of meaning for the language in question: this is to say that we are not particularly attached to any specific model but rather aim at the highest possible generality.

To arrive at this idea, the starting point in [14] constitutes the most general and yet reasonable assumption one could make, namely *there is some structure-preserving map from what a speaker means to what a speaker says*. Formally, this directly translates to modelling speakers as simple functors between categories. From this, multiple reasons are stated that lead to believe that the language category $\mathcal{L}$ should be treated as something that can be explicitly modelled and studied, while $\mathcal{E}^p$ should be treated as a black box. Then, it is shown how every functor can be factorised into a fibration via Theorem 2.4, obtaining a more workable definition of a speaker from the very abstract one we started from.

Definition 2.1 (Fibration)**.** A functor $p^\sharp : \mathcal{E} \to \mathcal{C}$ is a *discrete fibration* (for us, just a *fibration*) if, for every object E in $\mathcal{E}$ and every morphism $f : C \to p^\sharp E$ with C in $\mathcal{C}$, there exists a unique morphism $h : E' \to E$ such that $p^\sharp h = f$.

Notation 2.2. The domain of a fibration $p^\sharp : \mathcal{E} \to \mathcal{C}$ is usually called the *total category* of the fibration, and its codomain is the *base category*. Given any functor p we can define the *fibre* of p over an object $C \in \mathcal{C}$, i.e. the subcategory $\mathcal{E}_C = \{f : E \to E' \mid pf = 1_C\} \subseteq \mathcal{E}$.

Intuitively, a fibration is a functor $p^\sharp : \mathcal{E} \to \mathcal{C}$ that realises the category $\mathcal{E}$ as a 'covering' of $\mathcal{C}$, in such a way that morphisms in $\mathcal{C}$ can be lifted to $\mathcal{E}$, to induce functions between the fibres in $\mathcal{E}$, called *reindexing functions*. We have represented an elementary example of fibration in the following figure, with the action of the reindexing functions between the elements in the fibres made explicit. The gray rectangles are the *fibres* of the fibration.



A fundamental result in the theory of fibrations that we will often use is that fibrations over a category $\mathcal{L}$ are equivalent to functors out of $\mathcal{L}$, with codomain the category of sets and functions. More details on the construction, and a full explanation of its usefulness for FibLang, can be found in [14, A.6]; a classical reference for the theory of fibrations is [17].

Theorem 2.3. There is a category $\mathsf{DFib}/\mathcal{L}$ of fibrations over a given $\mathcal{L}$, where an object is a fibration $\begin{bmatrix} \mathcal{E}^p \\ p^\sharp \downarrow \\ \mathcal{L} \end{bmatrix}$ and a map $h : \begin{bmatrix} \mathcal{E}^p \\ p^\sharp \downarrow \\ \mathcal{L} \end{bmatrix} \to \begin{bmatrix} \mathcal{E}^q \\ q^\sharp \downarrow \\ \mathcal{L} \end{bmatrix}$ is a functor $h : \mathcal{E}^p \to \mathcal{E}^q$ such that $q^\sharp \cdot h = p^\sharp$.

There is an equivalence of categories:

$$\nabla - : \mathsf{DFib}/\mathcal{L} \cong [\mathcal{L}^{\mathrm{op}}, \mathsf{Set}] : \textstyle\int - \tag{2.1}$$

where at the right-hand side we have the category of all functors $\mathcal{L} \to \mathsf{Set}$ and natural transformations thereof. The functor $\int -$ is often called *the category of elements construction*, or in its most general form *the Grothendieck constuction*.

Theorem 2.3 is also instrumental in pointing out how FibLang– which postulates an approach going from meaning to language – can be made compatible with traditional models of meaning such as DisCoCat

– which postulate an approach going from language to meaning[3]. Many conceptual reasons are given in [14] to prefer the meaning-to-language approach, but Theorem 2.3 shows how the two are faces of the same medal.

As we remarked above, FibLang relies on some machinery to turn a model for speakers consisting of simple functors into a model consisting of fibrations. The main theorem allowing us to do so is the following:

Theorem 2.4 ([31, Theorem 3])**.** Any functor $p : \mathcal{D}^p \to \mathcal{L}$ can be written as a composition of functors $\mathcal{D}^p \xrightarrow{s} \mathcal{E}^p \xrightarrow{p^\sharp} \mathcal{L}$, such that $p^\sharp$ is a fibration.

We will make heavy use of Theorem 2.4 in the following sections to model language acquisition.

3 Vocabulary acquisition by direct example

Vocabulary acquisition denotes the act of acquiring meanings for a word previously unknown [26]. In this work, we aim to describe two main modes of vocabulary acquisition. In this section, we focus on *vocabulary acquisition by direct example*: this is the easiest method of language acquisition that we can describe and a commonly used method in children's education and monolingual fieldwork [11, 28]: it simply works by pointing at something and saying the word one is referring to. This process of language acquisition, as Wittgenstein explains in [35], can serve as a primitive example of a language game.

Example 3.1. Consider the following dialogue:

ALICE: Look, a cat!
BOB: A what?
[*Alice points to a cat*]
ALICE: That, a cat!
BOB: Oh!

What happened in that 'Oh!' can be mathematically modelled as a colimit in the fibrations that represent Alice and Bob.

Definition 3.2 (Vocabulary acquisition by language example)**.** Consider two speakers $\left[\begin{smallmatrix} & \mathcal{E}^p \\ p^\sharp & \downarrow \\ & \mathcal{L}\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} & \mathcal{E}^q \\ q^\sharp & \downarrow \\ & \mathcal{L}\end{smallmatrix}\right]$, which we will call *teacher* and *learner*, respectively.

Suppose that, for some $L \in \mathcal{L}$ – called *the linguistic element to learn*[4] – we have that $\mathcal{E}^p_L \neq \emptyset$ and $\mathcal{E}^q_L = \emptyset$. Fix a subset $S \subseteq \mathcal{E}^p_L$, called an *example* for L. Then we can define a new category $\mathcal{F}^q$ as follows:

$$\mathrm{obj}(\mathcal{F}^q) := \mathrm{obj}(\mathcal{E}^q) \sqcup S$$
$$\hom(\mathcal{F}^q) := \hom(\mathcal{E}^q)$$

[3]In the particular case of DisCoCat it would be more proper to say that the chosen approach is from *grammar* to *semantics*. The apparent dissonance is resolved by taking into account the agnostic approach of FibLang. The language category $\mathcal{L}$ can be purely grammatical, employing for instance a Lambek's pregroup [20] as in DisCoCat, or more expressive. Similarly, the meaning category $\mathcal{E}^p$ for a speaker p could be purely semantical – for instance, a distributional model or a conceptual space [12] – or more expressive.

[4]We will use the wording *linguistic elements* referring to words, entire sentences or something else, depending on the model we chose for $\mathcal{L}$, without committing to a particular choice. More formally, a linguistic element is the (possibly nonfull) subcategory of $\mathcal{L}$ spanned by a certain choice of objects.

and a functor $T : \mathcal{F}^q \to \mathcal{L}$ agreeing with $\begin{bmatrix} \mathcal{E}^q \\ q^\sharp \downarrow \\ \mathcal{L} \end{bmatrix}$ on every fibre $L' \neq L$, and sending every object of S to L. Relying on Theorem 2.4, the new fibration modelling the speaker q after learning L is the factorization $\tilde{q}^\sharp$ such that:

$$T = \left(\mathcal{F}^q \xrightarrow{s} \mathcal{E}^{\tilde{q}} \xrightarrow{\tilde{q}^\sharp} \mathcal{L}\right).$$

Let us unpack this definition. We consider two speakers p, q of the same language $\mathcal{L}$. Speaker q does not know the meaning of a given linguistic element which corresponds to an object $L \in \mathcal{L}$ –which is why we call q *learner*. The fibre $\mathcal{E}_L$ in $\mathcal{E}$ is the empty set, as L has no meaning for q. On the other hand, speaker p has some model of meaning for L –which is why we call p *teacher*, and we assume the fibre $\mathcal{E}^p_L$ to be not empty.

If p points out an instance of L to q, as in Example 3.1, it is reasonable to assume that the instance in question is itself part of the fibre $\mathcal{E}^p_L$, as p recognises the example as an instance of the concept L. So we postulate that the example identifies a subset $S \subseteq \mathcal{E}^p_L$. While following along with the example, q incorporates the set S as a fibre over L by extending its meaning.

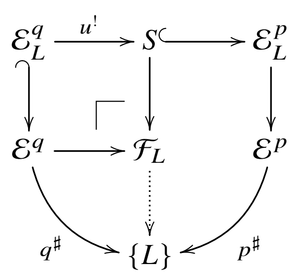
On the intuitive side, there is no reason to claim that forcefully adding a scattered set of notions to the ones previously mastered by a speaker is enough to let q 'understand' that said set of notions defines a new language term. Instead, to attain this level of understanding, q has to build meaningful relations between the new term L and all the others they master, in concordance with all the pre-existing relations between general terms.

On the mathematical side, there is no reason why the functor obtained by forcefully adding a nonempty fibre to a previously empty one shall remain a fibration; to be such, we must let the new fibre interact well with the environment, in concordance with the hom-sets $\mathcal{L}(X, L)$ and $\mathcal{L}(L, Y)$. This is why we consider the comprehensive factorisation of T and take into consideration only $\tilde{q}^\sharp$ instead of the whole T.

Example 3.3 (A language game)**.** Consider the following language game borrowed from Wittgenstein's *Philosophical Investigations* [35]: a builder A asks his assistant B to pass him the stones with which they are building, in the order A calls them out. In this situation, let us imagine that the language they use consists of only four words: block, pillar, slab, beam.

This language game can be interpreted as a particular case of vocabulary acquisition by example. The builder A, when requesting a slab, specifies the linguistic element slab that he wants to learn. On the other hand, the teacher B associates the right stone to the word slab and hands it over to A. This way, the fibre of A over the word slab is no longer empty since A has incorporated as part of its meaning the stone received by B.

Crucially, in the construction of Definition 3.2 we suppose that q has no meaning at all for the word L. What if this is not the case? More generally, we can model a vocabulary acquisition between two speakers that share some prior knowledge of L as a *pushout*, but this will render necessary the introduction of some compatibility conditions. Indeed, consider the following diagram:



(3.1)

The arrow marked as $u^!$ always exists whenever $\mathcal{E}_L^q$ is the empty set because of the universal property of initial objects. Whenever $\mathcal{E}_L^q$ is not empty, the arrow $u^!$ will have to be explicitly instantiated. This must be understood as: the meaning that q has for L must be compatible with the subset S that constitutes the meaning of the example for p. In simpler words, the example p is making *must make sense for* q.

4 Explanations

Our task for the remainder of this work is about modelling *vocabulary acquisition by paraphrasis*, which denotes the task of explaining a word by describing it with language, as it happens in a dictionary. To do so, we first need to model what an explanation is.

In stark contrast with works that are exclusively based on syntax, FibLang can describe linguistic constructions that have meaning for a given speaker *despite their ungrammaticality*. For example, it is a fact of life that often one can understand the meaning of unsound sentences, such as in *'I hungry now'*: this is because every proficient speaker can *interpolate* what is missing in the message they receive by analysing its context, in order to build a grammatical sentence. For that matter, this is precisely how Wittgenstein's language games unravel in disambiguating a sentence.

On the other hand, there are perfectly grammatical sentences, such as the famous *'Dogs dogs dog dog dogs'* (cf. [1]) that are grammatical (since *'dog'* is both a verb and a noun in English) but have no meaning when they are translated to any other speaker out of context. This tension stems from the fact that *acceptability* –i.e. the fact that a sentence has a meaning and *grammaticality* –the fact that a sentence is formed in observance of some generation rules do not fully overlap (cf. [22]).

As such, leveraging a grammar-based approach to infer meaning in semantics, as in the case of DisCoCat, is going to miss something –an important part, we say. By contrast, the fibrational approach of FibLang allows more fine-grained bookkeeping: grammaticality is completely encapsulated in the category $\mathcal{L}$ modelling language, whereas acceptability comes into play in the following definitions:

Definition 4.1 (Finite category, finite diagram)**.** A *finite category* is a category $\mathcal{A}$ having a finite set of morphisms. A *finite diagram* valued in $\mathcal{L}$ is a functor $\mathcal{A} \to \mathcal{L}$ whose domain is a finite category.

Definition 4.2 (Explanation)**.** Consider a speaker $\left[p^\sharp \begin{array}{c} \mathcal{E}^p \\ \downarrow \\ \mathcal{L} \end{array} \right]$. Fix moreover an object L of $\mathcal{L}$. An *explanation for L according p* is a finite diagram $D_L : \mathcal{A} \to \mathcal{L}$ such that the limit $\hat{L}$ of the diagram

$$\mathcal{A} \xrightarrow{D_L} \mathcal{L} \xrightarrow{\nabla p^\sharp} \mathsf{Set} \tag{4.1}$$

is a subset of the fibre $\mathcal{E}_L^p$ (Here $\nabla -$ is the functor of Theorem 2.3). If $\hat{L} = \mathcal{E}_L^p$, we call the explanation *exact*.

Here, the functor D_L is picking a collection of linguistic elements in the language intending to describe L. The finiteness requirement for $\mathcal{A}$ stems from the obvious fact that a linguistic sentence is always made of a finite number of words. The linguistic elements picked by D_L are then sent to the sets of sections sitting over them under $p^\sharp$, in accordance with Theorem 2.3. In postulating that p 'knows' how to make sense of some given complex concept $L \in \mathcal{L}$ by breaking it down into some atomic constituents, it becomes reasonable to assume that the combination of these atomic meanings – that is, the limit $\hat{L}$ – must itself be a concept representing L, and thus be a subset of the fibre over it. In light of this interpretation, an exact explanation is a selection of linguistic elements describing the fibre over L completely, that is, an explanation that more than any other, conveys exactly all the nuances that the meaning of L can assume according to the speaker.

Remark 4.3 (Acceptability vs. grammaticality)**.** As we remarked at the beginning of this section, a remarkable feature of Definition 4.2 is that explanations can be ungrammatical, as they evaluate the acceptability and not grammaticality. For instance, if $\mathcal{L}$ is a pregroup [20], the functor D_L can specify a bunch of elements that possibly do not reduce to a sentence type.

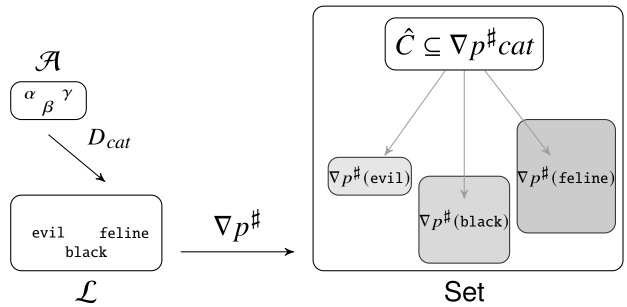


Figure 1: An explanation of the word *'cat'* as a *black, evil feline*: The limit of a certain diagram of elements having values in the fibres over concepts like *'black'*, *'evil'*, and *'feline'* is required to be in the fiber over *'cat'*.

Remark 4.4 (On the nature of explanations)**.** Explanations are engineered to be far from unique: there may be many functors D_L, with different domains, satisfying the property of Definition 4.2. This conforms to the idea that the same concept L can be explained in different ways (the same object can be the limit of many different diagrams) by different people, at different moments in time, in different cultures, in different communication settings.

Interestingly, for every object $L \in \mathcal{L}$, there is always an exact explanation for it by choosing $\mathcal{A}$ to be the terminal category and D_L to be the functor picking $L \in \mathcal{L}$; This explanation is *tautological*, as it affirms in essence that the meaning of the word *'cat'* is *'cat'*.

5 Vocabulary acquisition by paraphrasis

As remarked in the opening of section 4, by *vocabulary acquisition by paraphrasis* we denote the mechanism by which we give meaning to things using linguistic explanation. This is the method of language acquisition commonly used in books and papers like this one, in conversations with a blind man about cathedrals, and pretty much everywhere p needs to convey the meaning of something they know to a q who does not.

Example 5.1. Consider the following dialogue:

ALICE: I adopted a cat!
BOB: A what?
ALICE: You know, a cat: one of those felines.
BOB: Oh, you mean, like a tiger?
ALICE: No: a cat is smaller and it comes in various colours, not only stripes. Mine is black.
BOB: Oh, maybe I see. A cat is like. . . a lynx.
ALICE: Well, almost; black cats are cursed.
BOB: Ah, now I see.

Bob still has probably never seen a real cat when the dialogue ends, and he would have trouble recognizing one. Still, he can get a rough idea of 'a cat' by mixing concepts for which he already had a model of meaning.

Mathematically, what we would like to do is proceed as we did in Definition 3.2, by incorporating an explanation as in Definition 4.2 into the fibration of the learning speaker. In short, our strategy is the following:

- p has an explanation $D_L : \mathcal{A} \to \mathcal{L}$ of some word L;
- p shares $\mathcal{A}$ with q. This represents the act of p uttering the explanation to q;
- q computes the limit of the diagram $\mathcal{A}^{\mathrm{op}} \xrightarrow{D_L^{\mathrm{op}}} \mathcal{L}^{\mathrm{op}} \xrightarrow{\nabla q^\sharp} \mathsf{Set}$;
- q includes this limit in their own fibre over L.

Difficulties arise in the last point. The problem we face is that an explanation is a limit, and as such, a particular kind of cone in Set is composed not only of objects but also of morphisms. Unfortunately, including the whole cone in the total category of a fibration sometimes entails the impossibility of defining the functor $T : \mathcal{F}^q \to \mathcal{L}$ as we did in Definition 3.2.

Example 5.2. To see a practical example, consider the explanation as in Figure 1. Here, the cone legs are morphisms in Set connecting concepts signifying different *nouns* (for example, from *'cat'* to *'feline'*). If $\mathcal{L}$ is a Lambek pregroup, the only morphisms are reductions, so there are no morphisms between nouns in $\mathcal{L}$. Thus in including the whole cone in $\mathcal{F}^q$, we could not define the functor T on the cone legs. This is related to our previous considerations about grammaticality and acceptability: pregroups only represent grammatical connections and fail to see conceptual relatedness. Consequently, although pregroup grammars have been enriched to identify differently worded sentences in [9], the added relations are exquisitely grammatical and cannot account for context.

One possible solution to this problem would be to consider only the limit itself and proceed as in Definition 3.2. However, in doing so, we would miss the big opportunity of adding meaning while being mindful of the context in which this meaning lives.

Another –more stimulating– solution considers Wittgenstein's approach to language, which we briefly summarized in the introduction, and leverages the interplay between grammar and semantics. The main idea here is to use semantics data - i.e. the limiting cone - to enrich the grammar with new morphisms. Going back to Example 5.2, this means adding new morphisms to the pregroup grammar $\mathcal{L}$ that do not represent reductions but some sort of 'semantic connection' between words.

To add new morphisms to a category, we highlight the following procedure. Recall that there is an adjunction

$$(\_)^\delta : \mathsf{Set} \rightleftarrows \mathsf{Quiv} : (\_)_0 \tag{5.1}$$

sending every quiver Q to the set Q_0 of its vertexes, and every set X to the 'discrete quiver' X^δ with no edges, and an adjunction

$$F : \mathsf{Quiv} \rightleftarrows \mathsf{Cat} : U \tag{5.2}$$

sending every quiver Q to the free category FQ generated by it, and every category C to its underlying quiver UC.

The following construction is a recipe to add the set of edges E of a quiver Q to a (small) category C and form a category out of it when C and Q have the same set of vertices. Regarding a category as a monad in the bicategory of spans, the following Construction 5.3 consists of a particular instance of a free-monoid construction in a decent enough monoidal category (cf. [19]).

Construction 5.3 (FP construction)**.** Let C be a category, and $Q : E \rightrightarrows C_0$ a quiver over the same set C_0 of objects of C.[5]

[5] The notation is slightly overloaded here because we denote Q_0 the set of vertices of a quiver *and* C_0 the set of objects of the underlying quiver of a category C. This confusion is harmless.

- consider the underlying quiver UC of C;
- compute the pushout

$$\begin{array}{ccc} C_0^\delta & \longrightarrow & Q \\ \downarrow & & \downarrow \\ UC & \longrightarrow & UC +_{C_0^\delta} Q \end{array} \tag{5.3}$$

  in the category Quiv of quivers;
- Applying the free category functor $F : \mathsf{Quiv} \to \mathsf{Cat}$, the square remains a pushout, so we have a pushout

$$\begin{array}{ccc} FC_0^\delta & \longrightarrow & FQ \\ \downarrow & & \downarrow \\ FUC & \longrightarrow & FUC +_{FC_0^\delta} FQ \end{array} \tag{5.4}$$

  in Cat;
- compose with the counit ϵ of the adjunction $F \dashv U$ above:

$$FUC +_{FC_0^\delta} FQ \longrightarrow C +_{FC_0^\delta} FQ \tag{5.5}$$

The category $C \wr Q$ is the *FP collage* of Q, C.

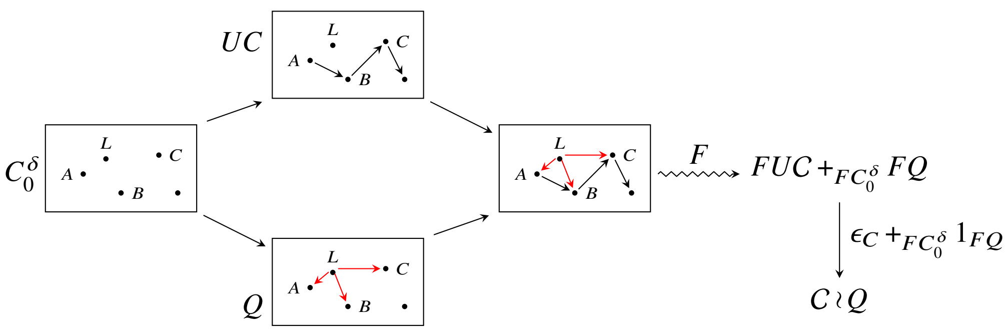


Figure 2: A graphical description of $C \wr Q$, with the pushout of quivers made explicit.

Remark 5.4. Note that there exists a canonical functor

$$K : C \longrightarrow C + FQ \longrightarrow C \wr Q \tag{5.6}$$

given by the coproduct embedding followed by the projection on the quotient realising the pushout $C \wr Q$; note that by construction this functor is the identity on objects, so it is induced in a canonical way by a monad $\mathfrak{q} : C \longrightarrow\!\!\!\!\!\!\mapstochar\;\; C$ on C in the category of profunctors, and K corresponds to the free functor into the Kleisli category of $\mathfrak{q}$.

Remark 5.5. A more general construction for $C \wr Q$ is then the following: fix a monad $\mathfrak{q}$ as above, and consider its Kleisli object $\bar{C}$; the free part of the Kleisli adjunction yields an identity on objects functor $C \to \bar{C}$. Given $\mathfrak{q}$, it can be highly complicated to describe its Kleisli category; what makes this construction combinatorially tamer is that the structure we are adding through the quiver Q is free.

Now we finally have all the needed tools to define vocabulary acquisition via paraphrasis satisfactorily.

Definition 5.6 (Vocabulary acquisition by paraphrasis)**.** Consider two speakers $\left[\begin{smallmatrix} & \mathcal{E}^p \\ p^\sharp & \downarrow \\ & \mathcal{L}\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} & \mathcal{E}^q \\ q^\sharp & \downarrow \\ & \mathcal{L}\end{smallmatrix}\right]$, which we will call *teacher* and *learner*, respectively.

Suppose that, for some $L \in \mathcal{L}$ –called *the linguistic element to learn*– we have that $\mathcal{E}^p_L \neq \emptyset$ and $\mathcal{E}^q_L = \emptyset$. Let $D_L : \mathcal{A} \to \mathcal{L}$ be an explanation of L according to p. Define the category $\mathcal{F}^q$ as $\mathcal{L} \wr Q$, where Q is the quiver obtained as follows:

- the vertices are the same of $\mathcal{L}$;
- there is an edge $L \to L'$ for each limiting cone leg $\lim\left(\mathcal{A}^{\mathrm{op}} \to \mathcal{L}^{\mathrm{op}} \xrightarrow{\nabla q^\sharp} \mathsf{Set}\right) \to \nabla q^\sharp L'$.

Now define a functor $T : \mathcal{F}^q \to \mathsf{Set}$ by mapping L to $\hat{L}^q$ and every other L' to $\nabla q^\sharp(L')$. On morphisms, T agrees with $\nabla q^\sharp$ wherever the latter is defined and maps the newly added edges of Q to the legs of the limit $\hat{L}^q$.

Using Theorem 2.3, the new fibration modelling q after learning L is $\int T$.

This definition is a bit terse and needs some unpacking, so let us piggyback to Example 5.1. In our situation, p knows at least partly what a *'cat'* (the object C) is, because the fibre $\mathcal{E}^p_C$ is not empty. We postulate that p can explain this concept in words, that is, p can define a functor D_C picking a bunch of linguistic elements in the language that mean *'cat'* to p since, by Definition 4.2, the limit $\hat{C}^p$ of $\nabla p^\sharp \circ D_C$ is a subset of $\nabla p^\sharp C$, which corresponds exactly to $\mathcal{E}^p_C$ via Theorem 2.3.

The teacher p would like to 'transmit' $\hat{C}^p$ to q, but this is not possible since, unless we postulate either of them is from the planet Vulcan, the only way p and q have to communicate is through the language $\mathcal{L}$. Still, p can utter the explanation and thus share the functor D_C with q. Notice how, at this stage, D_C is most likely *not* an explanation for q as $\nabla q^\sharp C = \mathcal{E}^q_C$ is empty by definition.

In any case, q can calculate the limit $\hat{C}^q$ of $\nabla q^\sharp \circ D_C$. The limits $\hat{C}^q$ and $\hat{C}^p$ will, in general, be different, as the same explanation makes sense to different speakers in different ways. Notice that whereas $\hat{C}^p$ is always non empty as D_C is defined to be an explanation for p, $\hat{C}^q$ can be empty. This happens when q is not able to successfully combine the meanings of the words in the explanation D_C: the explanation does not make sense to q. Interestingly, this is the case for the tautological explanation of Remark 4.4, which captures perfectly the meaning of C for p, but means absolutely nothing to q.

Whenever $\hat{C}^q$ is not empty, this is nothing more than a combination of concepts that q already knows, as illustrated in Figure 1. q includes this composition of concepts in the fibre over L, while the morphisms from the limit to its atomic constituents are included as morphisms between the fibres. In this procedure, the underlying language category for q changes, as we now have a fibration over the category $\mathcal{F}^q$, which is obtained by adding new morphisms to the language category $\mathcal{L}$. This is not a bug but a feature: in learning the meaning of a new concept, the speaker q also learns new ways to turn words and sentences into others.

Remark 5.7. The language game in Example 3.3 is taken up and debated later in [35] with a question: should we interpret the request '*Slab!*' as a word or a sentence? In the latter case, the sentence '*Slab!*' should be understood as a shortening of the sentence '*Bring me a slab!*'. Concerning this sentence, we can consider the roles as inverted with respect to Example 3.3: the builder makes a request, and the assistant can use Definition 5.6 to build a meaning for the sentence '*Bring me a slab!*'. Hopefully, this meaning will include things contextually relevant to the situation, allowing the assistant to identify the slab and fulfil the builder's request correctly.

Yet, this does not answer Wittgenstein's question, namely how we manage to go from '*Slab!*' to '*Bring me a slab!*'. Wittgenstein argues that the reason behind this semantic identification is the fact that these two sentences admit the same *contextual use*: there is no need for an explicit explanation, as context is directly responsible of disambiguating this sentence.

...But what is context mathematically? This is quite a thorny question. One possible solution to model context in our framework is using stronger models for the language category $\mathcal{L}$. In many examples, $\mathcal{L}$ is taken to be a pregroup, which corresponds to a context-free grammar in the sense of Chomsky [4]. We could instead make use of models of grammar that are *context-sensitive*, allowing for a finer degree of context management. In theory, we could have grammars where '*Slab!*' can be reduced to '*Bring me a slab!*' in a given context, and proceed as specified in Remark 5.7.

Yet, if there is something that Definition 5.6 taught us, it is that the process of acquiring vocabulary can result in enriching language with semantic meaning. Following the same idea, we could start with a model of $\mathcal{L}$ that is context-free, such as a pregroup, and gradually adding 'context-sensitive' morphisms that we borrow from the semantics, exactly as in Definition 5.6. Going back to Remark 5.7, the reduction from '*Slab!*' to '*Bring me a slab!*' would be added to $\mathcal{L}$ as a result of a language game previously played.

This last consideration hints at the fact that we shall be able to define disambiguation, and more in general communication, fibrationally. This is a broad topic, and a current matter of investigation.

6 Conclusion and future work

In this work, we used the framework provided by FibLang to take a first stab at describing vocabulary acquisition mathematically. In particular, we defined the concept of *explanation* to define, in turn, vocabulary acquisition by paraphrasis. A clear direction of future work is to use Definition 4.2 as the building block for a more general theory of communication: intuitively, speakers communicate in a game-theoretic fashion, exchanging explanations and using them to build meanings until they reach some kind of fixed point, at which providing further explanations does not result in building further meaning. This broader picture would fully capture Wittgenstein's ideas regarding language games, and we consider it an ambitious goal. With respect to this, a conjecture we are currently working on is to show how the bidirectional, game-theoretic nature of interactions between two speakers naturally suggests the use of self-dual categorical structures.

On a more practical standpoint, we would like to investigate a possible replacement of the limit in Definition 4.2 and in Definition 5.6 with something more linguistically sound: it is true that limits are universal constructions in category theory, and, as such, are mathematically well-behaved. Yet, in strictly applied contexts it may be useful to experiment with alternative definitions: for instance, considering a model of meaning where concepts in the fibres are images may be a sensible choice to experiment with machine-learning algorithms that merge images together [13].

References

[1] G. E. Barton, R. C. Berwick, and E. S. Ristad. *Computational Complexity and Natural Language*. Cambridge, MA, USA: MIT Press, 1987 (cit. on p. 229).

[2] J. Bénabou. "Fibered categories and the foundations of naive category theory". In: *The Journal of Symbolic Logic* 50.1 (1985), pp. 10–37 (cit. on p. 225).

[3] H. D. Brown et al. *Principles of language learning and teaching*. Vol. 4. longman New York, 2000 (cit. on p. 224).

[4] W. Buszkowski. "Lambek Grammars Based on Pregroups". In: *Proceedings of the 4th International Conference on Logical Aspects of Computational Linguistics* (2001), pp. 95–109 (cit. on p. 234).

[5] S. Chakraborty. "Language Acquisition: Seeing through Wittgenstein". In: *Indian Philosophical Quarterly* 45.2-3 (2018) (cit. on p. 225).
[6] N. Chomsky et al. *Language and mind*. Cambridge University Press, 2006 (cit. on p. 224).
[7] B. Coecke, M. Sadrzadeh, and S. Clark. "Mathematical Foundations for a Compositional Distributional Model of Meaning". In: *arXiv:1003.4394 preprint* (2010) (cit. on p. 225).
[8] B. Coecke. "Compositionality as we see it, everywhere around us". In: *arXiv:2110.05327 preprint* (2021). DOI: 10.48550/arXiv.2110.05327 (cit. on p. 225).
[9] B. Coecke and V. Wang. "Grammar equations". In: *arXiv:2106.07485 preprint* (2021) (cit. on p. 231).
[10] R. Ellis. *Understanding second language acquisition*. Vol. 31. Oxford university press Oxford, 1989 (cit. on p. 224).
[11] D. Everett. "Monolingual field research". In: *Linguistic fieldwork* (2001), pp. 166–188 (cit. on p. 227).
[12] P. Gardenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004 (cit. on pp. 225, 227).
[13] L. Gatys, A. Ecker, and M. Bethge. "A Neural Algorithm of Artistic Style". In: *Journal of Vision* 16.12 (2016), pp. 326–326. DOI: 10.1167/16.12.326 (cit. on p. 234).
[14] F. Genovese, F. Loregian, and C. Puca. *Fibrational linguistics I: first concepts*. 2022. arXiv: 2201.01136 [math.CT] (cit. on pp. 225–227).
[15] J. Hedges and M. Lewis. "Towards functorial language-games". In: *Proceedings CAPNS 2018* (2018). DOI: 10.4204/EPTCS.283.7 (cit. on p. 225).
[16] H. Ipek. "Comparing and Contrasting First and Second Language Acquisition: Implications for Language Teachers." In: *English Language Teaching* 2.2 (2009), pp. 155–163 (cit. on p. 224).
[17] B. Jacobs. *Categorical Logic and Type Theory*. SLFM 141. Elsevier, 1999 (cit. on pp. 225, 226).
[18] S. Krashen. "Theory versus practice in language training". In: *Enriching ESOL pedagogy*. Routledge, 2002, pp. 235–252 (cit. on p. 224).
[19] S. Lack. "Note on the Construction of Free Monoids". In: *Applied Categorical Structures* 18.1 (2010), pp. 17–29. DOI: 10.1007/s10485-008-9167-y (cit. on p. 231).
[20] J. Lambek. "Categorial and Categorical Grammars". In: *Categorial grammars and natural language structures*. Springer, pp. 297–317 (cit. on pp. 227, 230).
[21] J. Lambek. "The mathematics of sentence structure". In: *The American Mathematical Monthly* 65.3 (1958), pp. 154–170. DOI: 10.2307/2310058 (cit. on p. 225).
[22] E. Leivada and M. Westergaard. "Acceptable Ungrammatical Sentences, Unacceptable Grammatical Sentences, and the Role of the Cognitive Parser". In: *Frontiers in Psychology* 11 (2020), p. 364. DOI: 10.3389/fpsyg.2020.00364 (cit. on p. 229).
[23] J. Locke. *An essay concerning human understanding*. Kay & Troutman, 1847 (cit. on p. 224).
[24] A. Long, ed. *Plato: Meno and Phaedo*. Cambridge Texts in the History of Philosophy. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511780653 (cit. on p. 224).
[25] F. Loregian and E. Riehl. "Categorical notions of fibration". In: *Expositiones Mathematicae* (2019) (cit. on p. 225).
[26] M. G. McKeown and M. E. Curtis. *The nature of vocabulary acquisition*. Psychology Press, 2014 (cit. on p. 227).
[27] B. McLaughlin. "Theories of second-language learning". In: (1987) (cit. on p. 224).
[28] P. Newman and M. Ratliff. *Linguistic Fieldwork*. ERIC, 2001 (cit. on p. 227).
[29] G. A. Phillips. "Ludwig Wittgenstein: A Philosophical Theory of Language Acquisition and Use". In: *WORD* 27.1-3 (1971), pp. 139–157. DOI: 10.1080/00437956.1971.11435619 (cit. on p. 224).
[30] R. Street. "Fibrations in bicategories". In: *Cahiers de topologie et géométrie différentielle catégoriques* 21.2 (1980), pp. 111–160 (cit. on p. 225).
[31] R. Street and R. F. C. Walters. "The comprehensive factorization of a functor". In: *Bulletin of the American Mathematical Society* 79.5 (1973), pp. 936 –941. DOI: bams/1183534973 (cit. on p. 227).
[32] A. Vistoli. "Grothendieck topologies, fibered categories and descent theory". In: *Fundamental algebraic geometry*. Vol. 123. Math. Surveys Monogr. Providence, RI: Amer. Math. Soc., 2005, pp. 1–104 (cit. on p. 225).
[33] L. Wittgenstein. *The Blue and Brown Books: Preliminary Studies for the 'Philosophical Investigations'*. Oxford, England: Harper & Row, 1958 (cit. on p. 224).

[34] L. Wittgenstein. *Philosophical Investigations*. Ed. by G. Anscombe, P. Hacker, and J. Schulte. 4th. Chicester, UK: Wiley-Blackwell, 2009. 592 pp. (cit. on p. 224).

[35] L. Wittgenstein. *Philosophical Investigations*. John Wiley & Sons, 2010 (cit. on pp. 227, 228, 233).

Lax Liftings and Lax Distributive Laws

Ezra Schoen
University of Strathclyde
Glasgow, Scotland

Liftings of endofunctors on sets to endofunctors on relations are commonly used to capture bisimulation of coalgebras. Lax versions have been used in those cases where strict lifting fails to capture bisimilarity, as well as in modeling other notions of simulation. This paper provides tools for defining and manipulating lax liftings.

As a central result, we define a notion of a lax distributive law of a functor over the powerset monad, and show that there is an isomorphism between the lattice of lax liftings and the lattice of lax distributive laws.

We also study two functors in detail: (i) we show that the lifting for monotone bisimilarity is the minimal lifting for the monotone neighbourhood functor, and (ii) we show that the lattice of liftings for the (ordinary) neighbourhood functor is isomorphic to P(4), the powerset of a 4-element set.

1 Introduction

Coalgebras for an endofunctor are a general model of state-based transition systems. [9] Bisimulations are a central concept in the study of coalgebras, describing behavioral equivalence of states. Going back to [14], bisimulations of F-coalgebras in **Sets** have been defined as prefixed points of $\bar{F}$, the extension of F to **Rel**, the category of sets and relations.

One issue is that **Rel** places high demands on extensions: if $\widetilde{F} : \mathbf{Rel} \to \mathbf{Rel}$ is to be a strict functor that preserves the ordering of relations, and coincides with F on graphs of functions, then $\widetilde{F}$ only exists if F preserves weak pullbacks[5]; and if F preserves them, it is unique [4] and equal to the Barr lifting $\bar{F}$. [2] This situation is undesireable for two reasons:

- The elegant extension-based framework for bisimulation cannot be directly applied to coalgebras of type F when F does not preserve weak pullbacks. Neighbourhood-type functors are the most prominent example of such F.
- While the lifting $\bar{F}$ can be used to reason about bisimulation, other notions of simulation or equivalence of coalgebras cannot be expressed in the same way, since there are no other strict extensions.

To remedy this, various weaker notions of extension have been proposed.[18, 1, 8, 12]. Finding explicit examples has proceeded in a mostly ad-hoc fashion. The aim of this paper is to provide tools to reason about lax lifting in a more principled way. This paper is based on chapter 3 of the author's MSc thesis [16].

Our main contribution is a new notion of a *lax distributive law*, which we will show are in one-to-one correspondence with lax liftings. Distributive laws at their most general are simply natural transformations $FG \Rightarrow GF$ for two functors F, G. In most cases however, at least one of the two functors F and G is taken to be a monad, and the distributive law is required to interact 'nicely' with the monad structure.

The connection between liftings and distributive laws originates in [3], which focused on monad-monad interactions. Mulry [13] proved the equivalence between distributive laws of a functor F over a monad T and liftings of F to the Kleisli category of T.

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 237–253, doi:10.4204/EPTCS.380.14

More recently, some notions of 'weak distributive law' have been studied [17]; these, like Beck, pertain to monad-monad interaction, and involve weakening some of the conditions on Becks original distributive laws. Closer to the work in this paper are the lax distributive laws in [19], though again these focus on monad-monad interactions.

Aside from their connection to monads, distributive laws are of interest in their own right. They feature centrally in the bialgebraic approach to operational semantics. [20, 10] In the theory of automata, morphisms of distributive laws can provide various determinization procedures. [22]

We also analyse the liftings for two specific functors in detail:

- We prove that the minimal lifting for the monotone neighbourhood functor is given by the lifting $\widetilde{\mathscr{M}}$. This lifting has previously been used [15]; our result shows that $\widetilde{\mathscr{M}}$ is in some sense universal for $\mathscr{M}$.
- We give a complete description of the liftings for the ordinary neighbourhood functor. Equivalence notions between neighbourhood structures can be quite complex. [7] The classification in this paper shows that any notion of bisimulation between neighbourhood structures based on lax liftings will be almost trivial, since none of the 16 possible liftings make meaningful use of the input relation.

Outline

In section 2, we show that for a fixed functor, the lax liftings form a complete lattice. This implies that any functor admits a minimal, "maximally expressive" lifting. We show that for weak pullback-preserving functors, the minimal lifting coincides with the Barr lifting.

In section 3, we define lax distributive laws, and show that there is an isomorphism between the lattice of lax liftings, and the lattice of lax distributive laws. We also characterize those distributive laws that correspond to liftings that are symmetric and diagonal-preserving.

In section 4, we study the monotone and ordinary neighbourhood functors in more detail. For the monotone neighbourhood functor, we show that the known lifting $\widetilde{\mathscr{M}}$ is minimal. For the ordinary neighbourhood functor, we show that the lattice of liftings is isomorphic to $P(4)$ by giving an explicit description of all 16 liftings.

2 Preliminaries and basic properties

Definition 1. We write **Rel** for the category of sets and relations. The objects of **Rel** are sets, and a morphism $R \in \mathbf{Hom}_{\mathbf{Rel}}(X,Y)$ is given by a subset $R \subseteq X \times Y$.

Given two relations $R : X \multimap Y$ and $S : Y \multimap Z$, we write $R;S : X \multimap Z$ for their composition $R;S = \{(x,z) \in X \times Z \mid \exists y : xRySz\}$. Note that the order of composition is reversed from function composition.

Given a relation $R : X \multimap Y$, we write R° for its converse; that is,

$$R^\circ = \{(y,x) \mid (x,y) \in R\}$$

Given a function $f : X \to Y$, we write $\mathrm{gr}(f)$ for its *graph*, which is the relation

$$\mathrm{gr}(f) = \{(x,y) \mid f(x) = y\}$$

The category **Rel** is enriched over posets, where relations are ordered by inclusion. This makes **Rel** into a 2-category (in fact, it is the canonical example of an allegory). The operation $(-)^\circ$ is the morphism part of a functor $(-)^\circ : \mathbf{Rel} \to \mathbf{Rel}^{\mathrm{op}}$, which is an isomorphism of 2-categories. We write $\mathrm{gr}^\circ : \mathbf{Sets}^{\mathrm{op}} \to \mathbf{Rel}$ for the composition $(-)^\circ \circ \mathrm{gr}$.

Remark 2. The category **Rel** is isomorphic to the Kleisli category for the powerset monad. The assignment $f \mapsto \mathrm{gr}(f)$ is the morphism part of the left adjoint gr in the free-forgetful adjunction $\mathrm{gr} \dashv P$ that arises out of the Kleisli category construction. For a given relation $R : X \multimap Y$, we will write $\chi_R : X \to PY$ for the corresponding Kleisli morphism. Conversely, for a Kleisli morphism $f : X \to PY$, we will write $\lfloor f \rfloor : X \multimap Y$ for the corresponding relation.

The converse of a Kleisli morphism $f : X \to PY$ will be written as

$$f^\flat : Y \to PX : y \mapsto \{x \mid f(x) \ni y\}$$

Definition 3. Let $F : \mathbf{Sets} \to \mathbf{Sets}$ be a functor. A *(lax) F-lifting* is a lax 2-functor $L : \mathbf{Rel} \to \mathbf{Rel}$ such that

$$\begin{array}{ccc} \mathbf{Rel} & \xrightarrow{L} & \mathbf{Rel} \\ {\scriptstyle \mathrm{gr}}\uparrow & \leq & \uparrow{\scriptstyle \mathrm{gr}} \\ \mathbf{Sets} & \xrightarrow{F} & \mathbf{Sets} \end{array} \qquad \begin{array}{ccc} \mathbf{Rel} & \xrightarrow{L} & \mathbf{Rel} \\ {\scriptstyle \mathrm{gr}^\circ}\uparrow & \leq & \uparrow{\scriptstyle \mathrm{gr}^\circ} \\ \mathbf{Sets}^{\mathrm{op}} & \xrightarrow{F^{\mathrm{op}}} & \mathbf{Sets}^{\mathrm{op}} \end{array}$$

commute up to the indicated inequalities. A lifting L is called *symmetric* if

$$\begin{array}{ccc} \mathbf{Rel}^{\mathrm{op}} & \xrightarrow{L^{\mathrm{op}}} & \mathbf{Rel}^{\mathrm{op}} \\ {\scriptstyle (-)^\circ}\uparrow & & \uparrow{\scriptstyle (-)^\circ} \\ \mathbf{Rel} & \xrightarrow{L} & \mathbf{Rel} \end{array}$$

commutes; it is called *diagonal-preserving* if it strictly preserves identities.

Explicitly, we can expand the above into the following 5 conditions:

1. (2-cells) For all $R, S : X \multimap Y$, if $R \leq S$, then $LR \leq LS$.

2. (lax functoriality) For all $R : X \multimap Y$ and $S : Y \multimap Z$, we have $LR;LS \leq L(R;S)$.

3. (lifting) For all $f : X \to Y$, we have

$$\mathrm{gr}(Ff) \leq L\,\mathrm{gr}(f), \qquad \mathrm{gr}^\circ(Ff) \leq (L\,\mathrm{gr}(f))^\circ$$

4. (diagonal-preserving) For all X, we have $L\Delta_X \leq \Delta_{FX}$.

5. (symmetry) For all $R : X \multimap Y$, we have

$$L(R^\circ) = (LR)^\circ$$

Remark 4. The above includes various notions of lifting that have been previously been studied. Some authors (e.g. [11]) have taken “lifting” to be synonymous with the Barr lifting (see below). The notion of “(weak) relator” in [1] and [18] strengthen condition 2 to strict functoriality (although [18] does not require monotonicity). The notion used in [12] is almost identical, the only difference being that they require symmetry.

We give some examples:

Example 5. (5.1) For all functors $F : \mathbf{Sets} \to \mathbf{Sets}$, there is the lifting $F_\top : \mathbf{Rel} \to \mathbf{Rel}$ given by

$$F_\top(R : X \multimap Y) = FX \times FY$$

This lifting is symmetric, but does not preserve diagonals unless $|FX| \leq 1$ for all X.

(5.2) Any relation $R : X \multimap Y$ is presented as a span $R = \mathrm{gr}^\circ(\pi_1^R);\mathrm{gr}(\pi_2^R)$ by the two projection functions $\pi_1^R : R \to X$ and $\pi_2^R : R \to Y$. This motivates the definition

$$\bar{F}X = \mathrm{gr}^\circ(F\pi_1);\mathrm{gr}(F\pi_2)$$

$\bar{F}$ is known as the *Barr lifting*; it originates in [2]. In general, $\bar{F}$ is not lax but oplax, meaning $LR;LS \geq L(R;S)$. However, if F preserves weak pullbacks, then $\bar{F}$ is a strict functor which strictly preserves graphs and converse graphs.[11] Since the diagonal is the graph of the identity, $\bar{F}$ also preserves diagonals.

(5.3) The *Neighborhood functor* is defined to be the functor $\mathscr{N} = PP$. The action on a morphism $f : X \to Y$ is given by

$$(\mathscr{N} f)U = \{v \mid f^{-1}(v) \in U\}$$

The *Monotone neighborhood functor* is the subfunctor $\mathscr{M}$ of $\mathscr{N}$ defined by

$$\mathscr{M}X = \{U \in \mathscr{N}X \mid u \in U \text{ and } u \subseteq u' \implies u' \in U\}$$

One lifting for the monotone neighbood functor is given by

$$\begin{aligned}\widetilde{\mathscr{M}}(R : X \multimap Y) = \{(U,V) \mid & \forall u \in U \exists v \in V : \forall y \in v \exists x \in u : xRy \\ & \text{and } \forall v \in V \exists u \in U : \forall x \in u \exists y \in v : xRy\}\end{aligned}$$

This lifting originates in [15] where it was used to prove uniform interpolation for monotone modal logic. A closely related notion of bisimulation appeared earlier in [6].

We also state a simple lemma on lax liftings:

Lemma 6. *Let L be an F-lifting. For all relations $R : X \multimap Y$ and all functions $f : X' \to X$ and $g : Y' \to Y$, we have*

$$L(\mathrm{gr}(f);R;\mathrm{gr}^\circ(g)) = \mathrm{gr}(Ff);LR;\mathrm{gr}^\circ(Fg)$$

This is lemma 3.10(iii) in [16].

For a given functor $F : \mathbf{Sets} \to \mathbf{Sets}$, write $\mathbf{Lift}(F) = \{L : \mathbf{Rel} \to \mathbf{Rel} \mid L \text{ is an } F\text{-lifting}\}$. Liftings are naturally ordered pointwise: we say $L \leq L'$ if and only if for all R, we have $LR \leq L'R$.

Theorem 7. *Fix a functor $F : \mathbf{Sets} \to \mathbf{Sets}$. The class $\mathbf{Lift}(F)$ forms a complete lattice, with meets given by*

$$\left(\bigwedge_{i \in I} L_i\right) R := \bigcap_{i \in I} (L_i R)$$

Proof. See appendix. ☐

Since complete lattices have a minimal element, we get the following corollary:

Corollary 8. *Every endofunctor on* **Sets** *admits a minimal lifting.*

The significance of this corollary is the following: each lifting gives rise to a corresponding notion of simulation of coalgebras, as well as a modal logic. If for two liftings L, L' we have $L \leq L'$, then L-simulation distinguishes more states than L'-simulation, and L-logic is more expressive than L'-logic. A minimal lifting hence induces a maximally discerning notion of (bi)simulation, and a maximally expressive logic (among those that arise from lax liftings). [16]

In case F is weak pullback-preserving, we have an explicit description of its minimal lifting.

Proposition 9. *Let F :* **Sets** $\to$ **Sets** *be weak pullback-preserving. Then $\bar{F}$ is minimal among the F-liftings.*

Proof. Let L be a lifting for F. Then let $R : X \multimap Y$ be a relation. We know that R is presented as a span $R = \mathrm{gr}^\circ(\pi_X^R);\mathrm{gr}(\pi_Y^R)$ with $\pi_X^R : R \to X$ and $\pi_Y^R : R \to Y$ being the projection functions. So,

$$LR = L(\mathrm{gr}^\circ(\pi_X^R);\mathrm{gr}(\pi_Y^R)) \geq L(\mathrm{gr}^\circ(\pi_X^R));L(\mathrm{gr}(\pi_Y^R)) \geq \mathrm{gr}^\circ(F\pi_X^R);\mathrm{gr}(F\pi_Y^R) = \bar{F}R$$

□

There is also a natural involution on liftings, induced by $(-)^\circ$:

Definition 10. For an F-lifting L, we define the lifting $L^\sim$ as

$$L^\sim(R) := (L(R^\circ))^\circ$$

It is simple to prove that $L^\sim$ is a lifting when L is. [16]

Natural transformations between functors also induce a map between the associated liftings:

Theorem 11. *Let F, G :* **Sets** $\to$ **Sets** *be functors, and let $\eta : F \Rightarrow G$ be a natural transformation.*

(i) For every G-lifting L, the assignment

$$R \mapsto \{(x,y) \in FX \times FY \mid (\eta(x),\eta(y)) \in LR\}$$

constitutes an F-lifting $\eta^* L$.

(ii) η^* preserves arbitrary meets and $(-)^\sim$.

(iii) If L is symmetric, so is $\eta^* L$.

(iv) If η is everywhere injective, then if L preserves diagonals, so does $\eta^* L$.

Note that joins are not preserved in general: in particular, the minimal lifting is rarely preserved by η^*.

Proof. See appendix. □

From point (iv), together with the fact that the Barr lifting always preserves diagonals, we immediately get the following result:

Corollary 12. *All subfunctors of a weak pullback-preserving functor admit a diagonal-preserving lifting.*

This motivates the following conjecture:

Conjecture 13. The converse of the above: if F has a diagonal-preserving lifting, it can be embedded in a weak pullback-preserving functor.

3 Lax distributive laws

In this section, we give an alternative characterization of relation lifting in terms of distributive laws. We will write $\mu : P^2 \to P$ and $\eta : \mathrm{id} \to P$ for respectively the multiplication and unit of the powerset monad.

Definition 14. Let F : **Sets** $\to$ **Sets** be any functor. A *lax distributive law for F* is a collection of maps $\lambda_- : FP(-) \to PF(-)$, satisfying:

(Monotonicity) For any two functions $f, g : X \to PY$, if $f \leq g$, then

$$\lambda_Y \circ Ff \leq \lambda_Y \circ Fg$$

(Lax naturality) For any function $f : X \to PY$, we have

$$PFf \circ \lambda_X \leq \lambda_{PY} \circ FPf$$

(Lax Eilenberg-Moore) For any Z, we have

$$\mu_{FZ} \circ P\lambda_Z \circ \lambda_{PZ} \leq \lambda_Z \circ F\mu_Z \text{ and } \lambda_Z \circ F\eta_Z \geq \eta_{FZ}$$

There are also the optional properties

(Lax extensionality) For any Z,

$$\lambda_Z \circ F\eta_Z \leq \eta_{FZ}$$

(Symmetry) For any map $f : X \to PY$,

$$(\lambda_Y \circ Ff)^\flat = \lambda_X \circ F(f^\flat)$$

Definition 15. Let $\lambda : FP \rightsquigarrow PF$ be a lax distributive law. For a given relation $R : X \multimap Y$, we define $L^\lambda R$ as $L^\lambda R := \lfloor \lambda_Y \circ F\chi_R \rfloor$.

Conversely, for a lax lifting L of F, we define $\lambda^L : FP \rightsquigarrow PF$ as $\lambda^L := \chi_{L\ni}$.

The main theorem of this section states that these operations describe a bijective correspondence between lax liftings and lax distributive laws.

Theorem 16. *Let F :* **Sets** $\to$ **Sets** *be a functor.*

(i) The operations $L \mapsto \lambda^L$ and $\lambda \mapsto L^\lambda$ are inverse to each other.

(ii) If L is a F-lifting, then λ^L is a lax distributive law. Moreover, if L preserves diagonals then λ^L is laxly extensional, and if L is symmetric, then λ^L is symmetric.

(iii) If λ is a lax distributive law, then L^λ is a F-lifting. Moreover, if λ is laxly extensional, then L^λ preserves diagonals, and if λ is symmetric, then L^λ is symmetric.

Proof. (i) We calculate

$$\lfloor \lambda_Z^{L^\lambda} \rfloor = L^\lambda(\ni_Z) = \lfloor \lambda_Z \circ F\chi_\ni \rfloor = \lfloor \lambda_Z \circ F\,\mathrm{id}_{PZ} \rfloor = \lfloor \lambda_Z \rfloor$$

showing $\lambda^{L^\lambda} = \lambda$.

For the other equality, we get

$$
\begin{aligned}
L^{\lambda^L}(R) &= \lfloor \lambda^L \circ F\chi_R \rfloor \\
&= \lfloor \chi_{L\ni} \circ F\chi_R \rfloor \\
&\overset{*}{=} \lfloor \mu \circ P\chi_{L\ni} \circ \eta \circ F\chi_R \rfloor \\
&\overset{**}{=} \lfloor \eta \circ F\chi_R \rfloor ; \lfloor \chi_{L\ni} \rfloor \\
&= \mathrm{gr}(F\chi_R); L\ni \\
&= L(\mathrm{gr}(\chi_R);\ni) && \text{by lemma 6} \\
&\overset{***}{=} LR
\end{aligned}
$$

where in (*), we use one of the unit laws for monads, in (**) we use that $\lfloor - \rfloor$ turns Kleisli composition into relational composition, and in (***) we use the (easily verified) identity $\mathrm{gr}(\chi_R);\ni = R$.

(ii) We check the conditions in order.

(Monotonicity) We see that

$$\lfloor \lambda^L_Y \circ Ff \rfloor = \lfloor \lambda^L_Y \circ F\chi_{\lfloor f \rfloor} \rfloor = L^{\lambda^L}(\lfloor f \rfloor) = L(\lfloor f \rfloor)$$

where we use point (i) for the final equality. Now monotonicity of λ^L follows immediately from monotonicity of L.

(Lax naturality) Note that

$$(\ni_X; \mathrm{gr}(f)) \subseteq (\mathrm{gr}(Pf); \ni_{PY})$$

since if $A \ni x$, then $Pf[A] \ni f(x)$.

Now we see that

$$
\begin{aligned}
a \in PFf \circ \lambda^L_X(\Phi) &\iff \exists a' : a = Ff(a') \text{ and } a' \in \lambda^L_X(\Phi) \\
&\iff \exists a' : a = Ff(a') \text{ and } a' \in \chi_{L\ni}(\Phi) \\
&\iff \exists a' : a = Ff(a') \text{ and } (\Phi, a') \in L(\ni_X) \\
&\iff (\Phi, a) \in L(\ni_X); \mathrm{gr}(Ff) \\
&\implies (\Phi, a) \in L(\ni_X; \mathrm{gr}(f)) \\
&\implies (\Phi, a) \in L(\mathrm{gr}(Pf); \ni_{PY}) \\
&\iff (\Phi, a) \in \mathrm{gr}(FPf); L(\ni_{PY}) && \text{by lemma 6} \\
&\iff a \in \lambda^L_{PY} \circ FPf(\Phi)
\end{aligned}
$$

(Lax Eilenberg-Moore) First, we write out that

$$\mu \circ P\lambda^L_Z \circ \lambda^L_{PZ} = \mu \circ P(\chi_{L\ni}) \circ \chi_{L\ni} = \chi_{L\ni;L\ni}$$

since χ_{-} turns relational composition ; into Kleisli composition. Next, note that

$$\mathrm{gr}(\mu);\ni = \ni;\ni$$

since

$$\bigcup_{A \in \mathscr{A}} A \ni x \text{ if and only if } \exists A : \mathscr{A} \ni A \text{ and } A \ni x$$

So, we conclude that

$$\begin{aligned}
\lfloor \mu \circ P\lambda^L_Z \circ \lambda^L_{PZ} \rfloor &= L\ni; L\ni \\
&\leq L(\ni;\ni) \\
&= L(\mathrm{gr}(\mu);\ni) \\
&= \mathrm{gr}(F\mu); L\ni && \text{by lemma 6} \\
&= \lfloor \eta \circ F\mu \rfloor; \lfloor \chi_{L\ni} \rfloor \\
&\stackrel{*}{=} \lfloor \mu \circ P\chi_{L\ni} \circ \eta \circ F\mu \rfloor \\
&= \lfloor \chi_{L\ni} \circ F\mu \rfloor \\
&= \lfloor \lambda^L_Z \circ F\mu \rfloor
\end{aligned}$$

giving the first inequality; where in the equality (*) we use that $\lfloor - \rfloor$ turns relational composition into Kleisli composition.

For the second inequality, we simply note that $\lfloor \eta_Z \rfloor = \mathrm{gr}(\mathrm{id}_Z)$, and so

$$\lfloor \lambda^L_Z \circ F\eta_Z \rfloor = L^{\lambda^L} \lfloor \eta_Z \rfloor = L\,\mathrm{gr}(\mathrm{id}_Z) \geq \mathrm{gr}(F\,\mathrm{id}_Z) = \mathrm{gr}(\mathrm{id}_{FZ}) = \lfloor \eta_{FZ} \rfloor$$

(Lax extensionality) Assume that L is diagonal-preserving. We aim to show that λ_L is laxly extensional. This follows simply from

$$\lfloor \lambda^L_Z \circ F\eta_Z \rfloor = L\,\mathrm{gr}(\mathrm{id}_Z) \leq \mathrm{gr}(\mathrm{id}_{FZ}) = \lfloor \eta_{FZ} \rfloor$$

(Symmetry) If L is symmetrical, we get simply

$$\lfloor (\lambda^L_Y \circ Ff)^\flat \rfloor = (L\lfloor f \rfloor)^\circ = L(\lfloor f \rfloor^\circ) = \lfloor \lambda_X \circ F(f^\flat) \rfloor$$

(iii) We prove each of the five conditions.

(2-cells) If $S \leq R$, then

$$L^\lambda S = \lfloor \lambda_Y \circ F\chi_S \rfloor \leq \lfloor \lambda_Y \circ F\chi_R \rfloor = L^\lambda R$$

by monotonicity of λ.

(lax functoriality) Let $R : X \multimap Y$ and $S : Y \multimap Z$. We draw the following diagram:

$$\begin{array}{ccccc}
FX & \xrightarrow{F(\chi_{R;S})} & FPZ & \xrightarrow{\lambda_Z} & PFZ \\
\downarrow{\scriptstyle F\chi_R} & = & \uparrow{\scriptstyle F\mu_Z} & \geq & \uparrow{\scriptstyle \mu_{FZ}} \\
FPY & \xrightarrow{FP\chi_S} & FPPZ & \xrightarrow{P\lambda_Z \circ \lambda_{PZ}} & PPFZ \\
\downarrow{\scriptstyle \lambda_Y} & \leq & \downarrow{\scriptstyle \lambda_{PZ}} & = & \| \\
PFY & \xrightarrow{PF\chi_S} & PFPZ & \xrightarrow{P\lambda_Z} & PPFZ
\end{array}$$

The top left square is F applied to the Kleisli composite $\chi_{R;S}$. The top right square is lax Eilenberg-Moore, and the bottom left square is lax naturality. The bottom right square is a simple equality.

The above diagram shows that

$$L^\lambda R; L^\lambda S = \lfloor \mu_{FZ} \circ P(\lambda_Z \circ F\chi_S) \circ \lambda_Y \circ F\chi_R \rfloor \leq \lfloor \lambda_Z \circ F(\chi_{R;S}) \rfloor = L^\lambda(R;S)$$

as desired.

(lifting) Let $f : X \to Y$ be a morphism. Then

$$L^\lambda \operatorname{gr}(f) = L^\lambda(\lfloor \eta_Y \circ f \rfloor) = \lfloor \lambda_Y \circ F(\eta_Y \circ f) \rfloor = \lfloor \lambda_Y \circ F\eta_Y \circ Ff \rfloor \geq \lfloor \eta_{TY} \circ Ff \rfloor = \operatorname{gr}(Ff)$$

by lax Eilenberg-Moore. We also have

$$\begin{aligned}
\operatorname{gr}(Ff); L^\lambda \operatorname{gr}^\circ(f) &= \lfloor \mu_X \circ P\lambda_X \circ PF(\chi_{\operatorname{gr}^\circ(f)}) \circ \eta_{FX} \circ Ff \rfloor \\
&= \lfloor \lambda_X \circ F(\chi_{\operatorname{gr}^\circ(f)}) \circ Ff \rfloor \\
&= \lfloor \lambda_X \circ F(\chi_{\operatorname{gr}^\circ(f)} \circ f) \rfloor \\
&\overset{*}{\geq} \lfloor \lambda_X \circ F\eta_X \rfloor \\
&\overset{**}{\geq} \lfloor \eta_{FX} \rfloor = \Delta_{FX}
\end{aligned}$$

where in inequality (*) we use monotonicity of λ, together with the fact that $\chi_{\operatorname{gr}^\circ(f)} \circ f \geq \eta_X$; and inequality (**) is simply the unit part of lax Eilenberg-Moore. Since $\operatorname{gr}^\circ(Ff)$ is the least relation R with $\operatorname{gr}(Ff); R \geq \Delta_X$, we obtain

$$L^\lambda \operatorname{gr}^\circ(f) \geq \operatorname{gr}^\circ(Ff)$$

as desired.

(diagonal-preserving) Assume that λ is laxly extensional. Then

$$L^\lambda \Delta_Z = \lfloor \lambda_Z \circ \eta_Z \rfloor \leq \lfloor \eta_{FZ} \rfloor = \Delta_{FZ}$$

(symmetry) Assume that λ is symmetric. Then it follows immediately that

$$L^\lambda(R^\circ) = \lfloor \lambda_X \circ F(\chi_R^\flat) \rfloor = \lfloor (\lambda_Y \circ F\chi_R)^\flat \rfloor = \lfloor \lambda_Y \circ F\chi_R \rfloor^\circ = (L^\lambda R)^\circ$$

□

4 Explicit descriptions

Since the class of F-liftings forms a complete lattice for each F, it follows that each F has a minimal lifting $\tilde{F}$. In the case of weak-pullback preserving F, we know that $\tilde{F} = \bar{F}$, the Barr lifting. However, for non-weak-pullback preserving functors, giving an explicit description of the minimal lifting involves a non-trivial amount of effort.

In this section, we will study the minimal liftings for the neighborhood functor and the monotone neighborhood functor. For the (ordinary) neighborhood functor, we moreover give a full description of the complete lattice of liftings.

4.1 Monotone neighborhood functor

Recall the lifting $\widetilde{\mathscr{M}}$ from example (5.3).

Theorem 17. *The lifting $\widetilde{\mathscr{M}}$ is the minimal lifting for the monotone neighborhood functor $\mathscr{M}$.*

To prove this, we first need a lemma.

Lemma 18. *Let $R : X \multimap Y$ be a total surjective relation. Then $\widetilde{\mathscr{M}}R \leq LR$ for all liftings L.*

In [6], a similar statement appears as lemma 4.7.

Proof. Consider the two projection morphisms $\pi_X : R \to X$ and $\pi_Y : R \to Y$. Since R is total and surjective, both these functions are surjective.

We claim that $\widetilde{\mathscr{M}}R = (\mathscr{M}\,\pi_X)^\circ ; \mathscr{M}\,\pi_Y$. The inequality $\geq$ follows from $R = (\pi_X)^\circ ; \pi_Y$.

For $\leq$, let $(U,V) \in \widetilde{\mathscr{M}}R$. Then we set

$$\begin{aligned} W_0 &:= \{\{(x,y) \in R \mid x \in u\} \mid u \in U\} \\ W_1 &:= \{\{(x,y) \in R \mid y \in V\} \mid v \in V\} \\ W &:= \{w \mid \exists w' \in W_0 \cup W_1 : w' \subseteq w\} \end{aligned}$$

We claim that $\mathscr{M}\,\pi_X(W) = U$. For this, we need to show that (1) if $u \in U$, then $\pi_X^{-1}(u) \in W$, and (2) if $\pi_X^{-1}(u) \in W$, then $u \in U$.

(1) Clearly, if $u \in U$, then $\pi_X^{-1}(u) = \{(x,y) \in R \mid x \in u\} \in W$, so $\pi_X^{-1}(u) \in W$.

(2) Assume $\pi_X^{-1}(u) \in W$. There are two cases: (i) there is a $u' \in U$ with $\{(x,y) \in R \mid x \in u'\} \subseteq \pi_X^{-1}(u)$, or (ii) there is a $v \in V$ with $\{(x,y) \in R \mid y \in v'\} \subseteq \pi_X^{-1}(u)$.

 (i) In this case, we know that $\pi_X[\{(x,y) \in R \mid x \in u'\}] \subseteq \pi_X(\pi_X^{-1}(u))$. But since R was total, we know that $\pi_X[\{(x,y) \in R \mid x \in u'\}] = u'$ and $\pi_X[\pi^{-1}(u)] = u$. So $u' \subseteq u$, and hence $u \in U$.

 (ii) Clearly, $\pi_X[\{(x,y) \in R \mid y \in v\}] = \{x \mid \exists y \in v : xRy\}$. Since $(U,V) \in \widetilde{\mathscr{M}}R$, there is a $u' \in U$ such that for all $x \in u'$, there is a $y \in v$ with xRy. But this just says that $u' \subseteq \pi_X[\{(x,y) \in R \mid y \in v\}]$. So we conclude that there is a $u' \in U$ with

$$u' \subseteq \pi_X[\{(x,y) \in R \mid y \in v\}] \subseteq \pi_X(\pi_X^{-1}(u)) = u$$

 and hence $u \in U$.

So in both cases, we have $u \in U$, as desired.

The proof that $\mathscr{M}\,\pi_Y(W) = V$ is completely symmetrical; so, we can conclude that $(U,V) \in (\mathscr{M}\,\pi_X)^\circ ; \mathscr{M}\,\pi_Y$.

Now, let L be any lifting. Then

$$LR = L((\pi_X)^\circ ; \pi_Y) \geq L(\pi_X)^\circ ; L\pi_Y \geq (\mathscr{M}\,\pi_X)^\circ ; \mathscr{M}\,\pi_Y = \widetilde{\mathscr{M}}R$$

□

With this lemma, we can prove theorem 17.

Proof. Let $R : X \multimap Y$ be any relation. Let X' be the domain of R and Y' the range of R. Then we define $X_* = X \cup \{*\}, Y_* = Y \cup \{*\}$ and

$$R_* = R \cup \{(x,*) \mid x \in X \setminus X'\} \cup \{(*,y) \mid y \in Y \setminus Y'\} \cup \{(*,*)\}$$

Then $R_* : X_* \multimap Y_*$ is total and surjective.

Let $\iota_X : X \to X_*$ and $\iota_Y : Y \to Y_*$ be the natural inclusion functions. First, we note that $R = \iota_X ; R_* ; (\iota_Y)^\circ$ The inequality $\leq$ is clear, since $R \subseteq R_*$. For $\geq$, notice that $*$ is not in the range of either ι_X or ι_Y.

Now by lemma 6, we know that for any lifting L,

$$LR = (\mathscr{M}\,\iota_X) ; LR_* ; (\mathscr{M}\,\iota_Y)^\circ.$$

So we can calculate that

$$\begin{aligned} LR &= \mathscr{M}\iota_X; LR_*; (\mathscr{M}\iota_Y)^\circ \\ &\geq \mathscr{M}\iota_X; \widetilde{\mathscr{M}}R_*; (\mathscr{M}\iota_Y)^\circ \qquad \text{by lemma 18} \\ &= \widetilde{\mathscr{M}}R \end{aligned}$$

We conclude that $\widetilde{\mathscr{M}}$ is minimal. □

4.2 The neighborhood functor

We introduce an extremely minimal logic for neighborhood systems. This will consist of the following expressions:

$$\begin{aligned} \rho_0 &::= \Box\bot \mid \neg\Box\bot \\ \rho_1 &::= \Box\top \mid \neg\Box\top \\ \rho &::= (\rho_0, \rho_1) \end{aligned}$$

Given $(U,V) \in \mathscr{N}X \times \mathscr{N}Y$, satisfaction $(U,V) \Vdash \rho$ is defined as follows:

$$\begin{aligned} &(U,V) \Vdash \Box\bot \text{ iff } \varnothing \in U \implies \varnothing \in V \\ &(U,V) \Vdash \neg\Box\bot \text{ iff } \varnothing \notin U \implies \varnothing \notin V \\ &(U,V) \Vdash \Box\top \text{ iff } X \in U \implies Y \in V \\ &(U,V) \Vdash \neg\Box\top \text{ iff } X \notin U \implies Y \notin V \\ &(U,V) \Vdash (\rho_0,\rho_1) \text{ iff } (U,V) \Vdash \rho_0 \text{ and } (U,V) \Vdash \rho_1 \end{aligned}$$

Now let I be the set of all ρ's. For each $J \subseteq I$, we get a lifting L_J defined on a relation $R : X \multimap Y$ as

$$L_J(R) := \{(U,V) \in \mathscr{N}X \times \mathscr{N}Y \mid (U,V) \Vdash \rho \text{ for all } \rho \in J\}$$

Since these liftings do not depend on the chosen relation, we will omit R, writing simply $L_J : \mathscr{N}X \multimap \mathscr{N}Y$. Note also that $J \supseteq J'$ if and only if $L_J \leq L_{J'}$.

Theorem 19. *The lattice $(P(I), \supseteq)$ is isomorphic to $(\mathbf{Lift}(\mathscr{N}), \leq)$ via $J \mapsto L_J$.*

To prove this theorem, we will need the following lemma:

Lemma 20. *Let $(U,V) \in \mathscr{N}X \times \mathscr{N}Y$ and $(U',V') \in \mathscr{N}X' \times \mathscr{N}Y'$. Assume that for some ρ, we have $(U,V) \nVdash \rho$ and $(U',V') \nVdash \rho$. Then for each ρ', we have*

$$(U,U') \Vdash \rho', \quad (V,V') \Vdash \rho'$$

Proof. WLOG, we can assume that $\rho = (\Box\bot, \Box\top)$; all other cases are similar.

Then since $(U,V) \nVdash \rho$, we know $\varnothing \in U, \varnothing \notin V$ and $X \in U, Y \notin V$. Similarly, we know $\varnothing \in U', \varnothing \notin V$ and $X' \in U', Y' \notin V'$. But from these data, it follows immediately that for all ρ', we must have

$$(U,U') \Vdash \rho'$$

since U and U' agree on $\varnothing$ and the entire set. And of course the same holds for (V,V'). □

Now we can start the full proof.

Proof. First, we show that each L_J is a lifting. Since clearly $L_J = \bigwedge_{\rho \in J} L_{\{\rho\}}$, it suffices to show that each $L_{\{\rho\}}$ is a lifting.

They are clearly monotonic, since they do not depend on the input R. They are also clearly laxly functorial. Finally, if $f : X \to Y$ is a function, then for all $U \in \mathscr{N} X$ and all $\rho \in I$, we have

$$(U, (\mathscr{N} f)U) \Vdash \rho$$

since

$$(\mathscr{N} f)U \ni \varnothing \text{ iff } U \ni f^{-1}(\varnothing) \text{ iff } U \ni \varnothing \text{ and } (\mathscr{N} f) \ni X \text{ iff } U \ni f^{-1}(X) \text{ iff } U \ni Y$$

So indeed, each L_J extends the graph of $\mathscr{N} f$.

This shows that the map $J \mapsto L_J$ is well-defined. It is clearly injective and meet-preserving (recall that the meet in $(P(I), \supseteq)$ is given by union), so it remains to show that it is surjective. We will proceed in three steps:

1. The top element is preserved by $J \mapsto L_J$;
2. The bottom element is preserved by $J \mapsto L_J$;
3. If $L > L_J$, then there is some $J' \subsetneq J$ with $L \geq L_{J'}$.

These three steps together imply that $J \mapsto L_J$ is surjective, from which it then follows that it is an isomorphism.

For point 1: The top element of $(P(I), \supseteq)$ is $\varnothing$, and indeed $L_\varnothing(R : X \multimap Y) = X \times Y$.

For point 2: Let L be any symmetric lifting for $\mathscr{N}$. For a given X, write $0_X : X \multimap X$ for the empty relation. We will show that $(U, V) \in L0_X$ if U and V agree on $\varnothing$ and X.

We first assume that X contains some point x_0. Write $2 = \{a, b\}$ for the generic two-element set; by abuse of notation, we may also consider $a, b : X \to 2$ and $x_0 : X \to X$ as constant maps.

Let $U \in \mathscr{N} X$ be a neighborhood system. There are four cases:

(i) $\varnothing \notin U, X \notin U$. Then we see that

$$\mathscr{N} a(U) = \varnothing = \mathscr{N} b(U)$$

since for constant maps $c : X \to Y$, we have $c^{-1}(A) = \varnothing$ or $c^{-1}(A) = X$ for all A. We also clearly have $\mathscr{N} b(\varnothing) = \varnothing$. So, we have

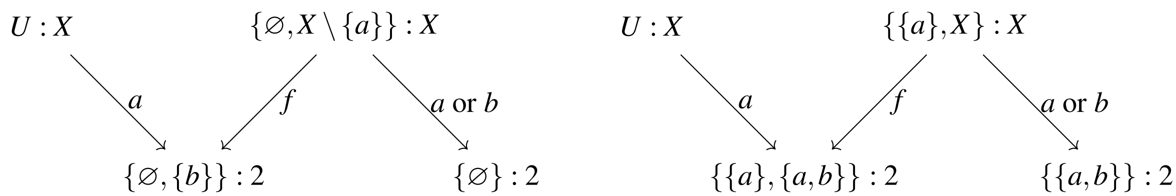
$$(U, \varnothing) \in \mathrm{gr}(\mathscr{N} a), (\varnothing, U) \in \mathrm{gr}^\circ(\mathscr{N} b)$$

for all U omitting $\varnothing$ and X. Now we have if U, V both omit $\varnothing$ and X, then

$$U(L\mathrm{gr}(a))\varnothing(L\mathrm{gr}^\circ(b))V$$

and hence

$$(U, V) \in L\mathrm{gr}(a); L\mathrm{gr}^\circ(b) \subseteq L0_X$$



(a) (b)

(ii) $\varnothing \notin U, X \in U$. Then $\mathscr{N}a(U) = \{\varnothing, \{b\}\}$. Take $f : X \to 2$ given by

$$f(x) = \begin{cases} b & x \neq x_0 \\ a & x = x_0 \end{cases}$$

Let $V = \{\varnothing, X \setminus \{x_0\}\}$. Then it is easily seen that $\mathscr{N}f(V) = \{\varnothing, \{b\}\}$. Finally, we have $\mathscr{N}a(V) = \mathscr{N}b(V) = \{\varnothing\}$, again by the remarks on inverse images along constant maps.

Now we have a 'zigzag' as in figure 1a. By tracing the definitions, we can see that $\mathrm{gr}(a);\mathrm{gr}^{\circ}(f) = \mathrm{gr}(x_0)$, and

$$\mathrm{gr}(a);\mathrm{gr}^{\circ}(f);\mathrm{gr}(a) = \mathrm{gr}(x_0);\mathrm{gr}(a) = \mathrm{gr}(a)$$

and similarly $\mathrm{gr}(a);\mathrm{gr}^{\circ}(f);\mathrm{gr}(b) = \mathrm{gr}(b)$. We now have that if U is such that $\varnothing \in U, X \notin U$, then

$$(U, \{\varnothing\}) \in L(\mathrm{gr}(a)), \text{ and } (\{\varnothing\}, U) \in L(\mathrm{gr}^{\circ}(b))$$

But now for all U, V which both contain $\varnothing$ and both omit X, we have

$$(U, V) \in L\mathrm{gr}(a); L(\mathrm{gr}^{\circ}(b)) \subseteq L(\mathrm{gr}(a);\mathrm{gr}^{\circ}(b)) = L0_X$$

(iii) Let U be such that $\varnothing \notin U, X \in U$. Then $\mathscr{N}a(U) = \{\{a\}, \{a, b\}\}$. Take $V = \{\{x_0\}, X\}$. Then with f as in point (ii), we have $\mathscr{N}f(V) = \{\{a\}, \{a, b\}\}$. Now for the constant maps a, b, we have $\mathscr{N}a(V) = \{\{a, b\}\} = \mathscr{N}b(V)$. Hence, we obtain a similar zigzag as in point (ii), as can be seen in figure 1b. From here, the argument is completely the same as in (ii): for U, V both omitting $\varnothing$ and both including X, we get

$$(U, \{\{a, b\}\}) \in L\mathrm{gr}(a), \qquad (\{\{a, b\}\}, V) \in L\mathrm{gr}^{\circ}(b)$$

showing that

$$(U, V) \in L(\mathrm{gr}(a); (\mathrm{gr}(b))^{\circ}) = L0_X$$

(iv) $\varnothing \in U, X \in U$. Then $\mathscr{N}a(U) = P2 = \mathscr{N}b(U)$, and so as in (i) we get for all U, V both including $\varnothing$ and X that

$$(U, P2) \in L\mathrm{gr}(a), \qquad (P2, V) \in L\mathrm{gr}^{\circ}(b)$$

and hence

$$(U, V) \in L(\mathrm{gr}(a);\mathrm{gr}^{\circ}(b)) = L0_X$$

Now we have that if X is nonempty, then $L0_X \supseteq L_I$. But of course, if X and Y are arbitrary, then the empty relation $0_{XY} : X \multimap Y$ factors through 0_{X+Y} via the inclusions $\iota_X : X \to X + Y, \iota_Y : Y \to X + Y$. From this, it follows easily that $L0_{XY} \supseteq L_I$. But now, for $R : X \multimap Y$ an arbitrary relation, we have that

$$LR \supseteq L0_{XY} \supseteq L_I$$

showing that L_I is minimal indeed.

For point 3: Let L be any lifting, and $J \subseteq I$ with $L > L_J$. Then there is some relation $R : X \multimap Y$ and some neighborhood systems $(U,V) \in \mathscr{N}X \times \mathscr{N}Y$ with $(U,V) \in LR$ and $(U,V) \nVdash \rho_0$ for some $\rho_0 \in J$.

We claim that now for $J' = J \setminus \{\rho_0\}$, we have $L \geq L_{J'}$. Again, we will show that $L0_{X'Y'} \geq L_{J'}$ for all X', Y'.

Now let $(U',V') \in L_{J'}$. There are two cases:

(i) $(U',V') \Vdash \rho_0$. Then $(U',V') \in L_J < L$, so $(U',V') \in L0_{X'Y'}$.

(ii) $(U',V') \nVdash \rho_0$. Since $(U,V) \nVdash \rho_0$ we know by lemma 20 that

$$(U',U) \in L_I, \quad (V,V') \in L_I$$

and hence

$$(U',V') \in L_I;LR;L_I \subseteq L0_{X'X};LR;L_{YY'} \subseteq L(0_{X'X};R;0_{YY'}) = L0_{X'Y'}$$

So indeed, $L0_{X'Y'} \geq L_{J'}$ and hence for arbitrary relations $R' : X' \multimap Y'$ we have

$$LR \geq L0_{X'Y'} \geq L_{J'}$$

as desired.

□

5 Conclusion and further research

We have shown that for a fixed functor, the lax liftings form a complete lattice. In particular, any functor admits a minimal, “maximally expressive” lifting. We have shown that for weak pullback-preserving functors, the least functor coincides with the Barr lifting.

We have defined lax distributive laws, and shown that there is an isomorphism between the lattice of lax liftings, and the lattice of lax distributive laws. We also characterized those distributive laws that correspond to liftings that are symmetric and diagonal-preserving.

We studied the monotone and ordinary neighbourhood functors in more detail. For the monotone neighbourhood functor, we have shown that the known lifting $\widetilde{\mathscr{M}}$ is minimal. For the ordinary neighbourhood functor, we have explicitly described all 16 liftings. This question was still open in [16].

The results in this paper are specific to the categories of **Sets** and 2-valued relations. Other kinds of liftings have been considered. For instance, in [21], liftings of fuzzy relations are defined. A natural direction of further research is to investigate if the results from this paper could be extended to cover a wider range of many-valued relations. More generally still, one can see **Sets** as the category of functions inside the allegory **Rel**. A possible approach would be to study liftings in the setting of arbitrary (power) allegories.

References

[1] Alexandru Baltag (2000): *A Logic for Coalgebraic Simulation*. *Electronic Notes in Theoretical Computer Science* 33, pp. 42–60, doi:10.1016/S1571-0661(05)80343-3. CMCS'2000, Coalgebraic Methods in Computer Science.

[2] Michael Barr (1970): *Relational algebras*. In S. MacLane, H. Applegate, M. Barr, B. Day, E. Dubuc, Phreilambud, A. Pultr, R. Street, M. Tierney & S. Swierczkowski, editors: *Reports of the Midwest Category Seminar IV*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 39–55, doi:10.1007/BFb0060439.

[3] Jon Beck (1969): *Distributive laws*. In B. Eckmann, editor: *Seminar on Triples and Categorical Homology Theory*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 119–140, doi:10.1007/BFb0083084.

[4] R. Bird & O. de Moor (1997): *Algebra of Programming*. Prentice-Hall international series in computer science, Prentice Hall. Available at `https://books.google.co.uk/books?id=P5NQAAAAMAAJ`.

[5] A. Carboni, G. M. Kelly & R. Wood (1991): *A 2-categorical approach to change of base and geometric morphisms I*. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* 32, pp. 47–95.

[6] Helle Hansen & Clemens Kupke (2004): *A Coalgebraic Perspective on Monotone Modal Logic*. *Electronic Notes in Theoretical Computer Science* 106, pp. 121–143, doi:10.1016/j.entcs.2004.02.028.

[7] Helle Hansen, Clemens Kupke & Eric Pacuit (2009): *Neighbourhood Structures: Bisimilarity and Basic Model Theory*. *Logical Methods in Computer Science* 5(2), doi:10.2168/lmcs-5(2:2)2009.

[8] Jesse Hughes & Bart Jacobs (2004): *Simulations in coalgebra*. *Theoretical Computer Science* 327(1), pp. 71–108, doi:10.1016/j.tcs.2004.07.022. Selected Papers of CMCS '03.

[9] Bart Jacobs (2016): *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, doi:10.1017/CBO9781316823187.

[10] Bartek Klin (2006): *Bialgebraic Methods in Structural Operational Semantics. Invited Talk*. *Electronic Notes in Theoretical Computer Science (ENTCS)* 175, pp. 33–43, doi:10.1016/j.entcs.2006.11.018.

[11] Alexander Kurz & Jiří Velebil (2016): *Relation lifting, a survey*. *Journal of Logical and Algebraic Methods in Programming* 85(4), pp. 475–499, doi:10.1016/j.jlamp.2015.08.002. Relational and algebraic methods in computer science.

[12] Johannes Marti & Yde Venema (2015): *Lax extensions of coalgebra functors and their logic*. *Journal of Computer and System Sciences* 81(5), pp. 880–900, doi:10.1016/j.jcss.2014.12.006. 11th International Workshop on Coalgebraic Methods in Computer Science, CMCS 2012 (Selected Papers).

[13] Philip S. Mulry (1994): *Lifting theorems for Kleisli categories*. In Stephen Brookes, Michael Main, Austin Melton, Michael Mislove & David Schmidt, editors: *Mathematical Foundations of Programming Semantics*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 304–319, doi:10.1007/3-540-58027-1˙15.

[14] J.J.M.M. Rutten (1998): *Relators and Metric Bisimulations: (Extended Abstract)*. *Electronic Notes in Theoretical Computer Science* 11, pp. 252–258, doi:10.1016/S1571-0661(04)00063-5. CMCS '98, First Workshop on Coalgebraic Methods in Computer Science.

[15] Luigi Santocanale & Yde Venema (2010): *Uniform Interpolation for Monotone Modal Logic*. In: *Advances in Modal Logic*.

[16] Ezra Schoen (2021): *Relation Lifting and Coalgebraic Logic*. Master's thesis, University of Amsterdam.

[17] Ross Street (2019): *Weak distributive laws*. *Theory and Applications of Categories* 22, pp. 313–320.

[18] Albert Marchienus Thijs (1996): *Simulation and fixpoint semantics*. Rijksuniversiteit Groningen.

[19] Walter Tholen (2016): *Lax Distributive Laws for Topology, I*, doi:10.48550/ARXIV.1603.06251. Available at `https://arxiv.org/abs/1603.06251`.

[20] Daniele Turi & Gordon D. Plotkin (1997): *Towards a mathematical operational semantics*, doi:10.1109/LICS.1997.614955. *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pp. 280–291.

[21] Paul Wild & Lutz Schröder (2020): *Characteristic Logics for Behavioural Hemimetrics via Fuzzy Lax Extensions*, doi:10.48550/ARXIV.2007.01033. Available at `https://arxiv.org/abs/2007.01033`.

[22] Stefan Zetzsche, Gerco van Heerdt, Matteo Sammartino & Alexandra Silva (2021): *Canonical Automata via Distributive Law Homomorphisms*. *Electronic Proceedings in Theoretical Computer Science* 351, pp. 296–313, doi:10.4204/eptcs.351.18.

A Additional proofs

Proof of theorem 7. We show that $\mathbf{Lift}(F)$ has all meets. Let $\{L_i \mid i \in I\}$ be any collection of F-liftings. For a given $R : X \multimap Y$, set

$$LR = \bigcap_{i\in I} L_i R$$

We show that L is again a lifting, by showing it satisfies conditions 1, 2 and 3.

(1) If $R \le S$, then

$$LR = \bigcap_{i\in I} L_i R \le \bigcap_{i\in I} L_i S = LS$$

(2) If $R : X \multimap Y$ and $S : Y \multimap Z$ are relations, then

$$\begin{aligned} LR;LS &= \left(\bigcap_{i\in I} L_i R\right) ; \left(\bigcap_{i\in I} L_i S\right) \\ &\le \bigcap_{i\in I}\bigcap_{j\in I} L_i R; L_j S \\ &\le \bigcap_{i\in I} L_i R; L_i S \\ &\le \bigcap_{i\in I} L_i (R;S) \\ &= L(R;S) \end{aligned}$$

(3) If $f : X \to Y$ is a function, then

$$L\,\mathrm{gr}(f) = \bigcap_{i\in I} L_i\,\mathrm{gr}(f) \ge \bigcap_{i\in I} \mathrm{gr}(Ff) = \mathrm{gr}(Ff).$$

The other inequality is similar.

So L is a lifting, and is clearly the greatest lower bound for the L_i.

□

Proof of theorem 11. (i) We check the three conditions.

(2-cells) If $R \le R'$, then

$$\eta^* L(R) = (\eta \times \eta)^{-1}(LR) \le (\eta \times \eta)^{-1}(LR') = \eta^* L(R')$$

since for any function f, we know that f^{-1} preserves inclusions.

(lax functoriality) If $R : X \multimap Y$ and $S : Y \multimap Z$, we have

$$\begin{aligned}\eta^*L(R;S) &= \{(x,z) \mid (\eta(x),\eta(z)) \in L(R;S)\} \\ &\geq \{(x,z) \mid \eta(x,z) \in LR;LS\} \\ &\geq \{(x,z) \mid \exists y \in Y : (\eta(x),\eta(y)) \in LR, (\eta(y),\eta(z)) \in LS\} \\ &= \eta^*L(R);\eta^*L(S)\end{aligned}$$

(lifting) Let $f : X \to Y$ be a function. Naturality of η states that $\mathrm{gr}(Ff);\mathrm{gr}(\eta) = \mathrm{gr}(\eta);\mathrm{gr}(Gf)$. From this, it follows that $\mathrm{gr}(Ff) \leq \mathrm{gr}(\eta);\mathrm{gr}(Gf);\mathrm{gr}^\circ(\eta)$. Hence, we have

$$\mathrm{gr}(Ff) \leq \mathrm{gr}(\eta);\mathrm{gr}(Gf);\mathrm{gr}^\circ(\eta) \leq \mathrm{gr}(\eta);L\,\mathrm{gr}(f);\mathrm{gr}^\circ(\eta) = \eta^*L(\mathrm{gr}(f))$$

and

$$\mathrm{gr}^\circ(Ff) \leq \mathrm{gr}(\eta);\mathrm{gr}^\circ(Gf);\mathrm{gr}^\circ(\eta) \leq \mathrm{gr}(\eta);L(\mathrm{gr}^\circ(f));\mathrm{gr}^\circ(\eta) = \eta^*L(\mathrm{gr}^\circ(f))$$

(ii) For meets, we have

$$\eta^*(\bigwedge_i L_i)(R) = (\eta\times\eta)^{-1}(\bigcap_i(L_iR)) = \bigcap_i(\eta\times\eta)^{-1}(L_iR) = \left(\bigwedge_i L_i\right)(R)$$

since meets are preserved by inverse images. For $(-)^\sim$, we have

$$\begin{aligned}\eta^*(L^\sim)(R) &= (\eta\times\eta)^{-1}(L^\sim R) \\ &= (\eta\times\eta)^{-1}((L(R^\circ))^\circ) \\ &= ((\eta\times\eta)^{-1}(L(R^\circ)))^\circ \\ &= (\eta^*L(R^\circ))^\circ \\ &= (\eta^*L)^\sim(R)\end{aligned}$$

(iii) This follows directly from preservation of $(-)^\sim$: we have

$$\begin{aligned}L \text{ is symmetric} &\iff L = L^\sim \\ &\implies \eta^*L = \eta^*(L^\sim) \\ &\iff \eta^*L = (^\sim\eta^*L) \\ &\iff \eta^*L \text{ is symmetric}\end{aligned}$$

(iv) Assume η is everywhere injective, and L preserves diagonals. Then let X be arbitrary. For all $(x,y) \in FX \times FX$, we have

$$\begin{aligned}(x,y) \in \eta^*L\Delta_X &\iff (\eta(x),\eta(y)) \in L\Delta_X \\ &\implies \eta(x) = \eta(y) && \text{since } L \text{ preserves diagonals} \\ &\implies x = y && \text{since } \eta \text{ is injective}\end{aligned}$$

and hence η^*L preserves diagonals.

□

Magnitude and Topological Entropy of Digraphs

Steve Huntsman
STR
Arlington, Virginia
steve.huntsman@str.us

Magnitude and (co)weightings are quite general constructions in enriched categories, yet they have been developed almost exclusively in the context of Lawvere metric spaces. We construct a meaningful notion of magnitude for flow graphs based on the observation that topological entropy provides a suitable map into the max-plus semiring, and we outline its utility. Subsequently, we identify a separate point of contact between magnitude and topological entropy in digraphs that yields an analogue of volume entropy for geodesic flows. Finally, we sketch the utility of this construction for feature engineering in downstream applications with generic digraphs.

1 Introduction

Let $\mathbf{M} = (\mathbf{M}, \otimes, 1)$ be a monoidal category (for background, see [26, 10]) and $\mathbf{C}$ a (small) $\mathbf{M}$-category, i.e., a (small) category enriched over $\mathbf{M}$. Recall that this means that $\mathbf{C}$ is specified by a set $\mathrm{Ob}(\mathbf{C})$; hom-objects $\mathbf{C}(j,k) \in \mathbf{M}$ for all $j,k \in \mathrm{Ob}(\mathbf{C})$; identity morphisms $1 \to \mathbf{C}(j,j)$ for all $j \in \mathrm{Ob}(\mathbf{C})$; and composition morphisms $\mathbf{C}(j,k) \otimes \mathbf{C}(k,\ell) \to \mathbf{C}(j,\ell)$ for all $j,k,\ell \in \mathrm{Ob}(\mathbf{C})$; moreover, these hom-objects and morphisms are required to satisfy associativity and unitality properties [19, 10].

The theory of magnitude [24, 23] incorporates a $\mathbf{M}$-category $\mathbf{C}$ and a semiring S via a "size" map $\sigma : \mathrm{Ob}(\mathbf{M}) \to S$ that is constant on isomorphism classes and that satisfies $\sigma(1) = 1$ and $\sigma(X \otimes Y) = \sigma(X) \cdot \sigma(Y)$, where the semiring unit and multiplication are indicated on the right-hand sides. If $n := |\mathrm{Ob}(\mathbf{C})| < \infty$ then its *similarity matrix* $Z \in M(n,S)$ has entries $Z_{jk} := \sigma(\mathbf{C}(j,k))$. Introducing the (common) notation

$$(f[X])_{jk} := f(X_{jk})$$

as a shorthand where X is a matrix over the semiring S and f is a function on S, we have $Z := \sigma[\mathbf{C}]$.

A *weighting* is a column vector w satisfying $Zw = 1$, where the semiring matrix multiplication and column vector of ones are indicated. A *coweighting* is the transpose of a weighting for Z^T. If Z has a weighting and a coweighting, its *magnitude* is the sum of the components of either one of these: a line of algebra shows these sums necessarily coincide.

The notion of magnitude has been the subject of increasing attention over the past 15 years, and over the last year or so applications have begun to emerge based on boundary-detecting properties of (co)weightings in the setting of metric spaces [2, 16], which is virtually the only case that has been explored to date. [1] This setting emerges from the choice $\mathbf{M} = (([0,\infty], \geq), +, 0)$, which with only a very mild continuity assumption requires $\sigma(x) = \exp(-tx)$ for some constant t; varying this constant leads to the notion of a *magnitude function*. The corresponding enriched categories are precisely the *Lawvere metric spaces*, also known as *extended quasipseudometric spaces* since they generalize metric spaces by allowing distances that are infinite (extended), asymmetric (quasi-), or zero (pseudo-). In §2 we will

[1] The only exception of which we are aware is [6], which details a nontrivial example of magnitude for a certain **Vect**-category; see also Example 6.4.5 of [23].

J. Master & M. Lewis (Eds.): Fifth International Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 254–267, doi:10.4204/EPTCS.380.15

show that seemingly adjacent monoidal structures on $([0,\infty],\geq)$ in fact lead to the same construction, so to move away from the generalized metric space setting at all, it is necessary to move quite far indeed.

However, there are other interesting monoidal categories that yield applicable instantiations of magnitude, though §2 shows that these must necessarily give rise to something quite different from metric spaces. In §3, we introduce such a construction via a monoidal category **Flow** of *flow graphs* that informs the analysis of computer programs (and also, e.g., business processes), encompassing constructs that represent the transfer of control and data [7, 31] as in Figure 1. This category has two monoidal products that model "series" and "conditional" (versus "parallel" *per se*) execution of programs as well as the structure of an operad in **Set** [15] that dovetails with a hierarchical representation of input/output structure [17].

For each generic flow graph D, there is a **Flow**-category described in Lemma 1. The *topological entropy* of hom-objects in this category provides a suitable map σ into the max-plus semiring, and the resulting weighting (resp., coweighting) indicate sub-flow graphs of maximal entropy in the "forward direction" (resp., "reverse direction"). These constructions are attractive from the point of view of feature engineering for graph matching [9] and machine learning problems involving flow graphs.

Meanwhile, once we consider interactions between magnitude and topological entropy in the setting of digraphs, another point of contact is readily discernible, and we discuss it in §4. The magnitude function of a ball in the universal cover of a strong loopless digraph is closely related to the topological entropy of the digraph. In §5 we provide evidence of the utility for feature engineering based on this observation in problems involving generic digraphs.

2 Rigidity of similarity matrix arithmetic

Here we show that there is even less choice in how the theory of magnitude can be applied to metric spaces and their ilk than §2.3 of [24] suggests, wherein the usual addition operation on $([0,\infty],\geq)$ is chosen for the monoidal structure. This rigidity illustrates that meaningful notions of magnitude outside its usual arena are likely to involve very different monoidal structures and/or categories.

Proposition 1. *Let f be a strictly increasing bijection from $[0,\infty]$ to a subset of $[-\infty,\infty]$ containing 0. Then $x\otimes y := f^{-1}(f(x)+f(y))$ gives rise to a strict symmetric monoidal structure on $([0,\infty],\geq)$ with monoidal (additive) unit $f^{-1}(0)$.* □

A category $\mathbf{C}$ enriched over the strict symmetric monoidal category above has, for every $j,k\in \mathrm{Ob}(\mathbf{C})$, some $\eta_{jk} := \mathbf{C}(j,k)\in[0,\infty]$ such that $\eta_{jj} = f^{-1}(0)$ and $\eta_{jk}\otimes\eta_{k\ell}\geq\eta_{j\ell}$. That is, we have the triangle inequality $f(\eta_{jk})+f(\eta_{k\ell})\geq f(\eta_{j\ell})$. Let us therefore assume $f[\eta]=d$, and furthermore stipulate that we want our similarity matrix Z to take values in the semiring $\mathbb{R}$ with the usual structure, as opposed to some more exotic choice. Then we require a function $\sigma:[0,\infty]\to\mathbb{R}$ such that $\sigma(x\otimes y)=\sigma(x)\cdot\sigma(y)$ in order to define $Z:=\sigma[\eta]$. If we require continuity, then this generalized Cauchy equation has the unique family of solutions $\sigma(x)=\exp(-\tau f(x))$ for $\tau\in\mathbb{R}$. Now $Z=\sigma[\eta]=\sigma[f^{-1}[d]]=\exp[-\tau d]$, just as usual: i.e., this attempted generalization actually has no material effect.

What about a more exotic semiring structure on $\mathbb{R}$? The proposition above has a close analogue:

Proposition 2. *Let g be a strictly increasing function from $[-\infty,\infty]$ to itself, and taking on the value 0 (and also 1 for the final part of the statement). Then $x\oplus y := g^{-1}(g(x)+g(y))$ gives rise to a strict symmetric monoidal structure on $([-\infty,\infty],\geq)$ with monoidal (additive) unit $g^{-1}(0)$. Moreover, additionally taking $x\odot y := g^{-1}(g(x)\cdot g(y))$ gives a semiring with multiplicative unit $g^{-1}(1)$.* [2] □

[2] We thank S. Tringali for this observation. If $g(x):=\mathrm{sgn}(x)\cdot|x|^p$ for $p>0$, we get the semiring $([-\infty,\infty],\oplus,0,\cdot,1)$. If $g(x):=\exp(-\tau x)$ for $\tau<0$, then we get the semiring $([-\infty,\infty],\oplus,-\infty,+,0)$.

Now the equation for a weighting is $\bigoplus_k (Z_{jk} \odot w_k) = g^{-1}(1)$, which unpacks to the matrix equation $g[Z]g[w] = 1$ in ordinary arithmetic. Recalling that $Z = \sigma[\eta]$ and $f[\eta] = d$, we have $Z = \sigma[f^{-1}[d]]$. Meanwhile, we have the generalized Cauchy equation $\sigma(x \otimes y) = \sigma(x) \odot \sigma(y)$, which unpacks to

$$\sigma(f^{-1}(f(x) + f(y))) = g^{-1}(g(\sigma(x)) \cdot g(\sigma(y))). \tag{1}$$

Defining $h := g \circ \sigma \circ f^{-1}$, this becomes $h(f(x) + f(y)) = h(f(x)) \cdot h(f(y))$, i.e., h satisfies the usual Cauchy equation; assuming continuity, we have $h[d] = \exp[-\tau d]$. Since $g[Z] = h[d]$, the weighting equation is $h[d]g[w] = 1$, which apart from the transformation of w is the same as in ordinary arithmetic.

In short, it appears to be at least difficult–perhaps impossible–to get substantially different arithmetic of similarity matrices than the "default" while still working over the extended real numbers, regardless of which underlying arithmetic we use. The thin silver lining is that we can legitimately apply a very broad class of componentwise transformations to a (co)weighting and still interpret the result as a (co)weighting also, albeit with respect to a different underlying semiring structure.

Nevertheless, the notion of magnitude still affords useful application to quite different monoidal categories; in the sequel, we give an example.

3 Max-plus magnitude for flow graphs

Throughout this paper, by *digraph* we mean the usual notion in combinatorics. In particular, we do not allow multiple edges between vertices (i.e., a quiver is generally not a digraph *per se*). See footnote 3.

Consider the specific notion of *flow graph* discussed in [15], viz. a digraph D with exactly one source and exactly one target, such that there is a unique (entry) edge from the source and a unique (exit) edge to the target, and such that identifying the source of the entry edge with the target of the exit edge yields a strong digraph (i.e., a digraph in which every two vertices are connected by some path). An example is the digraph in the right panel of Figure 1.

Let **Flow** be the full subcategory of reflexive digraphs [3] whose objects are (combinatorially realized as) flow graphs. It turns out that there are both "series" and "parallel" tensor products on **Flow**, as well as the structure of an operad in **Set** which has a conceptually and algorithmically attractive instantiation. We are presently interested in the "series" tensor product, denoted $\boxtimes$. The idea of $\boxtimes$ is just to identify the exit edge of its first argument with the entry edge of its second argument (so unlike the "parallel" tensor product, this does not give rise to a symmetric monoidal structure). It turns out that this yields (the monoidal base of) an enriched category, viz. the **Flow**-category $\mathbf{SubFlow}_D$ of sub-flow graphs of a flow graph D (these correspond to subroutines in the context of program control flow).

Lemma 1. *[15] For a flow graph D, we can form a category* $\mathbf{SubFlow}_D$ *enriched over* **Flow** *as follows:*

- $\mathrm{Ob}(\mathbf{SubFlow}_D) := E(D)$ *(i.e., the objects of* $\mathbf{SubFlow}_D$ *are the edges of the digraph D);* [4]

[3] An object in the category **Dgph** of reflexive digraphs is $G = (U, \alpha, \omega)$, where U is a set and $\alpha, \omega : U \to U$ are *head* and *tail* functions that satisfy $\alpha \circ \omega = \omega$ and $\omega \circ \alpha = \alpha$. For $G' = (U', \alpha', \omega')$, a morphism $f \in \mathbf{Dgph}(G, G')$ is a function $f : U \to U'$ such that $f \circ \alpha = \alpha' \circ f$ and $f \circ \omega = \omega' \circ f$. The *vertices* of $G = (U, \alpha, \omega)$ are the (mutual) image $V \equiv V(G)$ of α and ω; the *loops* are the set $L \equiv L(G) := \{u \in U : \alpha(u) = \omega(u)\}$ (so that $V \subseteq L$), and the *edges* are the set $E \equiv E(G) := U \backslash L$. We recover the usual notion of a digraph by considering $\alpha \times \omega$ and its appropriate restrictions on U^2, L^2, and E^2: e.g., we can abusively write $E = (\alpha \times \omega)(E^2)$, where the LHS and RHS respectively refer to usual and reflexive notions of digraph edges. Thus a morphism $f : U \to U'$ restricts to $f|_V : V \to V'$, $f|_L : L \to L'$, and $f|_E : E \to U'$. Since morphisms are only partially specified by their actions on vertices, defining **Flow** as a full subcategory of **Dgph** is essentially a convention about vertex identification.

[4] Loops and reflexive self-edges are not included here, though the former may be accommodated without substantial changes.

```
1   START
2   repeat
3       repeat
4           repeat
5               if b goto 7
6               if b
7                   repeat
8                       S
9                   until b
10              endif
11          until b
12          do while b
13              do while b
14                  repeat
15                      S
16                  until b
17              enddo
18          enddo
19      until b
20  until b
21  HALT
```

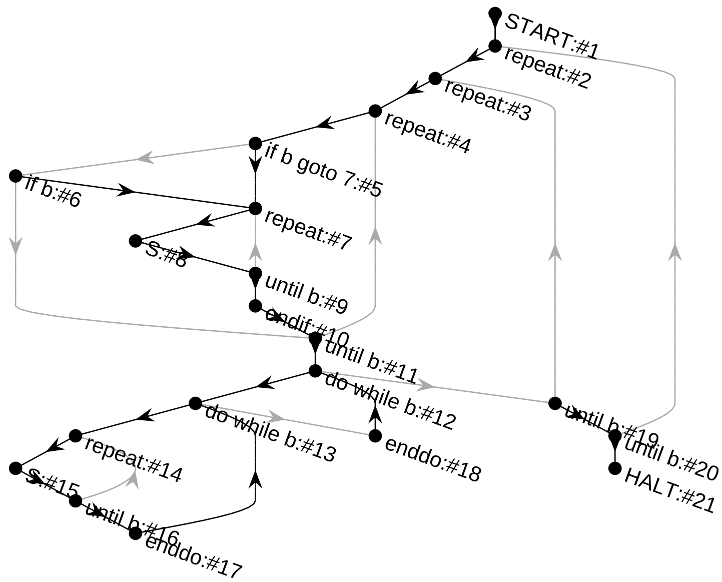


Figure 1: (L) A simple imperative program. S denotes a generic statement (or subroutine); b denotes a generic Boolean predicate. (R) The corresponding control flow graph: branches are shaded black (resp., gray) if the corresponding b evaluates to $\top$ or $\bot$.

- *for* $e_s, e_t \in \mathbf{SubFlow}_D$*, the hom object* $\mathbf{SubFlow}_D(e_s, e_t) \in \mathbf{Flow}$ *is the (possibly empty) induced sub-flow graph of* D *with entry edge* e_s *and exit edge* e_t*: we denote this by* $D\langle e_s, e_t\rangle$*;*
- *the composition morphism is induced by* $\boxtimes$*;*
- *the identity element is determined by the flow graph* e *with one edge.* □

A digraph D determines a *(sub)shift of finite type*, i.e., a dynamical system on the space of paths in D with an evolution operator that simply shifts path indices. The corresponding *topological entropy* $h(D) := \lim_{N\uparrow\infty} N^{-1} \log W(D,N)$ measures the growth of the number $W(D,N)$ of paths in D of length N [20]. A basic result in symbolic dynamics is that $h(D)$ is given by the logarithm of the spectral radius of the adjacency matrix of D. (If D is strong, the spectral radius is the Perron eigenvalue ≥ 1.)

Lemma 2. *For* $D_j \in \mathbf{Flow}$*,*

$$h(\boxtimes_j D_j) = \max_j h(D_j). \tag{2}$$

Proof. To see the $\geq$ direction, consider paths that are confined to whichever D_j has highest topological entropy. For the $\leq$ direction, note that the number of paths that are not so confined cannot grow at a faster rate. □

Remark 1. *In fact more is true: writing* $A(D)$ *for the adjacency matrix of* D*, we have via standard Perron-Frobenius theory that (as multisets)*

$$\operatorname{spec} A(\boxtimes_j D_j) = \{0\} \cup \bigcup_j \operatorname{spec} A(D_j).$$

(The zero is due to the first column/last row [using the obvious indexing] of $A(\boxtimes_j D_j)$ being identically zero.) Defining the zeta function $\zeta_D(t) := 1/\det(I - tA(D))$ [30], we furthermore have that

$$\zeta_{\boxtimes_j D_j} = \prod_j \zeta_{D_j}.$$

For examples, see Figures 2 and 3.

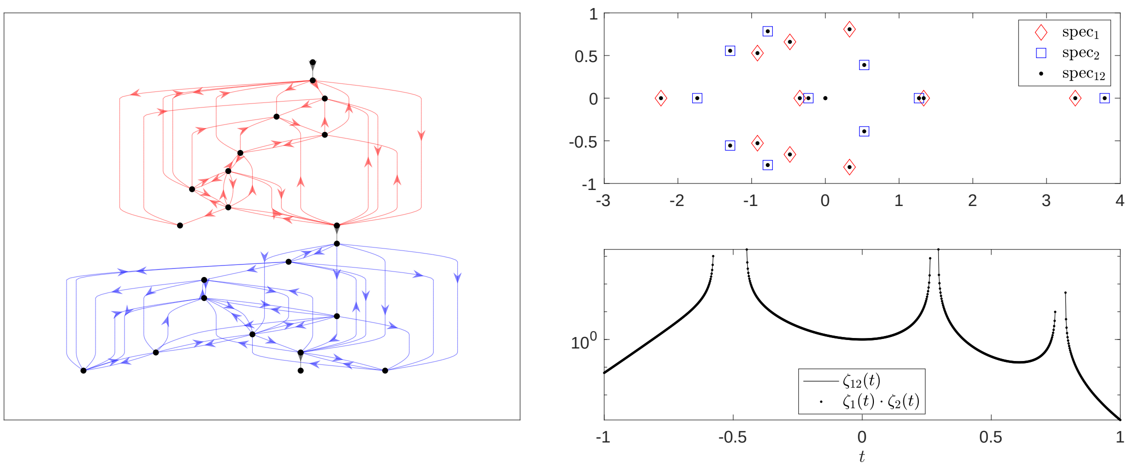


Figure 2: Left: $D_1 \boxtimes D_2$ for two flow graphs D_1 and D_2 on 10 vertices. Upper right: spectra $\text{spec}_x \subset \mathbb{C}$ of the adjacency matrices $A(D_x)$ for $x = 1$, $x = 2$, and $x = 12$ with $D_{12} := D_1 \boxtimes D_2$. Lower right: zeta functions ζ_{12} and $\zeta_1 \cdot \zeta_2$ with $\zeta_x \equiv \zeta_{D_x}$.

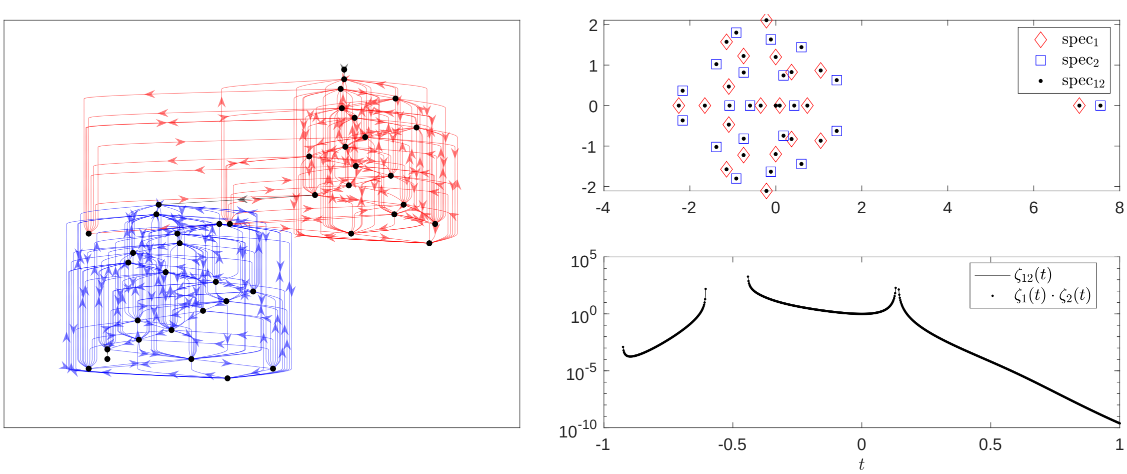


Figure 3: As in Figure 2, but for two flow graphs D_1 and D_2 on 20 vertices.

Recall that max furnishes a monoidal structure on the poset $([0,\infty], \geq)$ of extended nonnegative real numbers, and that categories enriched over this are Lawvere ultrametric spaces [32]. Similarly, $([-\infty,\infty), \leq, -\infty, \max)$ is a monoidal poset. This is sufficient data for us to define (following [24]) the

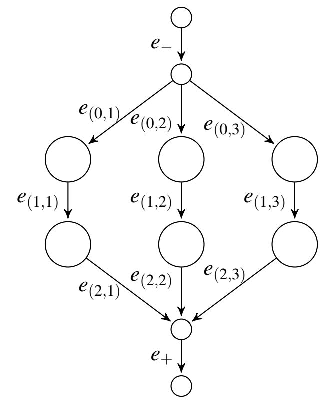


Figure 4: Flow graph of the form $D := \otimes_{k=1}^{K} \boxtimes_{j=1}^{J_k} D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ for $J_k \equiv 2$ and $K = 3$. The large nodes indicate nontrivial interiors of sub-flow graphs.

magnitude of $\mathbf{SubFlow}_D$ over the *max-plus or tropical semiring* [14, 33]. [5]

Unpacking the details, we have the similarity matrix

$$(Z_D^{\boxtimes})_{st} \equiv Z_D^{\boxtimes}(e_s, e_t) := h(D\langle e_s, e_t\rangle). \tag{3}$$

Now if there exist v, w satisfying the max-plus matrix (co)weighting equations

$$\max_s [v_s + (Z_D^{\boxtimes})_{st}] = 0 = \max_t [(Z_D^{\boxtimes})_{st} + w_t],$$

then the maxima of v and w coincide and also equal the magnitude of $Z_D^{\boxtimes}$. Such linear equations can be solved via methods described in [14], and we simply report the result here: the unique "principal solutions" (which may not be *bona fide* solutions in general) are $\hat{v}_s := -\max_t (Z_D^{\boxtimes})_{st}$; $\hat{w}_t := -\max_s (Z_D^{\boxtimes})_{st}$. We therefore obtain the following

Lemma 3. *$Z_D^{\boxtimes}$, and hence $\mathbf{SubFlow}_D$, has well-defined magnitude z over the max-plus semiring iff*

$$\max_s [-\max_t (Z_D^{\boxtimes})_{st}] = z = \max_t [-\max_s (Z_D^{\boxtimes})_{st}]. \quad \square \tag{4}$$

It is not obvious when such a z can exist. However, by Lemma 5 of [15], any nontrivial $D\langle e_s, e_t\rangle$ must be of the form $\boxtimes_j D\langle e_{j-1}, e_j\rangle$ where the $D\langle e_{j-1}, e_j\rangle$ are minimal. Appealing to Lemma 2, we therefore obtain the following

Theorem 1. *$Z_D^{\boxtimes}$, and hence $\mathbf{SubFlow}_D$, has well-defined magnitude over the max-plus semiring.* $\square$

Example 1. *Consider a flow graph of the form $D := \otimes_{k=1}^{K} \boxtimes_{j=1}^{J_k} D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$, where $\otimes$ denotes the parallel tensor/composition on* **Flow** *described in [15]. For an example, see Figure 4. For convenience, further assume that the program structure trees of $D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ are all trivial, i.e., there are no nontrivial sub-flow graphs. Then $(Z_D^{\boxtimes})_{(j_0,k),(j_1,k)} = \max_{j_0 < j \le j_1} h(D\langle e_{(j-1,k)}, e_{(j,k)}\rangle)$, $(Z_D^{\boxtimes})_{-\infty,\infty} = h(D)$, where $\mp\infty$ indicate the entry and exit edges of D, and all other entries of $Z_D^{\boxtimes}$ are trivial.*

The nontrivial weighting components are therefore

$$w_{(j,k)} = -\max_{j_0 < j} h(D\langle e_{(j_0,k)}, e_{(j,k)}\rangle) = -\max_{j_0 < j} h(D\langle e_{(j_0,k)}, e_{(j_0+1,k)}\rangle),$$

[5] It is important to distinguish between the magnitude of $\mathbf{SubFlow}_D$ as an enriched category and the magnitude of D as a digraph with the usual (asymmetric) notion of distance. Here we are concerned only with the former.

while the nontrivial coweighting components are

$$v_{(j,k)} = -\max_{j_1>j} h(D\langle e_{(j,k)}, e_{(j_1,k)}\rangle) = -\max_{j_1>j} h(D\langle e_{(j_1-1,k)}, e_{(j_1,k)}\rangle).$$

That is, the weighting and coweighting respectively encode the cumulative forward and reverse maxima of the topological entropy along the K "backbones" $\boxtimes_{j=1}^{J_k} D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ *of D. In particular,* $v_{(j_*-1,k)} = w_{(j_*,k)}$ *when* $j_* = \arg\max_j h(D\langle e_{(j-1,k)}, e_{(j,k)}\rangle)$.

Finally, it is evident that similar behavior to that detailed in Example 1 should occur when each $D\langle e_{(j-1,k)}, e_{(j,k)}\rangle$ is itself of the form $\otimes_m \boxtimes_\ell D'\langle e'_{(\ell-1,m)}, e'_{(\ell,m)}\rangle$, and so on. That is, (co)weightings reliably encode salient features for "series-parallel" flow graphs. It seems likely that the same is true for flow graphs that correspond to "structured" control flow, which can always be obtained from "unstructured" control flow [41] in the event that it makes any practical difference.

Operationally, the (co)weighting identifies regions of high topological entropy. [6] This echoes the observations of [2] that (co)weightings pick out salient features of Euclidean point clouds (e.g., "strata" of sampled psuedomanifolds). In turn, this suggests a strategy for "anchoring" graph matching methods for related flow graphs (e.g., for different versions of the same program or business process). Namely, iteratively coarsen suitably (re)structured flow graphs using the technique of [15], attempting to match regions of high topological entropy at each stage of the process. Recalling Example 1, suppose that $D' := \otimes_{k=1}^{K} \boxtimes_{j=1}^{J_k} D'\langle e'_{(j-1,k)}, e'_{(j,k)}\rangle$ is somehow related to D. We can hope to leverage the respective (co)weightings for graph matching between D and D'.

4 Magnitudes of balls in the universal cover of a digraph

For a finite strong digraph, a ball around any vertex (defined by, e.g. distance to or from that vertex) eventually saturates. It is helpful to shift perspectives to the *universal cover* [8] to avoid this saturation while using a notion of the size of these balls to characterize the digraph. [7] This perspective shift is motivated by the context of a (compact connected) Riemannian manifold, for which the *volume entropy* [27] is defined via $\lim_{r\uparrow\infty} r^{-1} \log \mathrm{vol}(B_x(r))$, where $B_x(r)$ is the ball of radius r around a point x in the universal cover of the manifold. It turns out that the volume entropy is independent of the point x. Also, the volume entropy is bounded above by the topological entropy of the geodesic flow, with equality in the case of nonpositive sectional curvature. Proposition 5 is a very close analogue of this result. [8]

Returning to the context of digraphs, the universal cover of a digraph is a *polytree*, (i.e., an acyclic digraph whose corresponding undirected graph is a tree) that "locally looks like the digraph everywhere." A telling advantage of this construction is that (at the cost of implicitly encoding structure) it renders the calculation of magnitude functions trivial:

Lemma 4. *Let F be a* polyforest*, i.e., an acyclic digraph whose corresponding undirected graph is a forest. Then the magnitude function of F (i.e., the magnitude of* $\exp[-td]$ *where d is the usual Lawvere metric on F) is* $|V(F)| - |E(F)|e^{-t}$. [9]

Proof (sketch). The proof can be adapted almost wholesale from an analogous result for undirected trees (or for that matter, forests) in §4 of [22]: apart from checking and slightly adjusting definitions, the

---

[6] NB. Both $Z_D^{\boxtimes}$ and its (co)weighting are efficiently computable, as is any necessary preprocessing/restructuring of D.

[7] For the conventional notion of a universal cover in topology, see [13, 11].

[8] There is a kind of volume entropy for metric graphs [25, 21] (see also [18]), but we are unaware of a digraph analogue.

[9] Note that if F is a polytree, then $|V(F)| = |E(F)| + 1$.

key observation is that the magnitude function of a digraph with a single (directed) edge is $2-e^{-t}$ (by comparison, the magnitude function of a graph with a single edge is $2(1+e^{-t})^{-1}$). $\square$

The universal cover $U_D := (V_U, E_U)$ of a weak digraph $D = (V,E)$ is a polytree defined as follows [8]: pick $v_0 \in V$ and set

$$V_U := \{(v_0, v_1, \dots, v_L) : (v_{j-1}, v_j) \in E; v_{j-1} \neq v_j\} \cup \{(v_L, v_{L-1}, \dots, v_0) : (v_j, v_{j-1}) \in E; v_j \neq v_{j-1}\}$$

where $v_j \in V$ and $e_j \in E$ identically; and set

$$\begin{aligned} E_U := &\{((v_0, v_1, \dots, v_{L-1}), (v_0, v_1, \dots, v_L)) : (v_{L-1}, v_L) \in E\} \\ &\cup \{((v_0, v_1, \dots, v_L), (v_0, v_1, \dots, v_{L-1})) : (v_L, v_{L-1}) \in E\}. \end{aligned}$$

Example 2. *Consider the digraph D in the left panel of Figure 5. Its universal cover has local structure shown in the right panel of Figure 5, and the covering map is depicted in Figure 6 (which also shows a larger local region of the universal cover).*

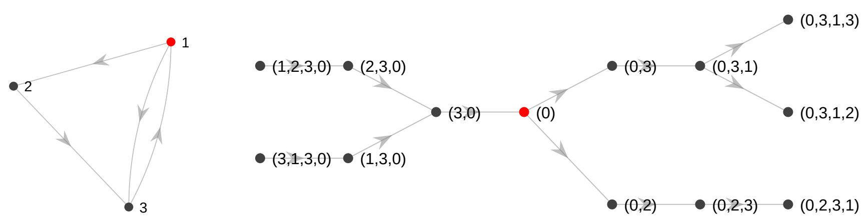


Figure 5: (L) A strong loopless digraph D with basepoint $v_0 = 1$ highlighted in red. (R) The portion of U_D with vertices at distance ≤ 3 to or from v_0. Vertices of U_D are labeled by the corresponding sequence of D-vertices, with 0 explicitly indicating the basepoint. The ball $B_0(3)$ is formed by taking the arborescence of depth 3 rooted at 0, i.e., the right-hand part.

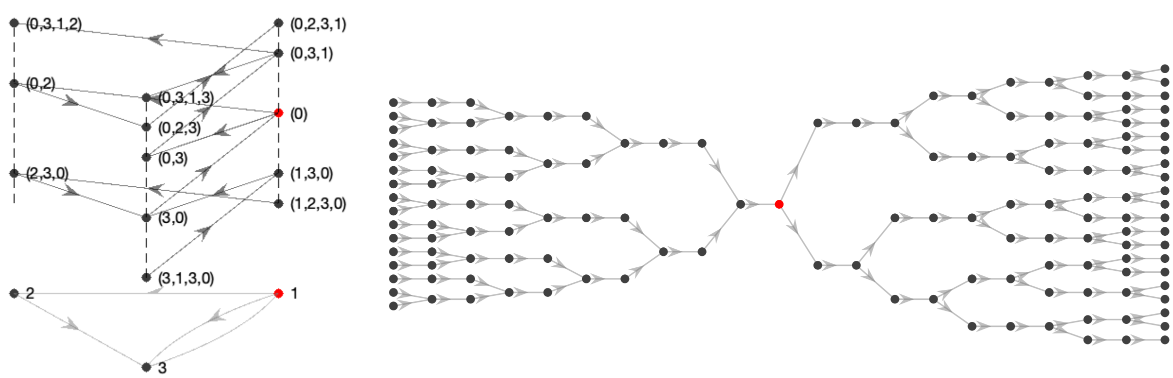


Figure 6: (Cf. Figure 5.) (L) The portion of U_D with vertices at distance ≤ 3 to or from v_0 with covering of D (at bottom) indicated. (R) The portion of U_D with vertices at distance ≤ 10 to or from v_0.

Proposition 3. *Let $\gamma \in V_U$. Then there is either a unique path in U_D from v_0 to γ or* vice versa*.* $\square$

The number of paths from v_0 of length L in U_D equals the number of loopless paths from v_0 of length L in D. Define $B_{v_0}(L)$ to be the sub-polytree of U_D (defined with basepoint v_0) induced by its vertices at (the usual notion of digraph) distance $\leq L$ from (versus to) v_0. We can compute the magnitude function of $B_{v_0}(L)$ very easily using the following proposition.

Proposition 4. *If D is loopless, then $B_{v_0}(L)$ is an arboresence with $|V(B_{v_0}(L))| = \sum_{\ell=0}^{L} \sum_k (A^\ell)_{jk}$, where A is the adjacency matrix of D and j is the matrix index corresponding to v_0.* □

Remark 2. *By comparison, the* Katz centrality *is $\sum_{\ell=1}^{\infty} \alpha^\ell \sum_i (A^\ell)_{ij}$, where α is restricted to ensure convergence [12]. The Katz centrality of the graph with all edges reversed is therefore $\sum_{\ell=1}^{\infty} \alpha^\ell \sum_k (A^\ell)_{jk}$.*

Since an arborescence (or more generally a polytree) has one more vertex than it has edges, Lemma 4 yields that for D loopless, the magnitude function of $B_{v_0}(L)$ is

$$\mathrm{Mag}(B_{v_0}(L),t) = |V(B_{v_0}(L))| - (|V(B_{v_0}(L))| - 1)e^{-t}, \tag{5}$$

and the most recent proposition gives an elementary algorithm for computing $|V(B_{v_0}(L))|$. If D is loopless and strong, we have $h(D) = h(U_D) =: \lim_{L\uparrow\infty} L^{-1} \log |V(B_{v_0}(L))|$ independent of the basepoint v_0.

Proposition 5. *Let D be a strong loopless digraph and $v_0 \in V(D)$. Then*

$$\lim_{L\uparrow\infty} L^{-1} \log \mathrm{Mag}(B_{v_0}(L),t) \leq h(D) \tag{6}$$

with equality at $t = \infty$, and the left hand side is independent of v_0 for any t. Here $\mathrm{Mag}(\cdot,t)$ *denotes the magnitude function of the first argument.* □

Example 3. *Continuing Example 2, $h(D) \approx 0.2812$ is the logarithm of the so-called* plastic number*, i.e., the unique real solution of $x^3 - x - 1 = 0$. Numerics suggest that $|V(B_1(L))|$ is given by [1]. Assuming this to obtain values for large L, we show the convergence of $L^{-1} \log \mathrm{Mag}(B_{v_0}(L),t)$ in Figure 7.*

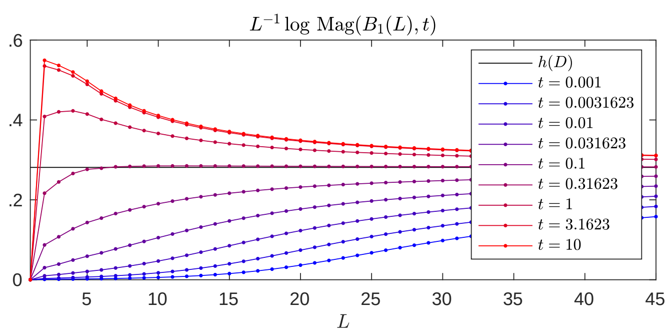


Figure 7: $L^{-1} \log \mathrm{Mag}(B_{v_0}(L),t) \to h(D)$ for $t > 0$, but depends strongly on t even for fairly large L.

5 Example: correlated features for digraph matching

In this section we detail how log-magnitudes of small balls associated to the Lawvere metric structure on a digraph are both interesting and useful from the perspective of feature engineering; for completeness and comparison, we start by considering the ambient (co)weighting. In keeping with the general theme of providing tools for graph matching, we focus on the import graph of the `Flare` software hierarchy, accessed from `https://observablehq.com/@d3/hierarchical-edge-bundling/2` in November 2020 and depicted in Figure 8.

As an experiment, we considered $N = 100$ realizations of a pair of random subgraphs of the "ambient" digraph of Figure 8 obtained by removing edges with probability $3/4$ and then retaining the largest weak component. We then computed the (co)weightings at scale 0, the log-magnitudes of balls of radius ≤ 3 at scale $t = 100$ (which is virtually equivalent to $t = \infty$), and various common vertex centrality measures. For each of these quantities and N realizations, we then computed the correlation coefficients on vertices shared by the pair of subgraphs. The results are shown in Figure 9, which shows that the coweighting and log-magnitudes of balls in the universal cover of the digraph with edges reversed are very strongly correlated. This suggests the utility of such features for graph comparison [38] and matching [9]. [10]

The strong correlations of log-magnitudes of balls are more robust than those of (co)weightings, as an experiment along the same lines as above but using different realizations of an Erdős-Renyí digraph ($n = 100$ vertices; edge probability $q = 4/n$) as the ambient digraph for each of $N = 100$ trials shows. We formed two subgraphs by removing edges with probability $1/2$, then retaining the largest weak component. Figure 10 shows the results, which are qualitatively echoed for different parameters.

One theoretical advantage of using log-magnitudes of balls is that unlike (co)weightings, these are nonnegative by construction. [11] [12] This may be advantageous in the context of graph matching via optimal transport techniques that require a sensible distribution on vertices. In particular, the recently developed *Gromov-Wasserstein distance* [28, 29] is useful for analyzing weighted digraphs endowed with measures [3] and has been applied to (mostly but not exclusively undirected) graph matching with state of the art performance [36, 40, 39, 5, 4, 37]. For instance, although [39] did not consider digraphs, it used a distribution proportional to $(\deg + a)^b$, where a and b are hyperparameters, and remarked that "the node distributions have a huge influence on the stability and the performance of our learning algorithms." Meanwhile, this particular sort of distribution is rather similar to the log-magnitude of a unit ball for $a = 1$ and $b = 0$. In short, we can plausibly expect to improve upon the approach of [39] in the context of digraphs by using weightings rather than a more *ad hoc* distribution.

Acknowledgement

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)

References

[1] Roger L. Bagula (2009): *Online Encyclopedia of Integer Sequences: sequence A167385*. Available at `http://oeis.org/A167385`.

[10] For a naive approach of matching nodes based on rankings derived from centralities, see Figure 7 of [34].

[11] NB. It is possible using elementary bounds to efficiently determine the minimal t such that a [co]weighting of $Z = \exp[-td]$ is nonnegative [16].

[12] Note that from the point of view of correlation analyses, log-magnitudes are more interesting than the magnitudes themselves: because the correlation coefficient is invariant under affine transformations of either argument, we have that $\rho(\mathrm{Mag}(B_{v_0}(L),t), \mathrm{Mag}(B'_{v_0}(L),t')) = \rho(|V(B_{v_0}(L))|, |V(B'_{v_0}(L))|)$. In any event, $\lim_{t\uparrow\infty} \mathrm{Mag}(B_{v_0}(L),t) = |V(B_{v_0}(L))|$.

[2] Eric Bunch et al. (2020): *Practical applications of metric space magnitude and weighting vectors*, doi:10.48550/arXiv.2006.14063.

[3] Samir Chowdhury & Facundo Mémoli (2019): *The Gromov–Wasserstein distance between networks and stable network invariants*. Information and Inference 8(4), pp. 757–787, doi:10.1093/imaiai/iaz026.

[4] Samir Chowdhury & Tom Needham (2020): *Gromov-Wasserstein averaging in a Riemannian framework*. In: *CVPR Workshops*, doi:10.1109/CVPRW50498.2020.00429.

[5] Samir Chowdhury & Tom Needham (2021): *Generalized spectral clustering via Gromov-Wasserstein learning*. In: *AISTATS*. Available at `http://proceedings.mlr.press/v130/chowdhury21a.html`.

[6] Joseph Chuang, Alastair King & Tom Leinster (2016): *On the magnitude of a finite dimensional algebra*. Theory and Applications of Categories 31(3), pp. 63–72. Available at `http://www.tac.mta.ca/tac/volumes/31/3/31-03abs.html`.

[7] Keith Cooper & Linda Torczon (2011): *Engineering a Compiler*. Elsevier, doi:10.1016/C2009-0-27982-7.

[8] Willibald Dörfler, Frank Harary & Günther Malle (1980): *Covers of digraphs*. Mathematica Slovaca 30(3), pp. 269–280. Available at `https://eudml.org/doc/34089`.

[9] Frank Emmert-Streib, Matthias Dehmer & Yongtang Shi (2016): *Fifty years of graph matching, network alignment and network comparison*. Information Sciences 346, pp. 180–197, doi:10.1016/j.ins.2016.01.074.

[10] Brendan Fong & David I Spivak (2019): *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge, doi:10.1017/9781108668804

[11] Robert W Ghrist (2014): *Elementary Applied Topology*. Createspace. Available at `https://www2.math.upenn.edu/~ghrist/notes.html`.

[12] Peter Grindrod & Desmond J Higham (2013): *A matrix iteration for dynamic network summaries*. SIAM Review 55(1), pp. 118–128, doi:10.1137/110855715.

[13] Allen Hatcher (2002): *Algebraic Topology*. Cambridge. Available at `https://pi.math.cornell.edu/~hatcher/AT/ATpage.html`.

[14] Bernd Heidergott, Geert Jan Olsder & Jacob Van Der Woude (2014): *Max Plus at Work*. Princeton, doi:10.1515/9781400865239.

[15] Steve Huntsman (2019): *The multiresolution analysis of flow graphs*. In: *WoLLIC*, doi:10.1007/978-3-662-59533-6˙20.

[16] Steve Huntsman (2022): *Diversity enhancement via magnitude*, doi:10.48550/arXiv.2201.10037.

[17] Richard Johnson, David Pearson & Keshav Pingali (1994): *The program structure tree: computing control regions in linear time*. In: *PLDI*, doi:10.1145/178243.178258.

[18] Steve Karam (2015): *Growth of balls in the universal cover of surfaces and graphs*. Transactions of the American Mathematical Society 367(8), pp. 5355–5373, doi:10.1090/S0002-9947-2015-06189-3.

[19] Gregory Maxwell Kelly (1982): *Basic Concepts of Enriched Category Theory*. Cambridge.

[20] Bruce P Kitchens (2012): *Symbolic Dynamics*. Springer, doi:10.1007/978-3-642-58822-8.

[21] Hyekyoung Lee et al. (2019): *Volume entropy for modeling information flow in a brain graph*. Scientific Reports 9(1), pp. 1–13, doi:10.1038/s41598-018-36339-7.

[22] Tom Leinster (2019): *The magnitude of a graph*. Mathematical Proceedings of the Cambridge Philosophical Society 166(2), pp. 247–264, doi:10.1017/S0305004117000810.

[23] Tom Leinster (2021): *Entropy and Diversity: the Axiomatic Approach*. Cambridge, doi:10.1017/9781108963558.

[24] Tom Leinster & Mark W Meckes (2017): *The magnitude of a metric space: from category theory to geometric measure theory*. In Nicola Gigli, editor: Measure Theory in Non-Smooth Spaces, De Gruyter, doi:10.1515/9783110550832-005.

[25] Seonhee Lim (2008): *Minimal volume entropy for graphs*. Transactions of the American Mathematical Society 360(10), pp. 5089–5100, doi:10.1090/S0002-9947-08-04227-X.

[26] Saunders Mac Lane (1978): *Categories for the working mathematician*. Springer, doi:10.1007/978-1-4757-4721-8.

[27] Anthony Manning (1979): *Topological entropy for geodesic flows*. Annals of Mathematics 110(3), pp. 567–573, doi:10.2307/1971239.

[28] Facundo Mémoli (2011): *Gromov–Wasserstein distances and the metric approach to object matching*. Foundations of Computational Mathematics 11(4), pp. 417–487, doi:10.1007/s10208-011-9093-5.

[29] Facundo Mémoli (2014): *The Gromov–Wasserstein distance: a brief overview*. Axioms 3(3), pp. 335–341, doi:10.3390/axioms3030335.

[30] Hirobumi Mizuno & Iwao Sato (2001): *Zeta functions of digraphs*. Linear Algebra and its Applications 336(1-3), pp. 181–190, doi:10.1016/S0024-3795(01)00318-4.

[31] Flemming Nielson, Hanne R Nielson & Chris Hankin (2004): *Principles of Program Analysis*. Springer, doi:10.1007/978-3-662-03811-6.

[32] nLab authors (2020): `http://ncatlab.org/nlab/show/enriched%20category`. Revision 94.

[33] nLab authors (2020): `http://ncatlab.org/nlab/show/max-plus%20algebra`. Revision 2.

[34] Ryoma Sato et al. (2020): *Fast and robust comparison of probability measures in heterogeneous spaces*, doi:10.48550/arXiv.2002.01615.

[35] David Selassie, Brandon Heller & Jeffrey Heer (2011): *Divided edge bundling for directional network data*. IEEE Transactions on Visualization and Computer Graphics 17(12), pp. 2354–2363, doi:10.1109/TVCG.2011.190.

[36] Titouan Vayer et al. (2019): *Optimal transport for structured data with application on graphs*. In: ICML. Available at `http://proceedings.mlr.press/v97/titouan19a.html`.

[37] Titouan Vayer et al. (2020): *Fused Gromov-Wasserstein distance for structured objects*. Algorithms 13(9), p. 212, doi:10.3390/a13090212.

[38] Peter Wills & François G Meyer (2020): *Metrics for graph comparison: a practitioner's guide*. PLoS ONE 15(2), p. e0228728, doi:10.1371/journal.pone.0228728.

[39] Hongteng Xu, Dixin Luo & Lawrence Carin (2019): *Scalable Gromov-Wasserstein learning for graph partitioning and matching*. In: NeurIPS. Available at `https://papers.nips.cc/paper/2019/hash/6e62a992c676f611616097dbea8ea030-Abstract.html`.

[40] Hongteng Xu et al. (2019): *Gromov-Wasserstein learning for graph matching and node embedding*. In: ICML. Available at `http://proceedings.mlr.press/v97/xu19b.html`.

[41] Fubo Zhang & Erik H D'Hollander (2004): *Using hammock graphs to structure programs*. IEEE Transactions on Software Engineering 30(4), pp. 231–245, doi:10.1109/TSE.2004.1274043.

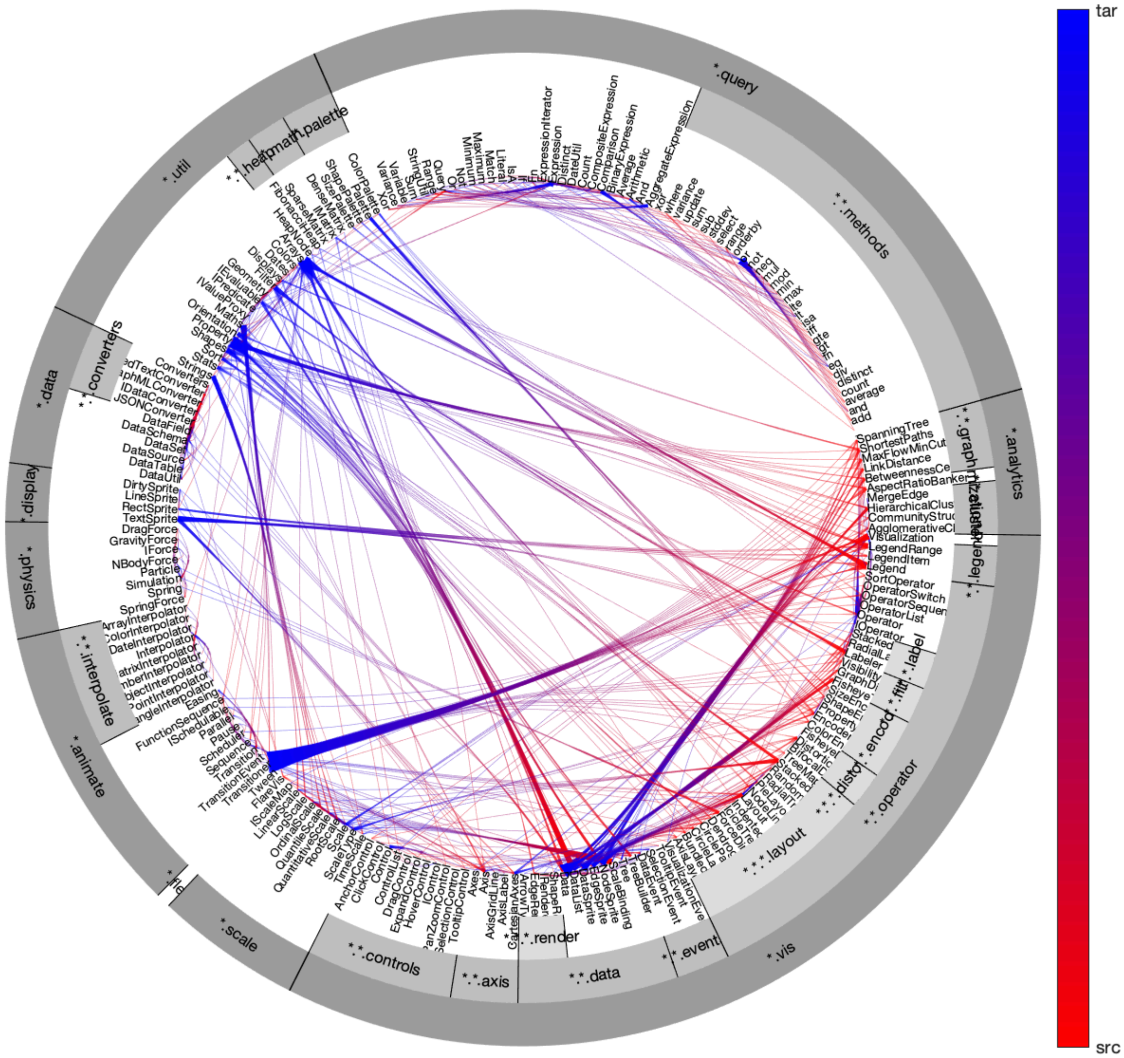


Figure 8: Import graph of the `Flare` software hierarchy, displayed using the divided edge bundling approach of [35]. Edge sources and targets are respectively colored red and blue; the hierarchy is depicted along the figure periphery (and without any material loss of information from occlusions).

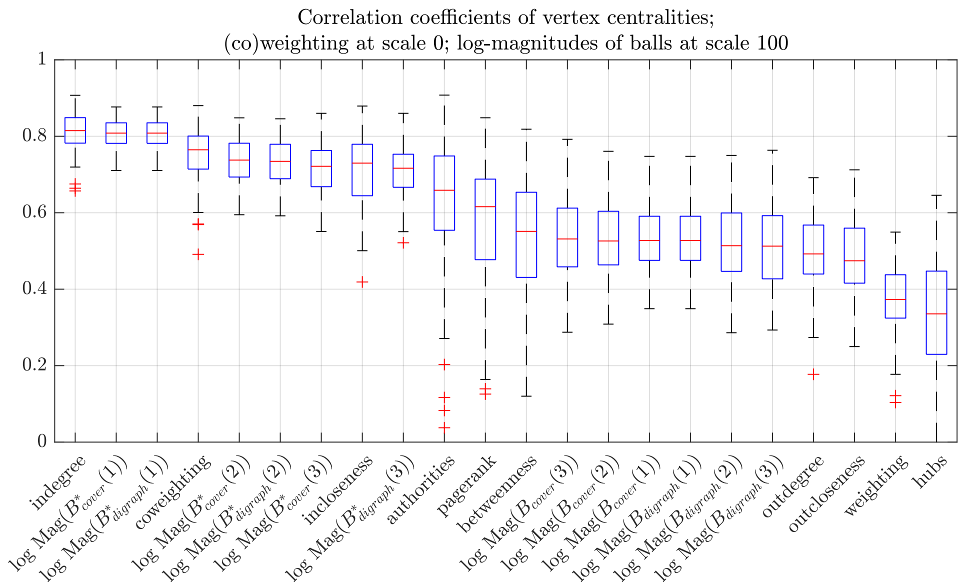


Figure 9: Distribution of correlations for various centralities between two random subgraphs of the digraph in Figure 8. * indicates a ball in the digraph with all edges reversed. As L increases, boundary effects cause the log-magnitudes of balls in the universal cover to become (slightly) more correlated to each other than the log-magnitudes of balls in the digraph itself. Note that the three best-performing centralities are computing almost exactly the same thing.

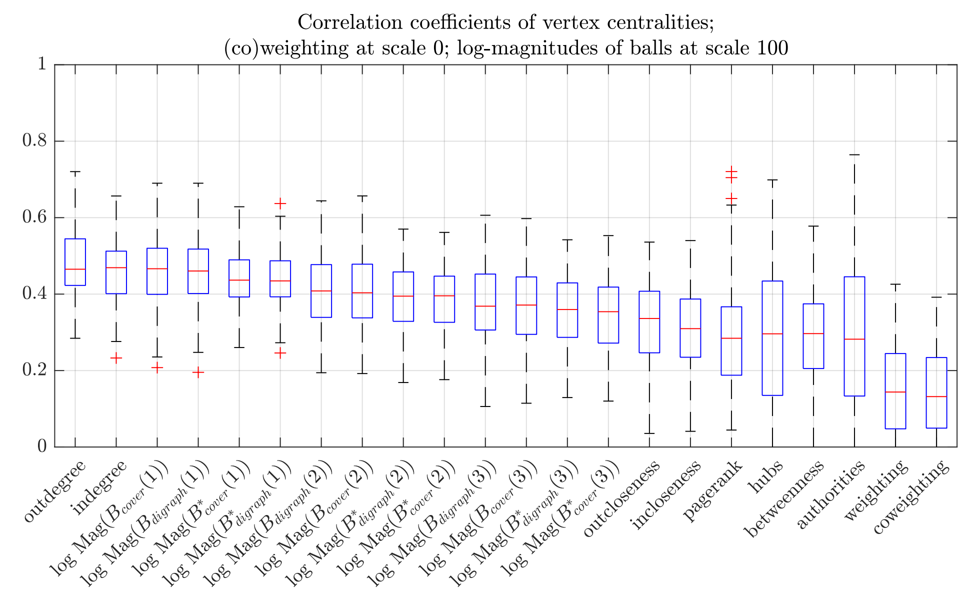


Figure 10: Distribution of correlations for various centralities between two random subgraphs of distinct Erdős-Renyí digraphs with $n = 100$ vertices and edge probability $q = 4/n$; cf. Figure 9.

Monoidal Width: Capturing Rank Width

Elena Di Lavore
Tallinn University of Technology

Paweł Sobociński
Tallinn University of Technology

Monoidal width was recently introduced by the authors as a measure of the complexity of decomposing morphisms in monoidal categories. We have shown that in a monoidal category of cospans of graphs, monoidal width and its variants can be used to capture tree width, path width and branch width. In this paper we study monoidal width in a category of matrices, and in an extension to a *different* monoidal category of open graphs, where the connectivity information is handled with matrix algebra and graphs are composed along edges instead of vertices. We show that here monoidal width captures rank width: a measure of graph complexity that has received much attention in recent years.

1 Introduction

Many applications of category theory rely on monoidal categories as algebras of processes [26, 15, 28, 18, 10, 25, 11, 17, 23, 27]. Morphisms are compound processes, defined as parallel and sequential compositions of simpler process components. The compositional nature of this modelling allows a compositional computation of the underlying semantics. But how efficient is this computation? Given two processes f and g, we can compute their semantics separately. However, computing the semantics of their sequential composition $f\,;g$ often requires an additional cost [36]. Indeed, the semantics of sequential composition often means resource sharing or synchronisation along the common boundary. This in turn carries a computational burden, dependent on the size of the boundary. On the other hand, computing the semantics of a parallel composition $f \otimes f'$ typically does not involve any resource sharing, as indicated by the wiring of the string diagrams, and thus typically does not require significant additional computational resources. Taking this into account, the choice of the *recipe* for a morphism in terms of parallel and sequential compositions influences the cost of computing its semantics. As shown in Figure 1, where vertical cuts represent sequential compositions and horizontal cuts represent parallel compositions, the same morphism can be defined in different ways with possibly different computational costs. Given a morphism, it is thefore desirable to find the least costly recipe of *decomposing* it in terms

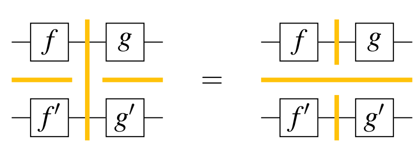


Figure 1: Two monoidal decompositions of the same morphism, the right one being the cheapest.

of more primitive components. We can rephrase the original question: what is the most efficient way to decompose a morphism in a monoidal category?

The authors recently proposed *monoidal width* [22] as a way of assigning a natural number to a morphism of a monoidal category, representing – roughly speaking – the cost of its most efficient decomposition. In turn, this is related to the cost of computing the semantics of this morphism.

Computing efficient decompositions is not a new problem. The graph theory literature abounds [6, 29, 38, 37, 39, 33, 20, 2, 3, 16] with notions of complexity of graphs that ultimately measure the difficulty

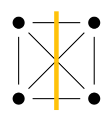
J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 268–283, doi:10.4204/EPTCS.380.16

of decomposing a graph into smaller components by cutting along the vertices or the edges of the graph. Measures such as tree width [6, 29, 38], path width [37], branch width [39], clique width [20] and rank width [33] are motivated by algorithmic considerations. Probably the best known among several results that etablish links with algorithms [8, 9, 19], the following shows the importance of tree width.

Theorem (Courcelle [19])**.** *Every property expressible in the monadic second order logic of graphs can be tested in linear time on graphs with bounded tree width.*

The different notions of complexity for graphs vastly differ in low-level "implementation details" but they all share a similar underlying idea: that of defining decompositions and suitably measuring their efficiency. One of our contributions is to exhibit monoidal width as a unifying framework for graph measures based on a notion of decomposition. In fact, by choosing a suitable algebra of composition for graphs — i.e. choosing the right monoidal cateory — we recover some of these known measures as particular instances of monoidal width. We have previously captured [22] tree width, path width and branch width by instantiating monoidal width and two variants in a category of cospans of graphs.

In this paper we focus on rank width [33] – a relatively recent development that has attracted significant attention in the graph theory community. A rank decomposition is a recipe for decomposing a graph into its single-vertex subgraphs by cutting along its edges. The cost of a cut is the rank of the adjacency matrix that represents it, as shown in Figure 2. A useful intuition for rank width is that it is a kind of "Kolmogorov complexity" for graphs. For example, although the family of cliques has unbounded tree width, the connectivity of cliques is quite simple to describe: and, in fact, all cliques have rank width 1.



$$\mathsf{rank}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix} = 1$$

Figure 2: A cut and its matrix in a rank decomposition.

To capture rank width as an instance of monoidal width, rather than taking cospans, we work in a different monoidal category of graphs. First introduced in [14], it was recently used [21] as a syntax for network games. This approach to computing with "open graphs" is more linear algebraic, building modularly on the theory of bialgebra, well known to be closely related to matrix algebra [41]. Indeed, the connectivity of graphs is handled with adjacency matrices, and boundary connections are matrices.

Related work. This manuscript, although self-contained, complements our previous work [22], where we considered tree width, path width and branch width as instances of monoidal width.

Previous syntactical approaches to graph widths are the work of Pudlák, Rödl and Savickỳ [35] and the work of Bauderon and Courcelle [5]. Their works consider different notions of graph decompositions, which lead to different notions of graph complexity. In particular, in [5], the cost of a decomposition is measured by counting *shared names*, which is clearly closely related to penalising sequential composition as in monoidal width. Nevertheless, these approaches are specific to particular, concrete notions of graphs, whereas our work concerns the more general algebraic framework of monoidal categories.

Recent abstract approaches focus on other graph widths. The work of Blume et. al. [7], characterises tree and path decompositions in terms of colimits. Abramsky et. al. [24] give a coalgebraic characterization of tree width of relational structures (and graphs in particular). Bumpus and Kocsis [13] also generalise tree width to the categorical setting, although their approach is far removed from ours.

Synopsis. Monoidal width is recalled in Section 2. In Section 3, we study the monoidal width of matrices. Section 4 recalls rank width and gives an equivalent recursive definition of it that will be useful as an intermediate step towards our main result, which is presented in Section 5.

Preliminaries. We use string diagrams [30, 40]: sequential and parallel compositions of f and g are drawn as in Figure 3, left and middle, respectively. Much of the bureaucracy, e.g. the interchange law $(f\,;g)\otimes(f'\,;g') = (f\otimes f')\,;(g\otimes g')$, disappears (Figure 3, right). *Props* [32, 31] are important examples of

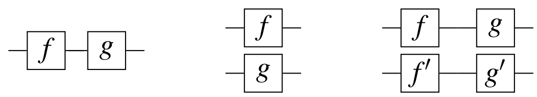


Figure 3: String diagrammatic notation.

monoidal categories. They are symmetric strict monoidal, with natural numbers as objects, and addition as monoidal product on objects. Roughly speaking, morphisms can be thought of as processes, and the objects (natural numbers) keep track of the number of inputs or outputs of a process.

2 Monoidal width

This section recalls the concept of monoidal width from [22]. Monoidal width records the cost of the most efficient way one can decompose a morphism into its atomic components, thus capturing—roughly speaking—its intrinsic structural complexity. A decomposition is a binary tree whose internal nodes are labelled with compositions or monoidal products, and whose leaves are labelled with atomic morphisms.

Definition 2.1 (Monoidal decomposition [22])**.** Let C be a monoidal category and $\mathscr{A}$ be a subset of its morphisms referred to as *atomic*. The set D_f of *monoidal decompositions* of $f\colon A\to B$ in C is defined:

$$
\begin{array}{rcll}
D_f & ::= & (f) & \text{if } f\in\mathscr{A}\\
 & \mid & (d_1,\otimes,d_2) & \text{if } d_1\in D_{f_1},\, d_2\in D_{f_2} \text{ and } f = f_1\otimes f_2\\
 & \mid & (d_1,;_X,d_2) & \text{if } d_1\in D_{f_1\colon A\to X},\, d_2\in D_{f_2\colon X\to B} \text{ and } f = f_1\,;f_2
\end{array}
$$

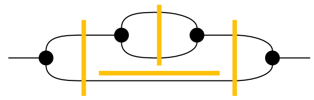
The cost of a decomposition depends on the operations and atoms present: each operation and each atomic morphism is associated with a cost, which we call weight. Roughly speaking, sequential composition is priced according to the size of the object the composition occurs over, while monoidal products are free. Finally, the weight of an atom is the application-specific cost of computing its semantics.

Definition 2.2 (Weight function [22])**.** Let C be a monoidal category and let $\mathscr{A}$ be a set of atoms for C. A weight function for $(\mathsf{C},\mathscr{A})$ is a function $\mathsf{w}\colon \mathscr{A}\cup\{\otimes\}\cup\mathsf{Obj}(\mathsf{C})\to\mathbb{N}$ such that

- $\mathsf{w}(X\otimes Y) = \mathsf{w}(X)+\mathsf{w}(Y)$,
- $\mathsf{w}(\otimes) = 0$.

Example 2.3. *Let* -●<$_1$: $1\to 2$ *and* >●-$_1$: $2\to 1$ *be the diagonal and codiagonal morphisms in a cartesian and cocartesian prop*[1] *s.t.* $\mathsf{w}($-●<$_1) = \mathsf{w}($>●-$_1) = 2$*. The following figure represents the monoidal decomposition of* -●< $;($-●< $\otimes$ —$)\,;($>●- $\otimes$ —$)\,;$ >●- *given by* (-●<, $;_2$, (((-●<, $;_2$, >●-), $\otimes$, —), $;_2$, >●-)).

[1] In a *cartesian prop* the $\otimes$ satisfies the universal property of products. Dually, in a *cocartesian prop*, the $\otimes$ satisfies the universal property of the coproduct.



The width of a decomposition is the cost of the most expensive node in the decomposition tree.

Definition 2.4 (Width of a monoidal decomposition [22])**.** Let w be a weight function for $(\mathsf{C},\mathscr{A})$. Let f be in C and $d \in D_f$. The width of d is defined recursively as follows:

$$\begin{aligned} \mathsf{wd}(d) := \mathsf{w}(f) & \qquad \text{if } d = (f) \\ \max\{\mathsf{wd}(d_1), \mathsf{wd}(d_2)\} & \qquad \text{if } d = (d_1, \otimes, d_2) \\ \max\{\mathsf{wd}(d_1), \mathsf{w}(X), \mathsf{wd}(d_2)\} & \qquad \text{if } d = (d_1, ;_X, d_2) \end{aligned}$$

As sketched in Example 2.3, decompositions can be seen as labelled trees (S,μ) where S is a tree and $\mu : \mathsf{vertices}(S) \to \mathscr{A} \cup \{\otimes\} \cup \mathsf{Obj}(\mathsf{C})$ is a labelling function. With this we can restate the width as:

$$\mathsf{wd}(d) = \mathsf{wd}(S,\mu) := \max_{v \in \mathsf{vertices}(S)} \mathsf{w}(\mu(v))$$

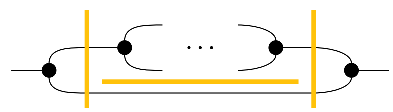
which may be familiar to those aquainted with graph widths.

Monoidal width is simply the width of the cheapest decomposition.

Definition 2.5 (Monoidal width [22])**.** Let w be a weight function for $(\mathsf{C},\mathscr{A})$ and f be in C. Then the *monoidal width* of f is $\mathsf{mwd}(f) := \min_{d \in D_f} \mathsf{wd}(d)$.

Example 2.6. *With the data of Example 2.3, define a family of morphisms $n\colon 1 \to 1$ inductively:*

- $1 := -_1$;
- $2 := -\!\!\bullet\!\!\!< \,;_2 >\!\!\!\bullet\!\!-$;
- $n+1 := -\!\!\bullet\!\!\!< \,;_2 (n \otimes 1) \,;_2 >\!\!\!\bullet\!\!-$ *for* $n \geq 2$.



Each n has a monoidal decomposition of width n: the root node is the composition along the n wires in the middle. However, $\mathsf{mwd}(n) = 2$ *for any n, with an optimal decomposition shown above.*

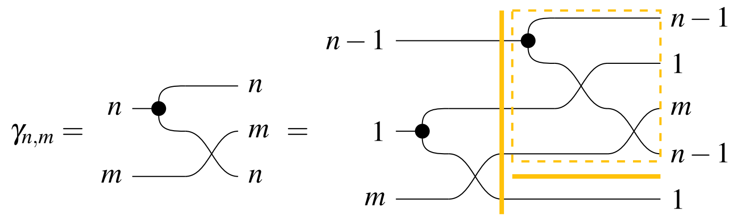
2.1 The width of copying

Before we begin with the original technical contributions of this paper in Section 3, we need to recall a technical result from [22] about decomposing copy morphisms. We consider symmetric monoidal categories equipped with such morphisms and show that copying n wires costs at most $n+1$.

Definition 2.7 (Copying)**.** Let X be a symmetric monoidal category with symmetries given by $\times_{X,Y}$. We say that X has *coherent copying* if there is a class of objects $\mathscr{C}_\mathsf{X} \subseteq \mathsf{Obj}(\mathsf{X})$, satisfying $X,Y \in \mathscr{C}_\mathsf{X}$ iff $X \otimes Y \in \mathscr{C}_\mathsf{X}$, such that every X in $\mathscr{C}_\mathsf{X}$ is endowed with a morphism $-\!\!\bullet\!\!\!<_X \colon X \to X \otimes X$. Moreover, $-\!\!\bullet\!\!\!<_{X\otimes Y} = (-\!\!\bullet\!\!\!<_X \otimes -\!\!\bullet\!\!\!<_Y) \,;(-_X \otimes \times_{X,Y} \otimes -_Y)$ for every $X,Y \in \mathscr{C}_\mathsf{X}$.

An example is any cartesian prop with $-\!\!\bullet\!\!\!<_n \colon n \to n+n$ given by the cartesian structure. We take $-\!\!\bullet\!\!\!<_X$, the symmetries $\times_{X,Y}$ and the identities $-_X$ as atomic for all objects X,Y, i.e. the set of atomic morphisms is $\mathscr{A} = \{-\!\!\bullet\!\!\!<_X, \times_{X,Y}, -_X \;:\; X,Y \in \mathscr{C}_\mathsf{X}\}$. The weight function is $\mathsf{w}(-\!\!\bullet\!\!\!<_X) := 2\cdot\mathsf{w}(X)$, $\mathsf{w}(\times_{X,Y}) := \mathsf{w}(X)+\mathsf{w}(Y)$ and $\mathsf{w}(-_X) := \mathsf{w}(X)$. In a prop, we take $\mathsf{w}(n) := n$. Note that $\mathsf{w}(-\!\!\bullet\!\!\!<_{X\otimes Y}) = 2\cdot\mathsf{w}(X\otimes Y) = 2\cdot(\mathsf{w}(X)+\mathsf{w}(Y))$, but utilising coherence we can do better, as illustrated below.

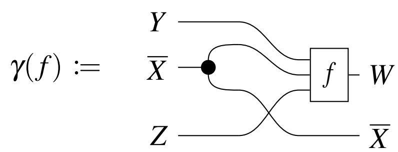
Example 2.8. *Let C be a prop with coherent copying and consider $-\!\!\bullet\!\!\!<_n \colon n \to 2n$. Let $\gamma_{n,m} := (-\!\!\bullet\!\!\!<_n \otimes -_m) \,;(-_n \otimes \times_{n,m}) \colon n+m \to n+m+n$. We can decompose $\gamma_{n,m}$ in terms of $\gamma_{n-1,m+1}$ (in the dashed box), $-\!\!\bullet\!\!\!<_1$ and $\times_{1,1}$ by cutting along at most $n+1+m$ wires:*



This allows us to decompose $-\!\bullet\!\!\subset_n = \gamma_{n,0}$ *cutting along at most* $n+1$ *wires. In particular,* $\mathsf{mwd}(-\!\bullet\!\!\subset_n) \leq n+1$.

The following result is a technical generalisation of the argument presented in Example 2.8.

Lemma 2.9 ([22])**.** *Let* X *be a symmetric monoidal category with coherent copying. Suppose that* $\mathscr{A}$ *contains* $-\!\bullet\!\!\subset_X$ *for* $X \in \mathscr{C}_{\mathsf{X}}$, *and* $\times_{X,Y}$ *and* $—_X$ *for* $X \in \mathsf{Obj}(\mathsf{X})$. *Let* $\overline{X} := X_1 \otimes \cdots \otimes X_n$, $f \colon Y \otimes \overline{X} \otimes Z \to W$ *and let* $d \in D_f$. *Let* $\gamma(f) := (—_Y \otimes -\!\bullet\!\!\subset_{\overline{X}} \otimes —_Z) ; (—_{Y \otimes \overline{X}} \otimes \times_{\overline{X},Z}) ; (f \otimes —_{\overline{X}})$.



There is a decomposition $\mathscr{C}(d)$ *of* $\gamma(f)$ *of bounded width:*

$$\mathsf{wd}(\mathscr{C}(d)) \leq \max\{\mathsf{wd}(d), \mathsf{w}(Y) + \mathsf{w}(Z) + (n+1) \cdot \max_{i=1,\dots,n} \mathsf{w}(X_i)\}.$$

3 Monoidal width in matrices

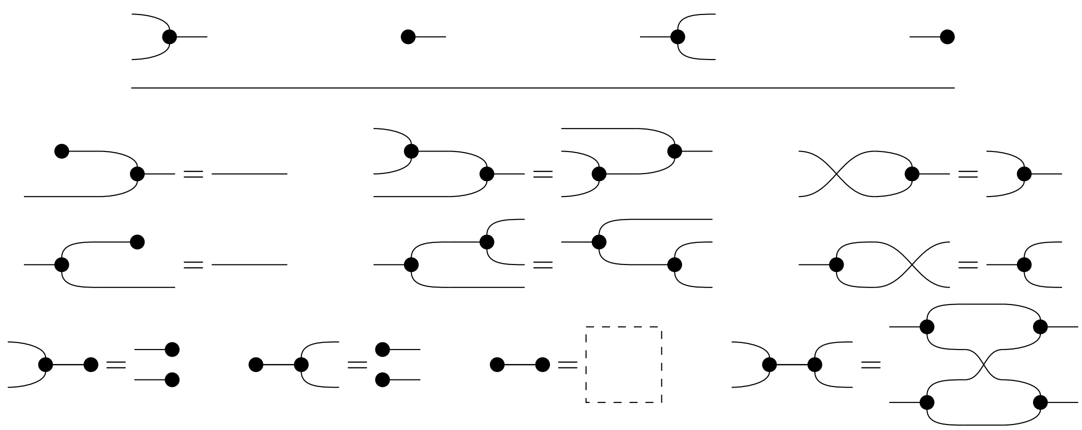


Figure 4: Bialgebra axioms

Given the ubiquity of matrix algebra, matrices are an obvious case study. Theorem 3.12 shows that the monoidal width of a matrix is, up to 1, the maximum of the ranks of its blocks.

Consider the monoidal category $\mathsf{Mat}_{\mathbb{N}}$ of matrices with entries in the natural numbers. The objects are natural numbers and morphisms from n to m are m by n matrices. Composition is the usual product of matrices and the monoidal product is the biproduct: $A \otimes B := \left(\begin{smallmatrix} A & \mathbb{0} \\ \mathbb{0} & B \end{smallmatrix}\right)$. Let us examine matrix decompositions enabled by this algebra. A matrix A can be written as a monoidal product $A = A_1 \otimes A_2$ iff the matrix has blocks A_1 and A_2, i.e. $A = \begin{pmatrix} A_1 & \mathbb{0} \\ \mathbb{0} & A_2 \end{pmatrix}$. On the other hand, a composition is related to the rank.

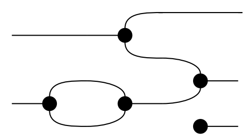
Lemma 3.1 ([34]). *Let* $A\colon n \to m$ *in* $\mathsf{Mat}_{\mathbb{N}}$. *Then* $\min\{k \in \mathbb{N} : A = B\,;_k C\} = \mathsf{rank}(A)$.

We first introduce a convenient syntax for matrices.

Proposition 3.2 ([41]). *The category* $\mathsf{Mat}_{\mathbb{N}}$ *is isomorphic to the prop* Bialg, *generated by* $-\!\!\bullet\!\!\!<\colon 1 \to 2$, $-\!\!\bullet\colon 1 \to 0$, $>\!\!\!\bullet\!\!-\colon 2 \to 1$ *and* $\bullet\!\!-\colon 0 \to 1$ *and quotiented by bialgebra axioms (Figure 4).*

For the uninitiated reader, let us briefly explain this correspondence. Every morphism $f\colon n \to m$ in Bialg corresponds to a matrix $A = \mathsf{Mat}(f) \in \mathsf{Mat}_{\mathbb{N}}(m,n)$: we can read the (i,j)-entry of A off the diagram of f by counting the number of paths from the jth input to the ith output.

Example 3.3. *The matrix* $\begin{pmatrix} 1 & 0 \\ 1 & 2 \\ 0 & 0 \end{pmatrix} \in \mathsf{Mat}_{\mathbb{N}}(3,2)$ *corresponds to*



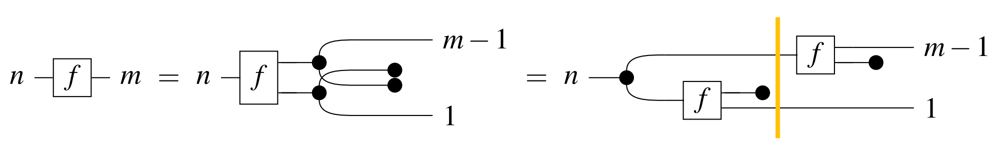
Definition 3.4. The atomic morphisms $\mathscr{A}$ are the generators of Bialg, with the symmetry and identity on 1: $\mathscr{A} = \{-\!\!\bullet\!\!\!<, -\!\!\bullet, >\!\!\!\bullet\!\!-, \bullet\!\!-, \times, -_1\}$. The weight $\mathsf{w}\colon \mathscr{A} \cup \{\otimes\} \cup \mathsf{Obj}(\mathsf{Bialg}) \to \mathbb{N}$ has $\mathsf{w}(n) := n$, for any $n \in \mathbb{N}$, and $\mathsf{w}(g) := \max\{m,n\}$, for $g\colon n \to m \in \mathscr{A}$.

3.1 Monoidal width in Bialg

The characterisation of the rank of a matrix in Lemma 3.1 hints at some relationship between the monoidal width of a matrix and its rank. In fact, we have Proposition 3.7, which bounds the monoidal width of a matrix with its rank. In order to prove this result, we first need to bound the monoidal width of a matrix with its domain and codomain, which is done in Proposition 3.5.

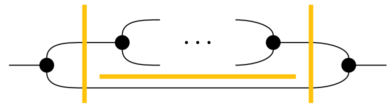
Proposition 3.5. *Let* P *be a cartesian and cocartesian prop. Suppose that* $-_1, -\!\!\bullet\!\!\!<_1, >\!\!\!\bullet\!\!-_1, -\!\!\bullet_1, \bullet\!\!-_1 \in \mathscr{A}$ *and* $\mathsf{w}(-_1) \leq 1$, $\mathsf{w}(-\!\!\bullet\!\!\!<_1) \leq 2$, $\mathsf{w}(>\!\!\!\bullet\!\!-_1) \leq 2$, $\mathsf{w}(-\!\!\bullet_1) \leq 1$ *and* $\mathsf{w}(\bullet\!\!-_1) \leq 1$. *Suppose that, for every* $g\colon 1 \to 1$, $\mathsf{mwd}(g) \leq 2$. *Let* $f\colon n \to m$ *be a morphism in* P. *Then* $\mathsf{mwd}(f) \leq \min\{m,n\} + 1$.

Proof sketch. The proof proceeds by induction on $\max\{m,n\}$. The base cases are easily checked. The inductive step relies on the fact that, applying Lemma 2.9, if $n < m$, we can decompose f as shown below by cutting at most $n+1$ wires or, if $m < n$, in the symmetric way by cutting at most $m+1$ wires.



□

We can apply the former result to Bialg and obtain Proposition 3.7 because the width of 1×1 matrices, which are numbers, is at most 2. This follows from the reasoning in Example 2.6 as we can write every natural number $k\colon 1 \to 1$ as the following composition:



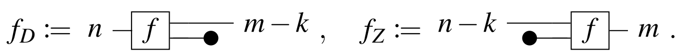
Lemma 3.6. *Let* $k\colon 1 \to 1$ *in* Bialg. *Then,* $\mathsf{mwd}(k) \leq 2$.

Proposition 3.7. *Let* $f\colon n \to m$ *in* Bialg. *Then,* $\mathsf{mwd}\, f \leq \mathsf{rank}(\mathsf{Mat}\, f) + 1$. *Moreover, if f is not $\otimes$-decomposable, i.e. there are no f_1, f_2 both distinct from f s.t. $f = f_1 \otimes f_2$, then* $\mathsf{rank}(\mathsf{Mat}\, f) \leq \mathsf{mwd}\, f$.

Proof sketch. This result follows from Lemma 3.1 and Proposition 3.5, which we can apply thanks to Lemma 3.6. □

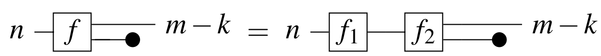
The bounds given by Proposition 3.7 can be improven when we have a $\otimes$-decomposition of a matrix, i.e. we can write $f = f_1 \otimes \ldots \otimes f_k$, to obtain Proposition 3.9. The latter relies on Lemma 3.8, which shows that discarding inputs or outputs cannot increase the monoidal width of a morphism in Bialg.

Lemma 3.8. *Let $f\colon n \to m$ in* Bialg *and $d \in D_f$. Let $f_D := f\,;(—_{m-k} \otimes -\!\bullet_k)$ and $f_Z := (—_{n-k'} \otimes \bullet\!-_{k'})\,; f$, where $-\!\bullet_k\colon k \to 0$ is the discard morphism with $k \leq m$ and $\bullet\!-_{k'}\colon 0 \to k$ is the zero morphism with $k' \leq n$.*



Then there are $\mathscr{D}(d) \in D_{f_D}$ and $\mathscr{Z}(d) \in D_{f_Z}$ such that $\mathsf{wd}(\mathscr{D}(d)) \leq \mathsf{wd}(d)$ *and* $\mathsf{wd}(\mathscr{Z}(d)) \leq \mathsf{wd}(d)$.

Proof sketch. By induction. The base cases are easy. If $f = f_1\,; f_2$, use the inductive hypothesis on f_2.



The $f = f_1 \otimes f_2$ case is similar. □

Proposition 3.9. *Let $f\colon n \to m$ in* Bialg *and $d' = (d'_1, ;_k, d'_2) \in D_f$. Suppose there are f_1 and f_2 such that $f = f_1 \otimes f_2$. Then, there is $d = (d_1, \otimes, d_2) \in D_f$ such that* $\mathsf{wd}(d) \leq \mathsf{wd}(d')$.

Proof sketch. By Lemma 3.1, $\mathsf{rank}(\mathsf{Mat}\, f_1) + \mathsf{rank}(\mathsf{Mat}\, f_2) = \mathsf{rank}(\mathsf{Mat}(f_1 \otimes f_2)) \leq k$ and, by Proposition 3.7, there is a monoidal decomposition d_i of f_i such that $\mathsf{wd}(d_i) \leq \mathsf{rank}(\mathsf{Mat}\, f_i) + 1$. Then, $\mathsf{wd}(d) := \mathsf{wd}((d_1, \otimes, d_2)) \leq \max\{\mathsf{rank}(\mathsf{Mat}\, f_1), \mathsf{rank}(\mathsf{Mat}\, f_2)\} + 1 \leq \mathsf{rank}(\mathsf{Mat}\, f_1) + \mathsf{rank}(\mathsf{Mat}\, f_2)$ whenever $\mathsf{rank}(\mathsf{Mat}\, f_1), \mathsf{rank}(\mathsf{Mat}\, f_2) > 0$. We apply Lemma 3.8 to obtain the same result if $\mathsf{rank}(\mathsf{Mat}\, f_1) = 0$ or $\mathsf{rank}(\mathsf{Mat}\, f_2) = 0$. □

We summarise Proposition 3.9 and Proposition 3.7 in Corollary 3.10.

Corollary 3.10. *Let $f = f_1 \otimes \ldots \otimes f_k$ in* Bialg. *Then,* $\mathsf{mwd}(f) \leq \max_{i=1,\ldots,k} \mathsf{rank}(\mathsf{Mat}(f_i)) + 1$. *Moreover, if f_i are not $\otimes$-decomposable, then* $\max_{i=1,\ldots,k} \mathsf{rank}(\mathsf{Mat}(f_i)) \leq \mathsf{mwd}\, f$.

Proof. By Proposition 3.9 there is a decomposition of f of the form $d = (d_1, \otimes, \cdots (d_{k-1}, \otimes, d_k))$, where we can choose d_i to be a minimal decomposition of f_i. Then, $\mathsf{mwd}(f) \leq \mathsf{wd}(d) = \max_{i=1,\ldots,k} \mathsf{wd}(d_i)$. By Proposition 3.7, $\mathsf{wd}(d_i) \leq r_i + 1$. Then, $\mathsf{mwd}(f) \leq \max\{r_1, \ldots, r_k\} + 1$. Moreover, if f_i are not $\otimes$-decomposable, Proposition 3.7 gives also a lower bound on their monoidal width: $\mathsf{rank}(\mathsf{Mat}(f_i)) \leq \mathsf{mwd}\, f_i$; and we obtain that $\max_{i=1,\ldots,k} \mathsf{rank}(\mathsf{Mat}(f_i)) \leq \mathsf{mwd}\, f$. □

The results so far show a way to construct efficient decompositions given a $\otimes$-decomposition of the matrix. However, we do not know whether $\otimes$-decompositions are unique. Proposition 3.11 shows that every morphism in Bialg has a unique $\otimes$-decomposition.

Proposition 3.11. *Let* C *be a monoidal category whose monoidal unit* 0 *is both initial and terminal, and whose objects are a unique factorization monoid. Let* f *be a morphism in* C*. Then* f *has a unique* $\otimes$*-decomposition.*

Our main result in this section follows from Corollary 3.10 and Proposition 3.11, which can be applied to Bialg because 0 is both terminal and initial, and the objects, being a free monoid, are a unique factorization monoid.

Theorem 3.12. *Let* $f = f_1 \otimes \ldots \otimes f_k$ *be a morphism in* Bialg *and its unique* $\otimes$*-decomposition given by Proposition 3.11, with* $r_i = \mathsf{rank}(\mathsf{Mat}(f_i))$*. Then* $\max\{r_1,\ldots,r_k\} \leq \mathsf{mwd}(f) \leq \max\{r_1,\ldots,r_k\}+1$.

Proof. This result is obtained by applying Corollary 3.10 to the $\otimes$-decomposition given by Proposition 3.11, which can be applied because, in Bialg, 0 is both terminal and initial, and the objects, being a free monoid, are a unique factorization monoid. ☐

Note that the identity matrix has monoidal width 1 and twice the identity matrix has monoidal width 2, attaining both the upper and lower bounds for the monoidal width of a matrix.

4 Graphs and rank width

Here we recall rank width [33] for undirected graphs.

Definition 4.1. An *undirected graph* $G = (V, E, \mathsf{ends})$ is given by a set of edges E, a set of vertices V and a function $\mathsf{ends}\colon E \to \wp_{\leq 2}(V)$ that gives the endpoints of each edge. We consider graphs *up to isomorphism*, or *abstract graphs*, thus the set of vertices can be fully characterised by its cardinality. An abstract graph can be equivalently given by an adjacency matrix $[G]$, where $G \in \mathsf{Mat}_{\mathbb{N}}(n,n)$ and n is the number of vertices. The equivalence class of adjacency matrices is defined by the equivalence relation

$$G \sim H \quad \text{iff} \quad G + G^{\top} = H + H^{\top}.$$

We will refer to abstract undirected graphs as simply graphs.

Definition 4.2. A *path* in a graph G is a sequence of edges $(e_1,\ldots,e_k)$ together with a sequence of distinct vertices $(v_1,\ldots,v_{k+1})$ of G such that, for every $i = 1,\ldots,k$, $\mathsf{ends}(e_i) = \{v_i, v_{i+1}\}$. A *tree* is a graph such that there is a unique path between any two of its vertices. Two vertices v and w in a graph G are *neighbours* if G has an edge between them. The *leaves* of a tree are those vertices with at most one neighbour. A *subcubic tree* is a tree where each vertex has between one and three neighbours.

A rank decomposition for a graph G is a tree whose leaves are labelled with the vertices of G.

Definition 4.3 ([33])**.** A *rank decomposition* (Y, r) of a graph G is given by a subcubic tree Y together with a bijection $r\colon \mathsf{leaves}(Y) \to \mathsf{vertices}(G)$.

Each edge b in the tree Y determines a splitting of the graph: it determines a two partition of the leaves of Y, which, through r, determines a two partition $\{A_b, B_b\}$ of the vertices of G. This corresponds to a splitting of the graph G into two subgraphs G_1 and G_2. Intuitively, the order of an edge b is the amount of information required to recover G by joining G_1 and G_2. Given the partition $\{A_b, B_b\}$ of the vertices of G, we can record the edges in G beween A_b and B_b in a matrix X_b. This means that, if $v_i \in A_b$ and $v_j \in B_b$, the entry (i, j) of the matrix X_b is the number of edges between v_i and v_j.

Definition 4.4 (Order of an edge)**.** Let (Y, r) be a rank decomposition of a graph G. Let b be an edge of Y. The order of b is the rank of the matrix associated to it: $\mathsf{ord}(b) := \mathsf{rank}(X_b)$.

Note that the order of the two sets in the partition does not matter as the rank is invariant to transposition. The width of a rank decomposition is the maximum order of the edges of the tree and the rank width of a graph is the width of its cheapest decomposition.

Definition 4.5 (Rank width)**.** Given a rank decomposition (Y,r) of a graph G, define its width as $\mathsf{wd}(Y,r) := \max_{b \in \mathsf{edges}(Y)} \mathsf{ord}(b)$. The *rank width* of G is given by the min-max formula:

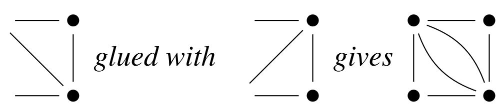
$$\mathsf{rwd}(G) := \min_{(Y,r)} \mathsf{wd}(Y,r).$$

4.1 Graphs with dangling edges

As intermediate step between rank decompositions and monoidal decompositions, we introduce recursive rank decompositions of *graphs with dangling edges* and we prove that they give a notion of width that is equivalent to rank width. Similar recursive characterisations were done for tree decompositions in [4] and for path and branch decompositions in [22]. We first need a notion of graph that is equipped with some "open" edges along which it can be glued with other graphs.

Definition 4.6. A *graph with dangling edges* $\Gamma = ([G], B)$ is given by an adjacency matrix $G \in \mathsf{Mat}_{\mathbb{N}}(k,k)$ that records the connectivity of the graph and a matrix $B \in \mathsf{Mat}_{\mathbb{N}}(k,n)$ that records the "dangling edges" connected to n boundary ports. We will sometimes write $G \in \mathsf{adjacency}(\Gamma)$ and $B = \mathsf{boundary}(\Gamma)$.

Example 4.7. *Two graphs with the same ports, as illustrated below, can be "glued" together:*



Decompositions are elements of a tree data type, with nodes carrying subgraphs Γ' of the ambient graph Γ. In the following Γ' ranges over the non-empty subgraphs of Γ: $T_\Gamma ::= (\Gamma') \mid (T_\Gamma, \Gamma', T_\Gamma)$. Given $T \in T_\Gamma$, the label function λ takes a decomposition and returns the graph with dangling edges at the root: $\lambda(T_1, \Gamma, T_2) := \Gamma$ and $\lambda(\Gamma) := \Gamma$.

Definition 4.8 (Recursive rank decomposition)**.** Let $\Gamma = ([G], B)$ be a graph with dangling edges, where $G \in \mathsf{Mat}_{\mathbb{N}}(k,k)$ and $B \in \mathsf{Mat}_{\mathbb{N}}(k,n)$. A recursive rank decomposition of Γ is $T \in T_\Gamma$ where either: Γ has at most one vertex and $T = (\Gamma)$; or $T = (T_1, \Gamma, T_2)$ and $T_i \in T_{\Gamma_i}$ are recursive rank decompositions of subgraphs $\Gamma_i = ([G_i], B_i)$ of Γ such that:

- The vertices are partitioned in two, $[G] = \left[\begin{pmatrix} G_1 & C \\ 0 & G_2 \end{pmatrix}\right]$;
- The dangling edges are those to the original boundary and to the other subgraph, $B_1 = (A_1 \mid C)$ and $B_2 = (A_2 \mid C^\top)$, where $B = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$.

As with before, the *recursive* rank width of a graph is the width of its cheapest decomposition.

Definition 4.9. Let T be a recursive rank decomposition of $\Gamma = ([G], B)$. Define the width of T recursively: if $T = (\Gamma)$, $\mathsf{wd}(T) := \mathsf{rank}(B)$, and, if $T = (T_1, \Gamma, T_2)$, $\mathsf{wd}(T) := \max\{\mathsf{wd}(T_1), \mathsf{wd}(T_2), \mathsf{rank}(B)\}$ Expanding this expression, we obtain $\mathsf{wd}(T) = \max_{T' \text{ subtree of } T} \mathsf{rank}(\mathsf{boundary}(\lambda(T')))$. The *recursive rank width* of Γ is defined by the min-max formula $\mathsf{rrwd}(\Gamma) := \min_T \mathsf{wd}(T)$.

We show that recursive rank width is the same as rank width, up to the rank of the boundary of the graph.

Proposition 4.10. *Let $\Gamma = ([G], B)$ be a graph with dangling edges and (Y,r) be a rank decomposition of G. Then, there is a recursive rank decomposition $\mathscr{I}(Y,r)$ of Γ s.t.* $\mathsf{wd}(\mathscr{I}(Y,r)) \leq \mathsf{wd}(Y,r) + \mathsf{rank}(B)$.

Before proving the lower bound for recursive rank width, we need a technical lemma that relates the width of a graph with that of its subgraphs.

Lemma 4.11. *Let T be a recursive rank decomposition of $\Gamma = ([G], B)$. Let T' be a subtree of T and $\Gamma' := \lambda(T')$ with $\Gamma' = ([G'], B')$. The adjacency matrix of Γ can be written as* $[G] = \left[\begin{pmatrix} G_L & C_L & C \\ \mathbb{0} & G' & C_R \\ \mathbb{0} & \mathbb{0} & G_R \end{pmatrix}\right]$ *and its boundary as* $B = \begin{pmatrix} A_L \\ A' \\ A_R \end{pmatrix}$. *Then,* $\mathsf{rank}(B') = \mathsf{rank}(A' \mid C_L^\top \mid C_R)$.

Proposition 4.12. *Let T be a recursive rank decomposition of $\Gamma = ([G], B)$ with $G \in \mathsf{Mat}_{\mathbb{N}}(k,k)$ and $B \in \mathsf{Mat}_{\mathbb{N}}(k,n)$. Then, there is a rank decomposition $\mathscr{I}^\dagger(T)$ of G such that* $\mathsf{wd}(\mathscr{I}^\dagger(T)) \leq \mathsf{wd}(T)$.

From Proposition 4.12 and Proposition 4.10 we conclude the following result.

Theorem 4.13. *Let $\Gamma = ([G], B)$. Then,* $\mathsf{rwd}(G) \leq \mathsf{rrwd}(\Gamma) \leq \mathsf{rwd}(G) + \mathsf{rank}(B)$.

5 Monoidal width and rank width

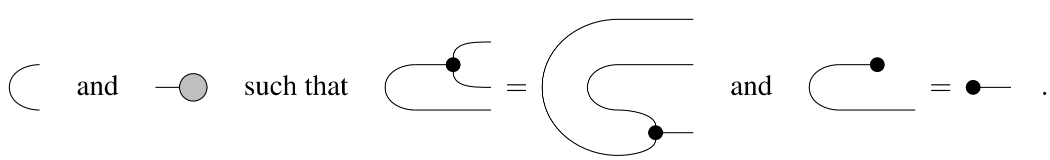
This section contains our main results. We prove that monoidal width in the prop of graphs Grph [14] corresponds to rank width, up to a constant multiplicative factor of 2.

We start by introducing the algebra of graphs with boundaries and its diagrammatic syntax [21]. A graph with boundaries is a graph together with two matrices L and R that record the connectivity of the vertices with the left and right boundary, a matrix P that records the passing wires from the left boundary to the right one and a matrix F that records the wires from the right boundary to itself.

Definition 5.1 ([21])**.** A *graph with boundaries* $g\colon n \to m$ is defined as $g = ([G], L, R, P, [F])$, where $[G]$ is the adjacency matrix of a graph on k vertices, with $G \in \mathsf{Mat}_{\mathbb{N}}(k,k)$; $L \in \mathsf{Mat}_{\mathbb{N}}(k,n)$, $R \in \mathsf{Mat}_{\mathbb{N}}(k,m)$, $P \in \mathsf{Mat}_{\mathbb{N}}(m,n)$ and $F \in \mathsf{Mat}_{\mathbb{N}}(m,m)$ recording connectivity information as explained above. Graphs with boundaries are taken up to an equivalence making the order of the vertices immaterial. Let $g, g'\colon n \to m$ on k vertices, with $g = ([G], L, R, P, [F])$ and $g' = ([G'], L', R', P, [F])$. The graphs g and g' are considered equal iff there is a permutation matrix $\sigma \in \mathsf{Mat}_{\mathbb{N}}(k,k)$ such that $g' = ([\sigma G \sigma^\top], \sigma L, \sigma R, P, [F])$.

Graphs with boundaries can be composed sequentially and in parallel [21], forming a symmetric monoidal category BGraph. The prop Grph provides a convenient syntax for graphs with boundaries. It is obtained by adding a cup and a vertex generators to the prop of matrices Bialg (Figure 4).

Definition 5.2 ([14])**.** The prop of graphs Grph is obtained by adding to Bialg the generators $\cup\colon 0 \to 2$ and $v\colon 1 \to 0$ with the equations below.



These equations mean, in particular, that the cup transposes matrices (Figure 5, left) and that we can express the equivalence relation of adjacency matrices: $G \sim H$ iff $G + G^\top = H + H^\top$ (Figure 5, right).

Proposition 5.3 ([21], Theorem 23)**.** *The prop of graphs* Grph *is isomorphic to the prop* BGraph.

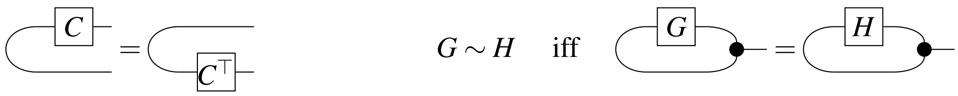
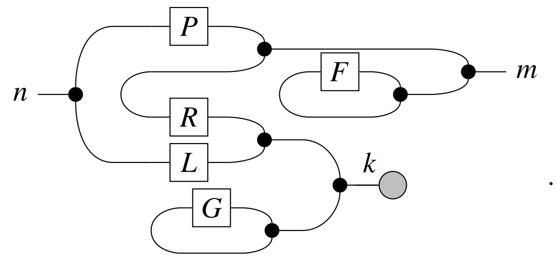


Figure 5: Adding the cup.

Proposition 5.3 means that the morphisms in Grph can be written in the following normal form

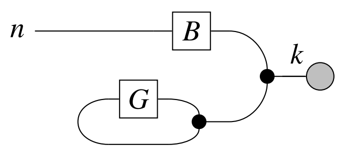


.

The prop Grph is more expressive than graphs with dangling edges (Definition 4.6): its morphisms can have edges between the boundaries as well. In fact, graphs with dangling edges can be seen as morphisms $n \to 0$ in Grph.

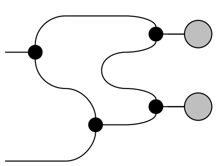
Example 5.4. *A graph with dangling edges $\Gamma = ([G], B)$ can be represented as a morphism in* Grph

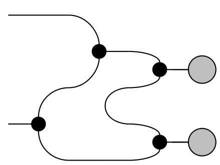
$$g = ([G], B, ¡, !, [()]) =$$

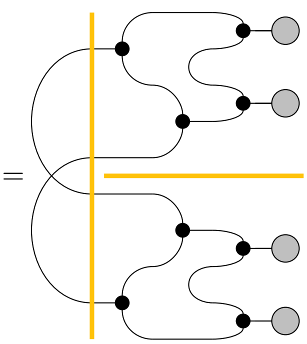


.

We can now formalise the intuition of glueing graphs with dangling edges as explained in Example 4.7. The two graphs there correspond to g_1 and g_2 below left and middle. Their glueing is obtained by precomposing their monoidal product with a cup, i.e. $\cup_2 ; (g_1 \otimes g_2)$, as shown below right.

$g_1 =$ 

$g_2 =$ 

$\cup_2 ; (g_1 \otimes g_2) =$ 

5.1 Rank width in open graphs

The technical content of our main result (Theorem 5.12) is split in two: an upper and a lower bound.

As in the prop of matrices Bialg, the cost of composing along n wires is n. All morphisms in Grph are chosen as atomic. One could restrict this to those with at most one vertex without affecting the results.

Definition 5.5. Let the set of *atomic morphisms* $\mathscr{A}$ be the set of all the morphisms of Grph. The *weight function* $\mathsf{w} \colon \mathscr{A} \cup \{\otimes\} \cup \mathsf{Obj}(\mathsf{Grph}) \to \mathbb{N}$ is defined, on objects n, as $\mathsf{w}(n) := n$; and, on morphisms $g \in \mathscr{A}$, as $\mathsf{w}(g) := k$, where k is the number of vertices of g.

Note that the monoidal width of g is bounded by the number of its vertices.

The upper bound (Proposition 5.8) is established by associating to each recursive rank decomposition a suitable monoidal decomposition. This mapping is defined inductively, given the inductive nature of both these structures. Given a recursive rank decomposition of a graph Γ, we can construct a decomposition of its corresponding morphism g as shown by the first equality in Figure 6. However, this

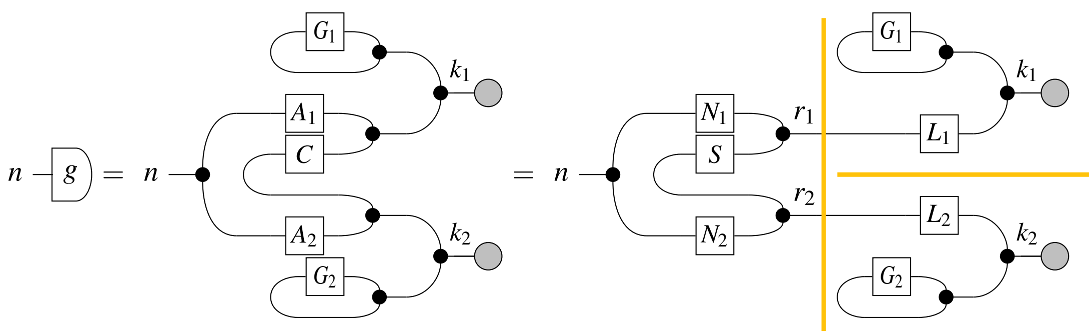
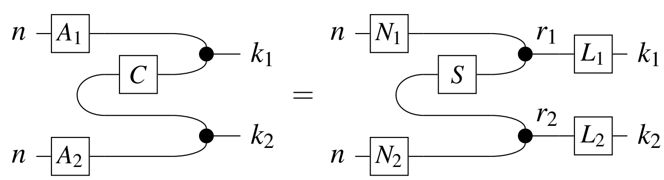


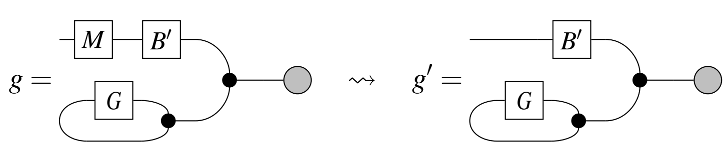
Figure 6: First step of a monoidal decomposition given by a recursive rank decomposition

decomposition is not optimal as it cuts along the number of vertices $k_1 + k_2$. But we can do better thanks to Lemma 5.6, which shows that we can cut along the ranks, $r_1 = \mathsf{rank}(A_1 \mid C)$ and $r_2 = \mathsf{rank}(A_2 \mid C^\top)$, of the boundaries of the induced subgraphs to obtain the second equality in Figure 6.



Lemma 5.6. *Let $A_i \in \mathsf{Mat}_{\mathbb{N}}(k_i, n)$, for $i = 1,2$, and $C \in \mathsf{Mat}_{\mathbb{N}}(k_1, k_2)$. Then, there are rank decompositions of $(A_1 \mid C)$ and $(A_2 \mid C^\top)$ of the form $(A_1 \mid C) = L_1 \cdot (N_1 \mid S \cdot L_2^\top)$, and $(A_2 \mid C^\top) = L_2 \cdot (N_2 \mid S^\top \cdot L_1^\top)$.*

Once we have performed the cuts in Figure 6 on the right, we have changed the boundaries of the induced subgraphs. This means that we cannot apply the inductive hypothesis right away, but we need to transform first the recursive rank decompositions of the old subgraphs into decompositions of the new ones, as shown in Lemma 5.7. More explicitly, when M has full rank, if we have a recursive rank decomposition of $\Gamma = ([G], B' \cdot M)$, which corresponds to g below left, we can obtain one of $\Gamma' = ([G], B')$, which corresponds to g' below right, of the same width.



Lemma 5.7. *Let T be a recursive rank decomposition of $\Gamma = ([G], B)$ and $B = B' \cdot M$, with M that has full rank. Then, there is a recursive rank decomposition T' of $\Gamma' = ([G], B')$ such that $\mathsf{wd}(T) = \mathsf{wd}(T')$ and such that T and T' have the same underlying tree structure.*

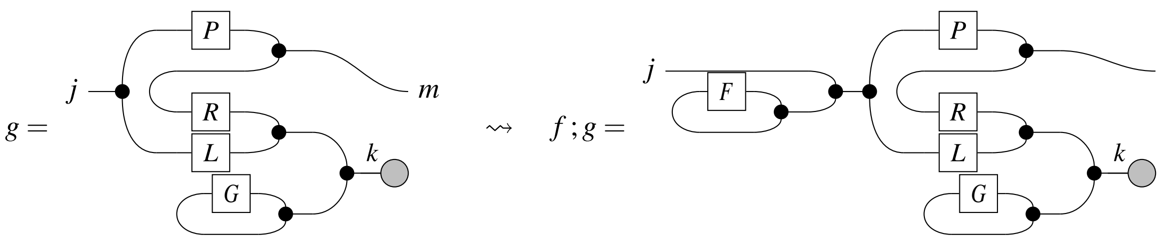
With the above ingredients, we can show that rank width bounds monoidal width from above.

Proposition 5.8. *Let* $\Gamma = ([G], B)$ *be a graph with dangling edges and* $g\colon n \to 0$ *be the morphism in* Grph *corresponding to* Γ*. Let* T *be a recursive rank decomposition of* Γ*. Then, there is a monoidal decomposition* $\mathscr{R}^\dagger(T)$ *of* g *such that* $\mathsf{wd}(\mathscr{R}^\dagger(T)) \leq 2 \cdot \mathsf{wd}(T)$*.*

Proof sketch. The proof proceeds by induction on T. The base cases are easily checked and the inductive step relies on the decomposition of g in Figure 6, which we can write thanks to Lemma 5.6. Applying the inductive hypothesis and Lemma 5.7, the width of this decomposition can be bounded by $\max\{r_1 + r_2, 2 \cdot \mathsf{wd}(T_1), 2 \cdot \mathsf{wd}(T_2)\} \leq 2 \cdot \mathsf{wd}(T)$, where $T = (T_1, \Gamma, T_2)$. □

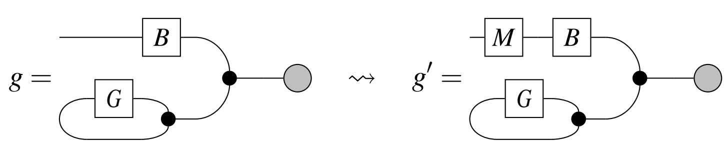
Proving the lower bound is similarly involved and follows a similar proof structure. From a monoidal decomposition we construct inductively a recursive rank decomposition of bounded width. The inductive step relative to composition nodes is the most involved and needs two additional lemmas, which allow us to transform recursive rank decompositions of the induced subgraphs into ones of two subgraphs that satisfy the conditions of Definition 4.8.

Applying the inductive hypothesis gives us a recursive rank decomposition of $\Gamma = ([G], (L \mid R))$, which is associated to g below left, and we need to construct one of $\Gamma' := (\left[G + L \cdot F \cdot L^\top\right], (L \mid R + L \cdot (F + F^\top) \cdot P^\top))$, which is associated to $f \,;g$ below right, of at most the same width.



Lemma 5.9. *Let* T *be a recursive rank decomposition of* $\Gamma = ([G], (L \mid R))$*. Let* $F \in \mathsf{Mat}_{\mathbb{N}}(j, j)$*,* $P \in \mathsf{Mat}_{\mathbb{N}}(m, j)$ *and define* $\Gamma' := (\left[G + L \cdot F \cdot L^\top\right], (L \mid R + L \cdot (F + F^\top) \cdot P^\top))$*. Then, there is a recursive rank decomposition* T' *of* Γ' *of bounded width:* $\mathsf{wd}(T') \leq \mathsf{wd}(T)$*.*

In order to obtain the subgraphs of the desired shape we need to add some extra connections to the boundaries. We have a recursive rank decomposition of $\Gamma = ([G], B)$, which corresponds to g below left, and we need one of $\Gamma' = ([G], B \cdot M)$, which corresponds to g' below right, of at most the same width.



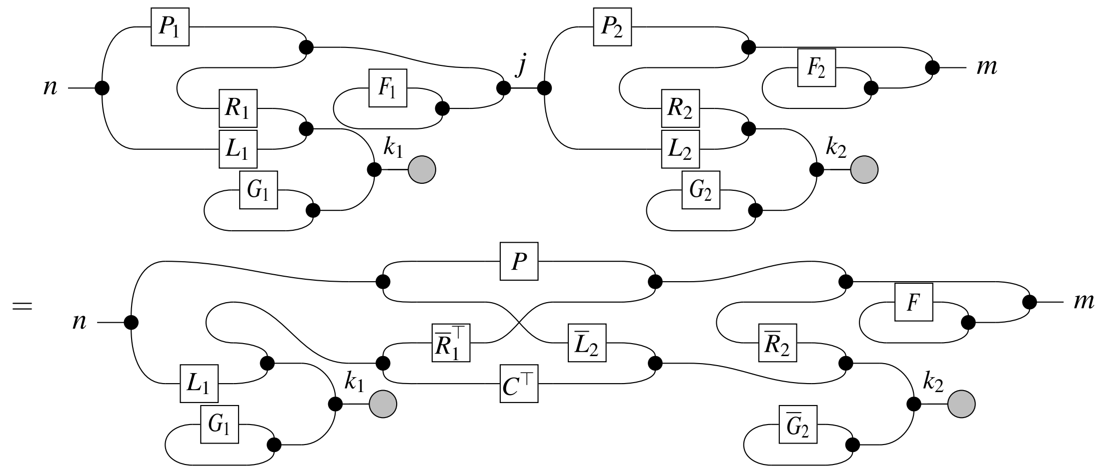
The following result and its proof are very similar to Lemma 5.7.

Lemma 5.10. *Let* T *be a recursive rank decomposition of* $\Gamma = ([G], B)$ *and let* $B' = B \cdot M$*. Then, there is a recursive rank decomposition* T' *of* $\Gamma' = ([G], B')$ *such that* $\mathsf{wd}(T') \leq \mathsf{wd}(T)$ *and such that* T *and* T' *have the same underlying tree structure. Moreover, if* M *has full rank, then* $\mathsf{wd}(T') = \mathsf{wd}(T)$*.*

Proposition 5.11. *Let* $g = ([G], L, R, P, [F])$ *in* Grph *and* $d \in D_g$*. Let* $\Gamma = ([G], (L \mid R))$*. Then, there is a recursive rank decomposition* $\mathscr{R}(d)$ *of* Γ *s.t.* $\mathsf{wd}(\mathscr{R}(d)) \leq 2 \cdot \max\{\mathsf{wd}(d), \mathsf{rank}(L), \mathsf{rank}(R)\}$*.*

Proof sketch. The proof proceeds by induction on d. The base case is easily checked, while the inductive steps are a bit more involved. If $d = (d_1, ;_j, d_2)$, then there are $g_i = ([G_i], L_i, R_i, P_i, [F_i])$ such that $g =$

$g_1\,;g_2$ and we can write g as follows.



In order to build a recursive rank decomposition of Γ, we need recursive rank decompositions of $\overline{\Gamma}_i = (\left[\overline{G}_i\right], \overline{B}_i)$, but we can obtain recursive rank decompositions of $\Gamma_i = ([G_i], (L_i \mid R_i))$ by applying only induction. Thanks to Lemma 5.9, we obtain a recursive rank decomposition of $\Gamma'_2 = (\left[G_2 + L_2 \cdot F_1 \cdot L_2^\top\right], (L_2 \mid R_2 + L_2 \cdot (F_1 + F_1^\top) \cdot P_2^\top))$. Lastly, we apply Lemma 5.10 to get recursive rank decompositions T_i of $\overline{\Gamma}_i$. Thanks to these, we can bound the width of $T := (T_1, \Gamma, T_2)$:

$$\mathsf{wd}(T) \leq 2 \cdot \max\{\mathsf{wd}(d_1), \mathsf{wd}(d_2), j, \mathsf{rank}(L), \mathsf{rank}(R)\} =: 2 \cdot \max\{\mathsf{wd}(d), \mathsf{rank}(L), \mathsf{rank}(R)\}.$$

If $d = (d_1, \otimes, d_2)$, we proceed similarly. □

From Proposition 5.8, Proposition 5.11 and Theorem 4.13, we obtain our main result.

Theorem 5.12. *Let G be a graph and let $g = ([G], ¡, ¡, (\,), [(\,)])$ be the corresponding morphism of* Grph. *Then,* $\frac{1}{2} \cdot \mathsf{rwd}(G) \leq \mathsf{mwd}(g) \leq 2 \cdot \mathsf{rwd}(G)$.

6 Conclusions and future work

We have shown that monoidal width, in a suitable category of graphs composable along "open" edges, yields rank width; a well-known measure from the graph theory literature.

Our goal with this line of research is to develop a generic, abstract "decomposition theory". We will study other graph widths like clique width [20] and twin width [12], as well as go beyond graphs: e.g. by focussing on tree width for hypergraphs and relational structures [1], branch width for matroids and widths for directed graphs. A part of "decomposition theory" means going beyond width as a mere number – in fact we believe that in each case the identification of a suitable monoidal category as an *algebra* of open graph structures is itself a worthwhile contribution. Indeed, having such an algebra means that a decomposition, rather than an ad hoc concept-specific construction, becomes more of a mathematical object in its own right. Such compositional algebras will add to the quiver of compositional structures of applied category theory; for example serving as syntax for more sophisticated applications [21].

Acknowledgements. Elena Di Lavore and Paweł Sobociński were supported by the European Union through the ESF funded Estonian IT Academy research measure (2014-2020.4.05.19-0001). This work was also supported by the Estonian Research Council grant PRG1210.

References

[1] Samson Abramsky, Anuj Dawar, and Pengming Wang. The pebbling comonad in finite model theory. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017. doi:10.1103/PhysRevLett.65.3373.

[2] D. Adolphson and T. C. Hu. Optimal linear ordering. *SIAM Journal on Applied Mathematics*, 25(3):403–423, 1973. doi:10.1137/0123021.

[3] Dorit Aharonov, Zeph Landau, and Johann Makowsky. The quantum fft can be classically simulated, 2006. arXiv:quant-ph/0611156.

[4] Stefan Arnborg, Bruno Courcelle, Andrzej Proskurowski, and Detlef Seese. An algebraic theory of graph reduction. *Journal of the ACM (JACM)*, 40(5):1134–1164, 1993. doi:10.1016/0095-8956(91)90061-N.

[5] Michel Bauderon and Bruno Courcelle. Graph expressions and graph rewritings. *Mathematical Systems Theory*, 20(1):83–127, 1987. doi:10.1007/BF01692060.

[6] Umberto Bertelè and Francesco Brioschi. On non-serial dynamic programming. *J. Comb. Theory, Ser. A*, 14(2):137–148, 1973. doi:10.1016/0097-3165(73)90016-2.

[7] Christoph Blume, HJ Sander Bruggink, Martin Friedrich, and Barbara König. Treewidth, pathwidth and cospan decompositions. *Electronic Communications of the EASST*, 41, 2011.

[8] Hans L Bodlaender. A tourist guide through treewidth. *Technical report*, 1992.

[9] Hans L Bodlaender and Arie MCA Koster. Combinatorial optimization on graphs of bounded treewidth. *The Computer Journal*, 51(3):255–269, 2008. doi:10.1093/comjnl/bxm037.

[10] Guillaume Boisseau and Paweł Sobociński. String diagrammatic electrical circuit theory. *CoRR*, abs/2106.07763, 2021. URL: https://arxiv.org/abs/2106.07763, arXiv:2106.07763.

[11] Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A survey of compositional signal flow theory. In Michael Goedicke, Erich J. Neuhold, and Kai Rannenberg, editors, *Advancing Research in Information and Communication Technology - IFIP's Exciting First 60+ Years, Views from the Technical Committees and Working Groups*, volume 600 of *IFIP Advances in Information and Communication Technology*, pages 29–56. Springer, 2021. doi:10.1007/978-3-030-81701-5_2.

[12] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width i: tractable fo model checking. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 601–612. IEEE, 2020. doi:10.1109/FOCS.2015.63.

[13] Benjamin Merlin Bumpus and Zoltan A Kocsis. Spined categories: generalizing tree-width beyond graphs. *arXiv preprint arXiv:2104.01841*, 2021.

[14] Apiwat Chantawibul and Paweł Sobociński. Towards compositional graph theory. *Electronic Notes in Theoretical Computer Science*, 319:121–136, 2015. doi:10.1016/j.entcs.2015.12.009.

[15] Kenta Cho and Bart Jacobs. Disintegration and Bayesian Inversion via String Diagrams. *Mathematical Structures in Computer Science*, pages 1–34, March 2019. arXiv:1709.00322, doi:10.1017/S0960129518000488.

[16] Maria Chudnovsky and Paul Seymour. A well-quasi-order for tournaments. *Journal of Combinatorial Theory, Series B*, 101(1):47–53, 2011. doi:10.1016/j.jctb.2010.10.003.

[17] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes - A first course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. doi:10.1017/9781316219317.

[18] Cole Comfort and Aleks Kissinger. A graphical calculus for lagrangian relations. *CoRR*, abs/2105.06244, 2021. URL: https://arxiv.org/abs/2105.06244, arXiv:2105.06244.

[19] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

[20] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000. doi:10.1016/S0166-218X(99)00184-5.

[21] Elena Di Lavore, Jules Hedges, and Paweł Sobociński. Compositional modelling of network games. In *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[22] Elena Di Lavore and Paweł Sobociński. Monoidal Width: Unifying Tree Width, Path Width and Branch Width, 2022. arXiv:2202.07582.

[23] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic simplification of quantum circuits with the zx-calculus. *Quantum*, 4:279, 2020. doi:10.22331/q-2020-06-04-279.

[24] Tomás Feder and Moshe Y Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. doi:10.1016/0196-6774(86)90023-4.

[25] Brendan Fong and David I Spivak. Seven sketches in compositionality: An invitation to applied category theory, 2018. arXiv:1803.05316.

[26] Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020. doi:10.1016/j.aim.2020.107239.

[27] Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 472–481. ACM, 2018. doi:10.1145/3209108.3209165.

[28] Dan R. Ghica, Achim Jung, and Aliaume Lopez. Diagrammatic semantics for digital circuits. In Valentin Goranko and Mads Dam, editors, *26th EACSL Annual Conference on Computer Science Logic, CSL 2017, August 20-24, 2017, Stockholm, Sweden*, volume 82 of *LIPIcs*, pages 24:1–24:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.CSL.2017.24.

[29] Rudolf Halin. S-functions for graphs. *Journal of geometry*, 8(1-2):171–186, 1976. doi:10.1007/BF01917434.

[30] André Joyal and Ross Street. The geometry of tensor calculus, i. *Advances in mathematics*, 88(1):55–112, 1991. doi:10.1016/0001-8708(91)90003-P.

[31] Stephen Lack. Composing PROPs. *Theor. App. Categories*, 13(9):147–163, 2004.

[32] Saunders Mac Lane. Categorical algebra. *Bull. Amer. Math. Soc.*, 71:40–106, 1965. doi:10.2307/1993828.

[33] Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006. doi:10.1016/j.jctb.2005.10.006.

[34] R Piziak and PL Odell. Full rank factorization of matrices. *Mathematics magazine*, 72(3):193–201, 1999. doi:10.1080/0025570X.1999.11996730.

[35] Pavel Pudlák, Vojtěch Rödl, and Petr Savickỳ. Graph complexity. *Acta Informatica*, 25(5):515–535, 1988.

[36] Julian Rathke, Paweł Sobociński, and Owen Stephens. Compositional reachability in petri nets. In *International Workshop on Reachability Problems*, pages 230–243. Springer, 2014. doi:10.1007/978-3-319-07734-5_9.

[37] Neil Robertson and Paul D. Seymour. Graph minors. I. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5.

[38] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

[39] Neil Robertson and Paul D. Seymour. Graph minors. X. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991. doi:10.1016/0095-8956(91)90061-N.

[40] Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010. doi:10.1090/conm/134/1187296.

[41] Fabio Zanasi. *Interacting Hopf Algebras - The Theory of Linear Systems*. PhD thesis, École Normale Supérieure de Lyon, 2015.

On the Pre- and Promonoidal Structure of Spacetime

James Hefford
University of Oxford, UK
james.hefford@cs.ox.ac.uk

Aleks Kissinger
University of Oxford, UK
aleks.kissinger@cs.ox.ac.uk

The notion of a joint system, as captured by the monoidal (a.k.a. tensor) product, is fundamental to the compositional, process-theoretic approach to physical theories. Promonoidal categories generalise monoidal categories by replacing the functors normally used to form joint systems with profunctors. Intuitively, this allows the formation of joint systems which may not always give a system again, but instead a generalised system given by a presheaf. This extra freedom gives a new, richer notion of joint systems that can be applied to categorical formulations of spacetime. Whereas previous formulations have relied on partial monoidal structure that is only defined on pairs of independent (i.e. spacelike separated) systems, here we give a concrete formulation of spacetime where the notion of a joint system is defined for any pair of systems as a presheaf. The representable presheaves correspond precisely to those actual systems that arise from combining spacelike systems, whereas more general presheaves correspond to virtual systems which inherit some of the logical/compositional properties of their "actual" counterparts. We show that there are two ways of doing this, corresponding roughly to relativistic versions of conjunction and disjunction. The former endows the category of spacetime slices in a Lorentzian manifold with a promonoidal structure, whereas the latter augments this structure with an (even more) generalised way to combine systems that fails the interchange law.

1 Introduction

Categorical approaches to the modelling of structures of spacetime have become increasingly rich topics of study leading to both the development of new mathematics and a greater understanding of the underlying structures of our theories of physics. Nevertheless, the precise categorical structures that should be present in a model of spacetime are far from settled. Monoidal structure is a common requirement, being a key part of Categorical Quantum Mechanics [1] and of many approaches to Topological Quantum Field Theory [2]. The key physical argument for the assumption of monoidal structure is simple: if one has a pair of systems, then one should be able to put them together and consider the composite as a new system.

While this assumption may be ideal in abstract process theories, say where one wishes to model arbitrary qubits as in the ZX-calculus [9], when we turn our attention to *decompositional* approaches to modelling physical systems [10], it becomes apparent that the universe does not behave in a fully monoidal fashion. Rather than starting with a collection of existent systems and presupposing that it is possible to join them together arbitrarily, we start with a global system - the whole of spacetime - and carve out systems with the hope of recovering some fragment of compositional structure.

In such a framework, the tensor becomes problematic, for instance, if we pick a particular system, say a specified qubit A, it is clearly not possible to form the product $A \otimes A$ in the usual sense, for what would it mean to consider the composite of a system with itself? Indeed, the fundamental issue here is trying to tensor two objects that are not independent and that can influence each other in non-trivial ways; we would also have issues taking the tensor of timelike separated systems, or of mixed systems whose environments are not causally disjoint.

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 284–306, doi:10.4204/EPTCS.380.17

There are two main obstructions to hoping for a total tensor product on a category modelling spacetime regions. Firstly, one would like the objects of the category to have a physical interpretation as systems existing in reality. It can often be the case though that no such physical system exists for the composite of physically reasonable systems. For instance, if we take the objects of our category to represent slices of spacetime - closed spacelike subsets of a Lorentzian manifold - when we try to join two slices together they will not form another slice unless the original slices were causally separated.

Secondly, functoriality can fail and one often finds that the interchange law does not hold:

$$(g \otimes 1)(1 \otimes f) \neq (1 \otimes f)(g \otimes 1) \tag{1}$$

while functoriality in each side of the tensor still holds $(1 \otimes f')(1 \otimes f) = (1 \otimes f'f)$ and $(g' \otimes 1)(g \otimes 1) = (g'g \otimes 1)$. This occurs because the systems involved in the tensor may not be independent - they might causally influence each other or possess a shared environment. Thus the casual ordering of f and g is vitally important.

One possible route forwards could be to define the tensor only partially. It was noted in [10] that one can recover a partial monoidal structure where the tensor product is only defined on regions of spacetime that are causally separated. A group theoretic approach was taken in [19] where the resulting category has partial monoidal structure defined only on compatible systems, which requires both the causal separation of systems and also their coupled environments. Another approach starting with a poset modelling the causal relationships of spacetime events [20], resulted in partial monoidality, again only defined on causally separated systems. Partial monoidality, due to similar causality obstructions has appeared in a proposal for modelling the Wolfram model [21].

Outside of applications to physics, partially monoidal categories have made an appearance in [4, 5, 6] where it was noted that the category of finite subsets of some given set N has a partially monoidal structure given by the union of disjoint sets. The authors develop a string diagrammatic language dubbed *nominal string diagrams*, where wires are labelled with elements from the fixed set N. There are similarities between this and the present work - our decompositional approach to physics also has a fixed global set from which we label all systems (a manifold $\mathscr{M}$) and the partial monoidal structure of spacetime slices developed here is also given by unions and intersections of sets. On the other hand, there is a major point of difference between our approach and that of Balco et. al.. While they made the partial monoidal structure total by working with categories internal to a monoidal category, we aim to totalise the partial monoidal structure by working with the presheaves of our category.

We propose the usage of weakenings of monoidal categories in the form of *pro*monoidal [13] and *pre*monoidal [29] categories to model causal curves in spacetime. Premonoidal categories are like monoidal ones but dropping the interchange law (1). They were developed for modelling computational semantics with side-effects and have been used previously to model spacetime particularly in relation to Algebraic Quantum Field Theories [11, 8], where it was argued one could use them to model the Einstein causality condition. Here, we reinforce their point and argue that the lack of bifunctoriality seems to be fundamental in a decompositional approach to spacetime. Premonoidal categories have also appeared elsewhere in applications to petri nets [3].

Promonoidal categories are loosely like monoidal categories into the presheaf category. To our knowledge they have not been directly used in a model of spacetime before. Here, we use them to extend the partial monoidality of spacetime to a total tensor by allowing us to assign useful mathematical objects to otherwise physically problematic ones. For instance, the union of two slices of spacetime is another region of the manifold but not necessarily a slice, thus lacking physical interpretation. We can assign the union a presheaf, with these presheaves being representable whenever the union is another slice. The non-representable presheaves can be thought to act like "virtual systems," they carry useful information but are not physically meaningful.

In sections 2.1 and 2.2 we recap promonoidal and premonoidal categories respectively. In section 3 we introduce toy categories Slice and Space of causal curves in spacetime before showing in section 4 that Slice is a promonoidal category under the operation of taking intersections of sets of causal curves. In section 5 we discuss the operation of taking unions of sets of causal curves and demonstrate that this gives a premonoidal structure on Space while Slice combines the structures of promonoidal and premonoidal categories. Under either of the tensor-like structures on Slice we prove that the presheaves assigned to the tensors are representable if and only if the slices are jointly spacelike and in doing so show that we recover a type of partial tensor product on causally separated regions. In the final section 6 we give the beginnings of a physical interpretation to the operations on Slice as capturing a kind of logical conjunction and disjunction of predicates about particles in spacetime.

2 Preliminaries

2.1 Promonoidal Categories

Before we introduce the formal definition of a promonoidal category let us comment on the intuition we hope to capture.

In a monoidal category $\mathscr{C}$, the tensor product of two objects of $\mathscr{C}$ returns another object in $\mathscr{C}$, that is, it is a functor $\mathscr{C} \times \mathscr{C} \to \mathscr{C}$. Returning to the example of a category of spacetime slices, it is problematic to assign an object of $\mathscr{C}$ to the tensor product whenever the regions of spacetime are timelike separated. The best we could hope for would be a *partial* monoidal structure which is only defined when regions are spacelike separated. Perhaps it might be possible though to assign the tensor of timelike separated regions to be a different sort of object, one that lives outside the category $\mathscr{C}$? What is a sensible choice of such "external" objects and how can we ensure that they work together compatibly such that we might describe the overall structure as something like a tensor product?

We will investigate the usage of promonoidal categories to deal with the aforementioned issues. Rather than assign an object of $\mathscr{C}$ to the tensor product, we assign it a *presheaf*: a functor $\mathscr{C}^{\mathrm{op}} \to \mathsf{Set}$. Presheaves are nicely-behaved mathematical objects: they form a category $[\mathscr{C}^{\mathrm{op}}, \mathsf{Set}]$ where the morphisms are natural transformations between the presheaves, and the Yoneda lemma provides a way of embedding of $\mathscr{C}$ fully and faithfully into its presheaves $\mathbb{Y} : \mathscr{C} \to [\mathscr{C}^{\mathrm{op}}, \mathsf{Set}]$. The image of this functor consists of the *representable* presheaves which are of the form $\mathbb{Y}_A \cong \mathscr{C}(-,A)$ for some object A of $\mathscr{C}$.

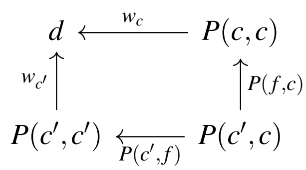
By working with promonoidal categories we are able to assign the tensor a presheaf $(A \otimes B)(-) : \mathscr{C}^{\mathrm{op}} \to \mathsf{Set}$, and in doing so, work with otherwise undefinable tensor products. Since $\mathscr{C}$ embeds into its presheaves, we do not lose any ability to still assign some tensor products to essentially be objects of $\mathscr{C}$. Indeed, when the tensor product of objects of $\mathscr{C}$ is a representable presheaf, $(A \otimes B)(-) \cong \mathscr{C}(-,C)$ we can identify $A \otimes B$ with C under the Yoneda embedding. In this way, promonoidal categories are like partially monoidal ones - when the presheaf is representable we essentially have an object of $\mathscr{C}$ again - but rather than the tensor being undefined elsewhere we can still assign otherwise "untensorable" objects a non-representable presheaf. For a more detailed discussion of the connections between partially monoidal and promonoidal categories see appendix A.

Now, let us start with some core definitions concerning profunctors and their composition. A more comprehensive study can be found in e.g. [26].

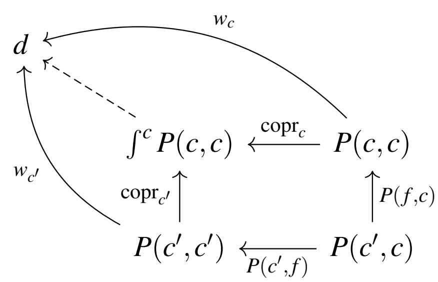
Definition 1 (Profunctor)**.** A profunctor $P : \mathscr{C} \nrightarrow \mathscr{D}$ is a functor $\mathscr{D}^{\mathrm{op}} \times \mathscr{C} \to \mathsf{Set}$.

Profunctors generalise functors in a similar way to how relations generalise functions between sets - profunctors are like “relations between categories,” (note that a relation $A \sim B$ is equivalently a function out of the cartesian product of the sets $A \times B \to \{0,1\}$). We will often use a shorthand Einstein-style notation for profunctors writing $P(d,c) = P^d_c$, with subscripts for covariant variables and superscripts for contravariant ones.

Definition 2 (Cowedge, Coend)**.** Given a profunctor $P : \mathscr{C}^{\text{op}} \times \mathscr{C} \to \mathsf{Set}$, a cowedge (d,w) for P is an object d of Set together with arrows $w_c : P(c,c) \to d$ making the following diagram commute for all f:



The coend of P is a universal cowedge $(\int^c P(c,c), \text{copr})$: this is the cowedge such that all other cowedges factorise uniquely through it:



Coends have a series of nice properties which help to justify the use of an integral symbol to represent them. Firstly, they satisfy a Fubini-style law allowing us to commute coends:

$$\int^c \int^d P(c,c,d,d) \cong \int^{(c,d) \in \mathscr{C} \times \mathscr{D}} P(c,c,d,d) \cong \int^d \int^c P(c,c,d,d)$$

Secondly, the Yoneda lemma implies the following identities, sometimes known as the ninja Yoneda lemma:

$$\int^c \mathscr{C}(-,c) \times F(c) \cong F(-) \qquad\qquad \int^c G(c) \times \mathscr{C}(c,-) \cong G(-) \tag{2}$$

for any functors $F : \mathscr{C}^{\text{op}} \to \mathsf{Set}$ and $G : \mathscr{C} \to \mathsf{Set}$. So the hom-profunctor behaves “like a Dirac-delta function”.

Definition 3 (Composition of Profunctors)**.** Given profunctors $P : \mathscr{C} \nrightarrow \mathscr{D}$ and $Q : \mathscr{D} \nrightarrow \mathscr{E}$, their composite is given by taking the coend

$$(Q \circ P)(-,=) = \int^d Q(-,d) \times P(d,=) : \mathscr{C} \nrightarrow \mathscr{E}$$

This coend can be characterised as the coequaliser:

$$\coprod_{f:d \to d'} Q(-,d) \times P(d',=) \rightrightarrows \coprod_d Q(-,d) \times P(d,=) \longrightarrow \int^d Q(-,d) \times P(d,=)$$

where the coequalised pair "act by f on the left and right under the profunctor". We can think of the resulting quotient set $(Q \circ P)(e,c)$ as equivalence classes of pairs (q,p) where $q \in Q(e,d)$ and $p \in P(d,c)$ under the relations $(Q(e,f)(q),p) \sim (q,P(f,c)(p))$. We will refer to these as the "sliding" relations since it is as though we can slide f from one side to the other (up to changing Q and P).

The composition of profunctors will be written as $(Q \circ P)(e,c) = Q^e_d P^d_c$ in the Einstein notation, where instead of the summation convention we have a "coend convention" - repeated indices, once covariant and once contravariant, are to be coend-ed out. In this way, one also sees the similarity between profunctor composition and matrix multiplication.

Categories, profunctors and natural transformations form a monoidal bicategory Prof where the monoidal product acts as $\mathscr{C} \times \mathscr{D}$ on 0-cells and as $(P \times Q)(c,d,e,f) = P(c,d) \times Q(e,f)$ on 1-cells. The hom-profunctors play a special role in Prof: they are the identity 1-cells by the ninja Yoneda lemma (2).

We are now in a position to define promonoidal categories.

Definition 4 (Promonoidal Category [13, 12])**.** A category $\mathscr{C}$ is promonoidal if it is equipped with

- a tensor product profunctor $\otimes : \mathscr{C} \times \mathscr{C} \nrightarrow \mathscr{C}$
- a unit profunctor $I : 1 \nrightarrow \mathscr{C}$, i.e. a presheaf $I : \mathscr{C}^{\mathrm{op}} \to \mathsf{Set}$

together with natural isomorphisms $\otimes(\otimes \times 1) \overset{\alpha}{\cong} \otimes(1 \times \otimes)$ and $\otimes(\otimes \times I) \overset{\rho}{\cong} 1 \overset{\lambda}{\cong} \otimes(I \times \otimes)$ subject to the triangle and pentagon coherence conditions similar to a monoidal category. A promonoidal category is *strict* when the coherence isomorphisms are identities. A promonoidal category is *symmetric* when there is a natural isomorphism $\sigma_{ABC} : \otimes^A_{BC} \to \otimes^A_{CB}$ satisfying the hexagon equation.

Remark. A very concise definition of a promonoidal category $\mathscr{C}$ is as a pseudomonoid in Prof.

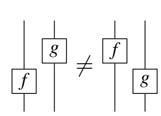
There are many similarities between the definitions of promonoidal and monoidal categories. One can think of promonoidal categories as what we get when we "upgrade" the functors of a monoidal category to profunctors. This really is an upgrade since every functor induces two profunctors by taking its covariant or contravariant Yoneda embeddings. Furthermore, by the following result we can consider promonoidal categories as strictly more general than monoidal ones.

Theorem 1 ([13, 12])**.** *All monoidal categories $(\mathscr{C}, \boxtimes, J)$ are promonoidal categories where we define the tensor profunctor as $(A \otimes B)(-) := \mathscr{C}(-, A \boxtimes B)$ and the unit profunctor as $I(-) := \mathscr{C}(-,J)$. Conversely, a promonoidal category whose tensor and unit are everywhere representable is a monoidal category.*

We will mostly think of the tensor product profunctor $\otimes : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \times \mathscr{C} \to \mathsf{Set}$ in its curried form as a functor into presheaves, $\otimes : \mathscr{C} \times \mathscr{C} \to [\mathscr{C}^{\mathrm{op}}, \mathsf{Set}]$ and in an abuse of notation we freely switch between using $\otimes$ for the tensor product in its three different forms (as a profunctor, a functor into Set and a functor into presheaves) so long as it is clear which we mean.

2.2 Premonoidal Categories

Alongside promonoidal categories, the other monoidal-like structures in this article are premonoidal categories. Premonoidal categories are a weakening of monoidal categories to allow for situations when one can join objects together but each half of the tensor is only individually functorial, that is, while it is the case that $(g' \otimes 1)(g \otimes 1) = (g'g \otimes 1)$ and $(1 \otimes f')(1 \otimes f) = (1 \otimes f'f)$ we have the following inequality:



These categories were originally introduced to model computational semantics with side-effects [29] but we expect categories of causal curves to have similar structure. If f and g act on slices which are timelike separated or have a non-trivial intersection, then their causal ordering can be vitally important; f could change the state space in ways that later influence g or vice-versa. These "hidden" influences between maps can be seen to be somewhat akin to the side-effects in the computational semantics for which premonoidal categories were originally intended.

A premonoidal category has for each object X, a pair of functors $X \rtimes -$ and $- \ltimes X$ acting as the left and right parts of the tensor product, together with compatibility between their actions on objects. More precisely:

- for each pair of objects X and Y of $\mathscr{C}$ there is an assigned object $X \boxtimes Y$ of $\mathscr{C}$,
- for each object X of $\mathscr{C}$, there is a functor $X \rtimes - : \mathscr{C} \to \mathscr{C}$ acting on objects as $X \rtimes Y = X \boxtimes Y$,
- for each object Y of $\mathscr{C}$, there is a functor $- \ltimes Y : \mathscr{C} \to \mathscr{C}$ acting on objects as $X \ltimes Y = X \boxtimes Y$.

There is no compatibility condition between the left and right parts on morphisms, so in general it will be the case that $(f \ltimes Y')(X \rtimes g) \neq (Y \rtimes g)(f \ltimes X')$ for $f : X \to Y$, $g : X' \to Y'$. Pairs of morphism for which such equalities hold, we can think of as acting like the normal tensor and can safely denote $f \otimes g$. In particular, there is a special name for those morphisms which commute with all others:

Definition 5 (Central Morphism [29])**.** A morphism $f : X \to Y$ is central if and only if for all $g : X' \to Y'$, the following two diagrams commute:

$$\begin{array}{ccc} X \otimes X' & \xrightarrow{X \rtimes g} & X \otimes Y' \\ {\scriptstyle f \ltimes X'}\downarrow & & \downarrow{\scriptstyle f \ltimes Y'} \\ Y \otimes X' & \xrightarrow[Y \rtimes g]{} & Y \otimes Y' \end{array} \qquad \begin{array}{ccc} X' \otimes X & \xrightarrow{X' \rtimes f} & X' \otimes Y \\ {\scriptstyle g \ltimes X}\downarrow & & \downarrow{\scriptstyle g \ltimes Y} \\ Y' \otimes X & \xrightarrow[Y' \rtimes f]{} & Y' \otimes Y \end{array}$$

In addition to the above data, a premonoidal category needs associativity and unit natural isomorphisms which are central:

Definition 6 (Premonoidal Category [29])**.** A category $\mathscr{C}$ is premonoidal if it is equipped with left and right tensor functors $X \rtimes -$ and $- \ltimes Y$ for each X and Y, such that they are compatible on objects, together with:

- a unit object I with central isomorphisms $\lambda_X : X \otimes I \to X$ and $\rho_X : I \otimes X \to X$ for each X,
- a central isomorphism $\alpha_{XYZ} : (X \otimes Y) \otimes Z \to X \otimes (Y \otimes Z)$ for each triple X, Y and Z,

such that the triangle and pentagon equations hold and so that the naturality squares for α, λ and ρ hold. A premonoidal category is *strict* when the coherence isomorphisms are identities.

It is possible to combine the left and right tensor functors $X \rtimes -$ and $- \ltimes Y$ into a single functor $\mathscr{C} \,\square\, \mathscr{C} \to \mathscr{C}$ from the *funny tensor product* [17]. A concise definition of the funny tensor is as follows,

Definition 7 (Funny tensor product [31])**.** The funny tensor product $\mathscr{C} \,\square\, \mathscr{D}$ is given by the following pushout

$$\begin{array}{ccc} \mathscr{C}_0 \times \mathscr{D}_0 & \xhookrightarrow{1 \times i_{\mathscr{D}}} & \mathscr{C}_0 \times \mathscr{D} \\ {\scriptstyle i_{\mathscr{C}} \times 1}\big\downarrow & \ulcorner & \big\downarrow \\ \mathscr{C} \times \mathscr{D}_0 & \longrightarrow & \mathscr{C} \,\square\, \mathscr{D} \end{array} \tag{3}$$

where $\mathscr{C}_0$ and $\mathscr{D}_0$ are the discrete categories of the objects of $\mathscr{C}$ and $\mathscr{D}$ respectively.

Explicitly, the category $\mathscr{C} \square \mathscr{D}$ has as objects pairs (c,d) of an object c of $\mathscr{C}$ and d of $\mathscr{D}$. The morphisms are generated by freely composing $(f;1) : (c,d) \to (c',d)$ where $f : c \to c'$ in $\mathscr{C}$ and $(1;g) : (c,d) \to (c,d')$ where $g : d \to d'$ in $\mathscr{D}$ with the rule that compositions exclusively in $\mathscr{C}$ or $\mathscr{D}$ may be contracted: $(f';1)(f;1) = (f'f;1)$ and $(1;g')(1;g) = (1;g'g)$ but $(f;1)(1;g) \neq (1;g)(f;1)$ and thus there is no sensible notion of "$(f;g)$". There is a oplax monoidal functor $\mathscr{C} \square \mathscr{D} \to \mathscr{C} \times \mathscr{D}$ induced by the universal property of the pushout, which forces the interchange squares to commute.

3 A Category of Spacetime Slices

The aim of the remainder of this article is to develop a toy category of spacetime slices and causal curves and then demonstrate that it exhibits both premonoidal and promonoidal structures.

3.1 Spacetimes and Causal Curves

From now on we fix a connected Lorentzian manifold $\mathscr{M}$ with metric g. A tangent vector X is said to be *spacelike*, *timelike* or *null* if $g(X,X) > 0, g(X,X) < 0$ or $g(X,X) = 0$, respectively. $\mathscr{M}$ is said to be *time-orientable* if it has a non-vanishing timelike vector field and the timelike tangent vectors at each point can be divided (in a continuous fashion) into two classes: a *future-directed* and a *past-directed* class. We assume that $\mathscr{M}$ is time-orientable and fix a time-orientation. The assumptions we make of our spacetime are fairly weak causality-wise, and are weaker than those of past- and future-distinguishability [28, 24] (which was assumed by [20]) and certainly weaker than the existence of a Cauchy slice (equivalently global hyperbolicity) [18]. As a result we have not ruled out the existence of closed timelike curves in the spacetime.

A simple example of the kinds of manifolds we are interested in is Minkowski space $\mathbb{R}^{n+1}$ equipped the metric $g(X,X) = |x|^2 - t^2$ for $X = (t,x)$. The timelike vectors are those (t,x) where $t^2 > |x|^2$, of which there are two classes $t > |x|$ and $t < -|x|$ consisting of vectors which point forwards and backwards in time, respectively; a timelike vector (t,x) is future-directed when $t > 0$ and past-directed when $t < 0$. There is no issue with restricting oneself to Minkowski space for the remainder of the article, but we note that the results hold in the fully general case.

A *path* in $\mathscr{M}$ is a continuous map $\mu : \iota \to \mathscr{M}$ where $\iota \subseteq \mathbb{R}$ is a (possibly unbounded) real interval. Such a path is *smooth* if it is infinitely differentiable and *regular* if its first derivative is non-vanishing. A smooth regular path is *causal* when the tangent vector is timelike or null at all points in the path and a causal path is *future-directed* when the tangent at every point is future-directed. For a point $x \in \mathscr{M}$, the set of all points $y \in \mathscr{M}$ with a future-directed path x to y is called the *future light cone* of x, whereas the set of all points with a future-directed path from y to x is called the *past light cone* of x.

Often it is more convenient to work with equivalence classes of paths, up to reparametrisation, i.e. $\mu \sim \mu'$ if and only if there exists a monotone map $r : \iota \to \iota'$ such that $\mu' \circ r = \mu$. An equivalence class of causal paths is called a *causal curve*. Since being future-directed is preserved by $\sim$, we can also say a causal curve is future-directed without ambiguity.

A point $x \in \mathscr{M}$ *causally precedes* another point $y \in \mathscr{M}$, written $x \prec y$, if there exists a future-directed causal curve from x to y, or if $x = y$. The assumption of time-orientability of $\mathscr{M}$ is not enough to ensure that $\prec$ gives a total order on points in a causal curve - for instance there could be closed timelike curves in $\mathscr{M}$ containing points $x \neq y$, for which $x \prec y$ and $y \prec x$.

A *region* is any arbitrary subset $A \subseteq \mathscr{M}$ of the manifold. Regions are too general to be useful for many practical applications, they might contain points which causally precede each other or they might have insufficient topological properties to make them well-behaved. As a result we will be more interested in a restricted class of regions, the *spacelike* regions, where for all $x, y \in \Sigma$, $x \neq y$, x does not causally precede y and thus there are no future-directed causal curves connecting x with y, or y with x. For instance, in Minkowski space the surfaces given by fixed times $t = \tau$ are examples of spacelike sets.

Definition 8 (Spacelike Slice)**.** A spacelike slice (or simply a "slice") is a closed spacelike set.

It is worth noting that slices may still be too weak for many applications, and it may be necessary to demand further properties of them, by working with the Cauchy slices for instance. Whilst we do not make these restrictions in this work, in principle, there is no obstacle to applying many of the same methods to categories of more restrictive classes of slices.

We will be very interested in the causal relationship between slices X and Y, which motivates the following definition.

Definition 9 (Jointly Spacelike Slices)**.** Slices X and Y are jointly spacelike if their union $X \cup Y$ is spacelike.

Given regions $A, B \subseteq \mathscr{M}$, $A \neq B$, we say that a future-directed causal curve γ with representative path $\mu : \iota \to \mathscr{M}$, *passes through A and then B* if there exists a $q \in \iota$ with $\mu(q) \in B$ and for all such q there exists $p \leq q \in \iota$ such that $\mu(p) \in A$. We write $\mathscr{C}[A,B]$ for the set of future-directed causal curves passing through A and then B. We write $\mathscr{C}[A] := \mathscr{C}[A,A]$ for the set of future-directed causal curves which pass through A (with no constraint on other regions through which they must pass). It is worth noting that a closed timelike curve γ containing both the points $a \in A$ and $b \in B$ will be in the sets $\mathscr{C}[A,B]$ and $\mathscr{C}[B,A]$.

3.2 A Category of Causal Curves

With these definitions in place we can define the following categories of slices and regions of spacetime:

Definition 10 ($\mathsf{Slice}, \mathsf{Space}$)**.** The category Slice has as objects slices $X \subset \mathscr{M}$ (closed spacelike sets). For two slices $X, Y \subset \mathscr{M}$, the homset $\mathsf{Slice}(X,Y) := \mathscr{P}(\mathscr{C}[X,Y])$ is the powerset of $\mathscr{C}[X,Y]$, that is, a morphism $X \to Y$ is a set of future-directed causal curves through X then Y. Given two subsets $S : X \to Y$ and $T : Y \to Z$, their composition is given by intersection: $T \circ S := T \cap S \subset \mathscr{C}[X,Z]$. The identity morphism $1_X : X \to X$ is given by the set $\mathscr{C}[X,X]$ of all curves through X.

The category Space has as objects arbitrary regions $A \subseteq \mathscr{M}$. All other data is as Slice.

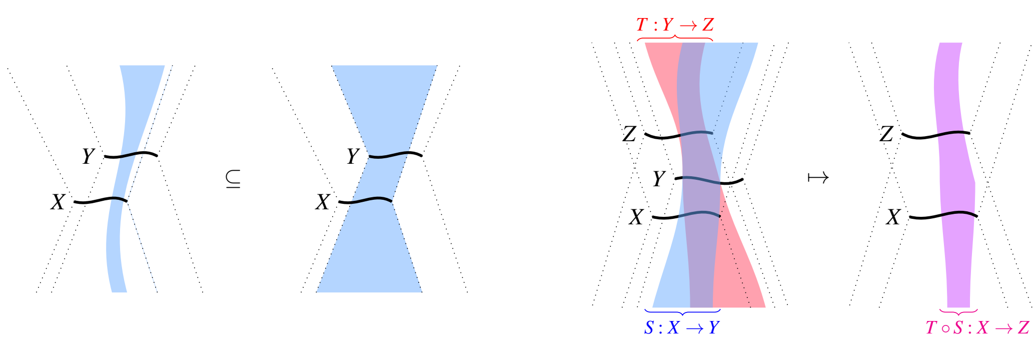


Figure 1: *Left:* A morphism in the category Slice is a set of causal curves passing first through X then through Y. *Right:* Composition of two morphisms in Slice via intersection. Note that in both pictures, past as future light cones of slices are depicted as dotted lines, and sets of many causal curves are depicted as filled-in regions.

Proposition 1. Slice *and* Space *are categories.*

Proof. Composition is associative because intersection is. Given a set of causal curves $S : X \to Y$, by definition all curves in S pass through X, thus we see $S \circ 1_X = S \cap \mathscr{C}[X,X] = S$. Similarly for the left composition with identity morphisms. □

Now we examine a few basic categorical properties of Slice and Space.

Proposition 2. Slice *and* Space *have equalisers and coequalisers, given by the complement of the symmetric difference.*

Proof. Take $f,g : A \to B$. This pair of parallel arrows is equalised by $(f \triangle g)^c : A \to A$ and coequalised by $(f \triangle g)^c : B \to B$ where $(f \triangle g)^c = \mathscr{C}[A,B] \setminus (f \triangle g) = (f \cup g)^c \cup (f \cap g)$. Any other arrow h making the parallel pair f and g equal factorises uniquely via $(f \triangle g)^c$ because this morphism contains every causal curve that is in both f and g, or neither. Thus h must be a subset of $(f \triangle g)^c$. □

It is interesting that equalisers and coequalisers essentially coincide in Slice - in part this is down to the fact that composition is, up to types, commutative - e.g. for endomorphisms $f \circ g = g \circ f$.

Proposition 3. *Let X and Y be jointly spacelike slices with $X \cap Y = \varnothing$. Then the product and coproduct of X and Y exist in* Slice *and are given by the set theoretic union $X \times Y = X \oplus Y = X \cup Y$.*

Proof. Proof given in appendix B.1. □

While we do have products and coproducts of non-intersecting jointly spacelike slices in Slice, the (co)products of other regions e.g. timelike separated regions and of intersecting slices do not exist. These regions are the main issue preventing the set theoretic union from being a monoidal structure on Slice.

Proposition 4. Slice *is **not** a monoidal category under a monoidal product given by taking the union of regions and curves $X \otimes Y := X \cup Y$ and $S \otimes T := S \cup T$.*

Proof. The union of slices is not always a slice so $X \cup Y$ may not be an object of Slice. For the occasions when it is, $\otimes$ cannot in general be bifunctorial. For arbitrary $S : X \to Y$, $S' : Y \to Z$, $T : X' \to Y'$ and $T' : Y' \to Z'$, we have $(S' \otimes T') \circ (S \otimes T) = (S' \cup T') \cap (S \cup T) \supset (S' \cap S) \cup (T' \cap T) = (S' \circ S) \otimes (T' \circ T)$. □

One might hope that by relaxing the sorts of objects we are considering and working instead with the category Space, we could find a monoidal product given by union. Whilst this resolves the issue of the non-existence of the object $X \cup Y$ for arbitrary X and Y, we still find that the union cannot be bifunctorial and thus Space is also not a monoidal category under union.

We also cannot hope that Slice or Space are monoidal categories under intersection because there exist causally connected slices which have an empty intersection:

Proposition 5. Slice *and* Space *are **not** monoidal categories under a monoidal product given by taking the intersection of regions and curves $X \otimes Y := X \cap Y$ and $S \otimes T := S \cap T$.*

Proof. Suppose X and Y are causally connected slices so $\mathscr{C}[X,Y] \neq \varnothing$ but with $X \cap Y = \varnothing$. Then $1_X \otimes 1_Y = \mathscr{C}[X,X] \cap \mathscr{C}[Y,Y] \neq \varnothing$ because there exists a causal curve passing through X and Y. On the other hand we see that $1_{X \cap Y} = 1_\varnothing = \varnothing$. □

In the following sections we will show that while Slice and Space are not monoidal categories in either of these ways, Slice is a promonoidal category under intersection. Under union, Space is premonoidal while Slice combines both promonoidal and premonoidal structures.

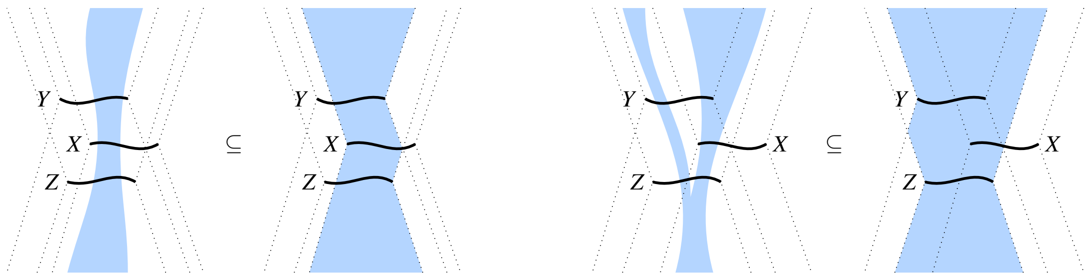


Figure 2: *Left:* An element $S \in (X \owedge Y)(Z)$, as defined in Section 4. *Right:* An element $T \in (X \ovee Y)(Z)$, as defined in Section 5.

4 A Promonoidal Structure on Slice

We now aim to show that Slice is a promonoidal category under intersection, that is, it is equipped with a tensor product functor $\mathsf{Slice} \times \mathsf{Slice} \to [\mathsf{Slice}^{\mathrm{op}}, \mathsf{Set}]$ and unit presheaf $\mathsf{Slice}^{\mathrm{op}} \to \mathsf{Set}$ subject to associativity and unit laws.

To each pair of objects X and Y we assign the presheaf $(X \owedge Y)(-) : \mathsf{Slice}^{\mathrm{op}} \to \mathsf{Set}$ which sends a slice Z to the powerset of causal curves which pass through Z and then both X and Y

$$(X \owedge Y)(Z) := \mathscr{P}(\mathscr{C}[Z,X] \cap \mathscr{C}[Z,Y])$$

On morphisms $S : Z' \to Z$ this presheaf acts by intersection:

$$(X \owedge Y)(S) : (X \owedge Y)(Z) \to (X \owedge Y)(Z') :: C \mapsto C \cap S$$

Lemma 1. $(X \owedge Y)(-)$ *is a presheaf.*

Proof. $(X \owedge Y)(1_Z) :: C \mapsto C \cap 1_Z = C$ because every curve in $(X \owedge Y)(Z)$ passes through Z. Thus $(X \owedge Y)(1_Z) = 1_{(X \owedge Y)(Z)}$. Now $(X \owedge Y)(T) \circ (X \owedge Y)(S) :: C \mapsto C \cap S \mapsto (C \cap S) \cap T$ while $(X \owedge Y)(S \circ T) :: C \mapsto C \cap (S \cap T)$ and these are equal by the associativity of intersection. □

To each $(S,T) : (X,Y) \to (X',Y')$ we are required to assign a natural transformation between the presheaves $S \owedge T : (X \owedge Y)(-) \implies (X' \owedge Y')(-)$.

For $S : X \to X'$ there is a natural transformation with components

$$(S \owedge Y)_Z : (X \owedge Y)(Z) \to (X' \owedge Y)(Z) :: C \mapsto C \cap S$$

and for $T : Y \to Y'$ there is a natural transformation with components

$$(X \owedge T)_Z : (X \owedge Y)(Z) \to (X \owedge Y')(Z) :: C \mapsto C \cap T$$

These natural transformations commute, $(S \owedge Y')_Z (X \owedge T)_Z = (X' \owedge T)_Z (S \owedge Y)_Z$ and we can define $(S \owedge T)$ to be given by their composition.

Lemma 2. $(S \owedge Y)$ *and* $(X \owedge T)$ *are natural transformations with* $(S \owedge Y')_Z (X \owedge T)_Z = (X' \owedge T)_Z (S \owedge Y)_Z$.

Proof. Proof given in appendix B.2. □

Lemma 3. *The assignment* $(X,Y) \mapsto (X \owedge Y)(-)$ *and* $(S,T) \mapsto (S \owedge T)$ *gives a functor* $\mathsf{Slice} \times \mathsf{Slice} \to [\mathsf{Slice}^{op}, \mathsf{Set}]$.

Proof. Proof given in appendix B.3. □

We are now in a position to prove the main result of this section:

Theorem 2. Slice *is a symmetric promonoidal category where the tensor is given above and the unit presheaf is given by* $I(Z) := \mathscr{P}(\mathscr{C}[Z,Z])$.

Proof. Proof given in appendix B.4. □

Now we know that Slice is promonoidal under intersection, we will study when the presheaves assigned by this tensor are representable. This allows us to ascertain where $\otimes$ acts like a standard monoidal product on Slice and where it is possible for us to consider the tensor of slices to be another slice.

Theorem 3. *When X and Y are jointly spacelike slices, the presheaf $(X \otimes Y)(-)$ is representable.*

Proof. Suppose X and Y are jointly spacelike. Note $\mathscr{C}[Z,X] \cap \mathscr{C}[Z,Y] \supseteq \mathscr{C}[Z,X \cap Y]$. Suppose there exists $\gamma \in \mathscr{C}[Z,X] \cap \mathscr{C}[Z,Y]$ with $\gamma \notin \mathscr{C}[Z,X \cap Y]$. Then γ must pass through some $x \in X \backslash Y$ and some $y \in Y \backslash X$ but this would imply that X and Y are not jointly spacelike. Thus γ cannot exist and it follows that $(X \otimes Y)(Z) = \mathscr{P}(\mathscr{C}[Z,X \cap Y]) = \mathsf{Slice}(Z,X \cap Y) = \mathbb{Y}_{X \cap Y}(Z)$, noting that $X \cap Y$ is a slice because $X \cap Y \subseteq X$ and thus is an object of Slice. □

In particular, the previous theorem shows that on jointly spacelike slices $\otimes$ acts like intersection and we can make the identification $(X \otimes Y)(-) \simeq X \cap Y$. On the other hand, when the slices are not jointly spacelike there is no representative for $(X \otimes Y)(-)$. To show this we need the following lemma:

Lemma 4. *Let $A \subseteq \mathscr{M}$ be a closed subset of $\mathscr{M}$. Then for any $x \in \mathscr{M}$, $x \notin A$, there exists a causal curve through x which does not intersect A.*

Proof. The timelike vector field is non-vanishing on $\mathscr{M}$ and as a result there must be a causal curve γ through x. In a sufficiently small neighbourhood U of x, γ must restrict to a causal curve which is contained entirely within U. Since A is closed and $\mathscr{M}$ is Hausdorff, this neighbourhood can be made sufficiently small such that $U \cap A = \varnothing$. □

Theorem 4. *When X and Y are not jointly spacelike, the presheaf $(X \otimes Y)(-)$ is not representable.*

Proof. We make much use of Lemma 4. Suppose X and Y are not jointly spacelike and suppose for a contradiction that $(X \otimes Y)(-) = \mathsf{Slice}(-,Z)$ for some slice Z.

Now suppose there exists a $z \in Z$ such that $z \notin X \cup Y$. We can find a causal curve γ through z that does not also pass through $X \cup Y$. It follows that $\gamma \in \mathsf{Slice}(Z,Z)$, but $\gamma \notin (X \otimes Y)(Z)$. So Z cannot represent the presheaf and we conclude $Z \subseteq X \cup Y$.

Now take a $x \in X \backslash Y$. There exists a causal curve γ passing through x but not Y. Suppose that $x \in Z$, then $\gamma \in \mathsf{Slice}(Z,Z)$, but $\gamma \notin (X \otimes Y)(Z)$. So $x \notin Z$.

A similar argument shows that any $y \in Y \backslash X$ cannot be in Z and thus $Z \subseteq X \cap Y$.

Since X and Y are not jointly spacelike, $X \cup Y$ is not spacelike and there exists a causal curve γ from $X \cup Y$ to itself. In particular γ must pass through a point of X and a point of Y, and not, say, through two points of X, since X and Y are slices. Then we would have $\gamma \in (X \otimes Y)(X)$ but $\gamma \notin \mathsf{Slice}(X,Z)$ because if $\gamma \in \mathsf{Slice}(X,Z)$ it would pass through X and $X \cap Y \subseteq X$, a contradiction with X being a slice. □

So we have shown that $(X \otimes Y)(-)$ is representable if and only if X and Y are jointly spacelike. Note that one **cannot** define a partially monoidal category by just working with $\otimes$ where it is representable because the unit presheaf is not representable (the whole manifold is not a slice) and therefore there is no unit object available in Slice.

5 The Structure of Slice and Space under Union

Let us now consider the structure of Slice and Space under union of slices and sets of curves. The larger category Space where the objects are arbitrary subsets of the manifold $\mathscr{M}$ and the homsets are powersets of causal curves is a premonoidal category:

Proposition 6. Space *is a strict premonoidal category under the operation of taking the union of regions and curves.*

Proof. For objects X and Y assign them the object $X \otimes Y := X \cup Y$. The assignment $(T : Y \to Y') \mapsto (\mathscr{C}[X] \cup T : X \cup Y \to X \cup Y')$ gives a functor $X \rtimes - : \mathscr{C} \to \mathscr{C}$ because

$$X \rtimes 1_Y = \mathscr{C}[X] \cup \mathscr{C}[Y] = \mathscr{C}[X \cup Y] = 1_{X \cup Y}$$

$$(X \rtimes f')(X \rtimes f) = (\mathscr{C}[X] \cup f') \cap (\mathscr{C}[X] \cup f) = \mathscr{C}[X] \cup (f' \cap f) = X \rtimes f'f$$

Similarly the assignment $(S : X \to X') \mapsto (S \cup \mathscr{C}[Y])$ extends to a functor $- \ltimes Y : \mathscr{C} \to \mathscr{C}$. The unit object is $I := \varnothing$ and the unit and associativity isomorphisms are identities, which it is straightforward to check are central. □

The above has a clear issue - $X \cup Y$ is generally not another slice and thus not an object of Slice. This means Slice cannot form a premonoidal category under union and we need to search for something that combines both premonoidal and promonoidal structures together.

There is no obstacle to defining presheaves $(X \ovee Y)(-) : \mathsf{Slice}^{\mathrm{op}} \to \mathsf{Set}$ which send a slice Z to the powerset of causal curves through Z and either X or Y:

$$(X \ovee Y)(Z) := \mathscr{P}(\mathscr{C}[Z, X] \cup \mathscr{C}[Z, Y])$$

On morphisms $S : Z' \to Z$ this presheaf acts by intersection:

$$(X \ovee Y)(S) : (X \ovee Y)(Z) \to (X \ovee Y)(Z') :: C \mapsto C \cap S$$

Lemma 5. $(X \ovee Y)(-)$ *is a presheaf.*

Similarly, there is no obstacle to defining natural transformations acting on either the left or right of $\ovee$. For $S : X \to X'$ there is a natural transformation with components

$$(S \ovee Y)_Z : (X \ovee Y)(Z) \to (X' \ovee Y)(Z) :: C \mapsto C \cap (S \cup \mathscr{C}[Y])$$

and for $T : Y \to Y'$ there is a natural transformation with components

$$(X \ovee T)_Z : (X \ovee Y)(Z) \to (X \ovee Y')(Z) :: C \mapsto C \cap (\mathscr{C}[X] \cup T)$$

Lemma 6. $(S \ovee Y)$ *and* $(X \ovee T)$ *are natural transformations.*

What fails in comparison to $\owedge$ is that, in general, the components of these natural transformations do not obey the interchange law, so we cannot hope that these data give a functor $\mathsf{Slice} \times \mathsf{Slice} \to [\mathsf{Slice}^{\mathrm{op}}, \mathsf{Set}]$. Nevertheless, the natural transformations are functorial on each side of the tensor and it is easy to verify that the assignment does give a functor $\ovee : \mathsf{Slice} \square \mathsf{Slice} \to [\mathsf{Slice}^{\mathrm{op}}, \mathsf{Set}]$ where $\square$ is the funny tensor product of categories.

Lemma 7. *The data of Lemmas 5 and 6 specify a functor* $\mathsf{Slice} \square \mathsf{Slice} \to [\mathsf{Slice}^{op}, \mathsf{Set}]$

Proof. Proof given in appendix B.5. □

In this way Slice seems to combine both the structures of premonoidal and promonoidal categories. We leave it as future work to make rigorous the associativity and unitality of this structure but we note that the representable presheaf at the empty slice $\mathbb{Y}_\varnothing$ is likely the unit of a suitably defined structure.

Similarly to the intersection case we can study when the presheaves $(X \circledcirc Y)(-)$ are representable:

Theorem 5. *When X and Y are jointly spacelike, the presheaf $(X \circledcirc Y)(-)$ is representable.*

Proof. Suppose X and Y are jointly spacelike. Then $(X \circledcirc Y)(Z) = \mathscr{P}(\mathscr{C}[Z,X] \cup \mathscr{C}[Z,Y]) = \mathscr{P}(\mathscr{C}[Z,X \cup Y]) = \mathbb{Y}_{X\cup Y}(Z)$ where we have used the fact that $X \cup Y$ is spacelike and thus an object of Slice. □

Theorem 6. *When X and Y are not jointly spacelike, the presheaf $(X \circledcirc Y)(-)$ is not representable.*

Proof. We make use of Lemma 4. Suppose X and Y are not jointly spacelike and suppose for a contradiction that $(X \circledcirc Y)(-) = \mathsf{Slice}(-,Z)$ for some slice Z. By the same argument made in the proof of Theorem 4 we must have $Z \subseteq X \cup Y$.

Since X and Y are not jointly spacelike, $X \cup Y$ is not spacelike and thus there exists a causal curve γ connecting two points of $X \cup Y$. It must be the case that one of these points is in $X \backslash Y$ and the other in $Y \backslash X$ else X or Y could not be slices. Write $x \in X \backslash Y$ and $y \in Y \backslash X$ for these points that γ passes through and note that they can be the only points of $X \cup Y$ that γ intersects else X or Y could not be slices.

Now note that γ restricts to a causal curve γ_x which passes through x but not y and similarly a causal curve γ_y which passes through y but not x.

Suppose that $x \notin Z$, then $\gamma_x \in (X \circledcirc Y)(X)$ but $\gamma_x \notin \mathsf{Slice}(X,Z)$, noting that $Z \subseteq X \cup Y$ so that γ_x intersects Z at only x. So we conclude that $x \in Z$.

Similarly, suppose that $y \notin Z$, then $\gamma_y \in (X \circledcirc Y)(Y)$ but $\gamma_y \notin \mathsf{Slice}(Y,Z)$. So we conclude that $y \in Z$.

We see that γ is a causal curve connecting two distinct points of Z and consequently Z cannot be a slice. □

So we have shown that the presheaf $(X \circledcirc Y)(-)$ is representable if and only if X and Y are jointly spacelike. By restricting $\circledcirc$ to these slices we can recover a partial premonoidal structure on Slice by defining the tensor to be given by the representative. The unit of this partial premonoidal category is the empty slice $\varnothing$.

Now that we have two tensor-like structures on Slice we would like to know how they interact. Given that $\circledcirc$ behaves like union and $\oslash$ like intersection, it seems reasonable to expect some sort of distributivity between them. To understand this at the level of the profunctors we require the following definition:

Definition 11 (Multiplicative Kernel [14])**.** Let $(\mathscr{C}, P, I)$ and $(\mathscr{D}, Q, J)$ be promonoidal categories. A multiplicative kernel is a profunctor $K : \mathscr{C} \nrightarrow \mathscr{D}$ such that

$$Q(K \times K) \cong KP \qquad\qquad KI \cong J$$

where concatenation is profunctor composition.

Remark. Viewing $\mathscr{C}$ and $\mathscr{D}$ as pseudomonoids in Prof, a multiplicative kernel is a homomorphism of these monoids.

Each slice X determines an endoprofunctor $(X \circledcirc -)(-) : \mathsf{Slice} \nrightarrow \mathsf{Slice}$ and it is the case that each of these is a multiplicative kernel for Slice equipped with $\oslash$.

Theorem 7. *For every slice X, $(X \circledcirc -)(-)$ is a multiplicative kernel for* $(\mathsf{Slice}, \oslash)$.

Proof. Proof given in appendix B.6. □

6 Interpreting the operations in Slice

We have shown that Slice admits two operations $\ovee$ and $\owedge$ taking a pair of spacelike slices to a "generalised" slice, i.e. a presheaf over slices. Here, we give (the beginnings of) a physical interpretation for these operations.

First, it is helpful to shift from thinking geometrically about slices to thinking logically about them. That is, we can think of a slice X as a logical predicate, namely that a system satisfies a certain property at a certain moment in time. The simplest non-trivial example is 1+1 dimensional Minkowski space, where a particle in 1D space traces out a causal curve through $\mathbb{R}^{1+1}$.

As a simple case, we can consider slices of the form $X := \{t_1\} \times P$ for a time $t_1 \in \mathbb{R}$ and a closed subset $P \subseteq \mathbb{R}$. We can now think of P as saying something about the position of a particle at time t_1, e.g. "the particle's position is $\geq x_1$". Similarly, another slice $Y := \{t_2\} \times Q$, expresses that a certain property Q holds for a particle at time t_2, e.g. "the particle's position is $\leq x_2$".

We can now think about whether it makes sense to take conjunctions or disjunctions of these kinds of predicates. If $t_1 = t_2$, then everything works out exactly as one would expect. Namely, $X \owedge Y = \{t_1\} \times (P \cap Q)$, which captures the statement that at time t_1, "the particles position is $\geq x_1$ AND it is $\leq x_2$". Similarly, $X \ovee Y = \{t_1\} \times (P \cup Q)$, capturing the OR if predicates P and Q at a fixed time t_1.

If we look at arbitrary pairs of jointly spacelike slices X and Y, then much the same interpretation holds, but rather than separating the time and space coordinates in a fixed reference frame, we can regard X and Y as living on the same spacelike hypersurface.

The more interesting case is of course when X and Y are not jointly spacelike. While we can't make sense of $X \owedge Y$ and $X \ovee Y$ as spacelike slices themselves, we can make sense of them relative to a third, "probe" slice Z. If we restrict to the simpler case where $X = \{t_1\} \times P$ and $Y = \{t_2\} \times Q$, now with $t_1 \neq t_2$ and possibly some causal curves between X and Y, then any $S \in (X \owedge Y)(Z)$ is a set of causal curves that first passes through Z then must satisfy P at t_1 AND Q at t_2. Hence, $\owedge$ captures conjunction, but with predicates at different times. Similarly, $\ovee$ captures this generalised kind of disjunction.

We can apply this kind of interpretation to arbitrary pairs of slices X, Y, not just those which take a product form in a fixed reference frame, however the meaning is slightly less intuitive in some cases, like when X and Y intersect and are furthermore not jointly spacelike. Nevertheless, we obtain a notion of conjunction and disjunction which is defined everywhere, and thanks to Theorem 7, distributes as one would expect. Hence, we have the beginnings of a logic for (generalised) spacetime slices. However there is much still to explore. For example, there is no clear "universal" notion of negation here, but one may be able to negate a slice relative to another one, e.g. some Cauchy surface containing the slice.

7 Conclusion and Future Work

We have shown that the category Slice of spacelike slices and causal curves admits two generalised tensor-like structures, corresponding to conjunction and disjunction. We see several avenues of future work. One is the complete characterisation of the structure $\ovee$ defined in Section 5, which combines elements of both a premonoidal and promonoidal product. As promonoidal and strict premonoidal categories can be formalised as pseudomonoids in a suitable monoidal bicategory, one might hope to do similar for the "pre-promonoidal" structure $\ovee$.

As hinted at the end of the previous section, there seems to be much more left to say about the logical interpretation of connectives in Slice. For instance, one could try to obtain an analogue to full classical logic by introducing a (suitably localised) negation. It also seems natural to study non-commutative connectives such as the "sequence" product $\triangleleft$ present in the logic BV [22], which was recently shown to capture the *one-way signalling* processes in the Caus$[-]$ construction [30].

Another direction is to investigate other places tensor-like structures appear, particularly within models that have some notion of "causality" which may be different from the usual relativistic one. For example, by imposing restrictions on the Petri nets of [3], one may force the monoidal category FP developed there to be only promonoidal. In such a case it seems that the fibres FP_i are no longer premonoidal but can be described by a pre-promonoidal category.

While the category Slice we defined here gives an interesting toy theory for exploring spacetime, causal curves, and associated notions of logic and compositionality, it is by no means the "one true" category of spacetime. It would be interesting to study variations on this structure, which may have different, possibly more natural notions of composition. For example, instead of intersecting sets of curves, one could define a category Slice' where composition is given by "gluing" curves together, somewhat in the same spirit as [20]. Such a category seems more amenable to an alternative view of AQFT, as functors $\mathsf{Slice}' \to \mathsf{Alg}_k$, or indeed with codomain taken to be any reasonable process theory.

Finally, one could compare our approach to other categorical models of causality and spacetime, such as the formulation using idempotent subobjects [15, 16], the order-theoretic formulation of [20], and the aforementioned Caus$[-]$ construction [23, 30].

Acknowledgements

JH is supported by University College London and the EPSRC [grant number EP/L015242/1]. AK is supported by Grant No. 61466 from the John Templeton Foundation as part of the QISS project. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation. Many thanks to Matt Wilson, Cole Comfort, John van de Wetering, Bob Coecke, and Raymond Lal for useful recent (and not-so-recent!) discussions on categories of spacetime; and to Stefano Gogioso for noticing a few issues with an early draft of this work.

References

[1] Samson Abramsky & Bob Coecke (2004): *A categorical semantics of quantum protocols*. In: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, pp. 415–425, doi:10.1109/LICS.2004.1319636.

[2] John C. Baez & James Dolan (1995): *Higher-dimensional algebra and topological quantum field theory*. *Journal of Mathematical Physics* 36(11), pp. 6073–6105, doi:10.1063/1.531236.

[3] John C. Baez, John Foley & Joe Moeller (2019): *Network Models from Petri Nets with Catalysts*. *Compositionality* 1, doi:10.32408/compositionality-1-4.

[4] Samuel Balco (2019): *Display calculi and nominal string diagrams*. Ph.D. thesis, University of Leicester, doi:10.25392/leicester.data.13315373.v1.

[5] Samuel Balco & Alexander Kurz (2018): *Partially monoidal categories and the algebra of simultaneous substitutions*. Available at `https://gdlyrttnap.pl/resources/papers/syco1.pdf`.

[6] Samuel Balco & Alexander Kurz (2019): *Nominal String Diagrams*. In Markus Roggenbach & Ana Sokolova, editors: *8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 139, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 18:1–18:20, doi:10.4230/LIPIcs.CALCO.2019.18.

[7] Jean Bénabou (2000): *Distributors at Work*. Available at `http://www.entretemps.asso.fr/maths/Distributors.pdf`. Notes for a course on distributors at TU Darmstadt.

[8] Richard Blute & Marc Comeau (2015): *Von Neumann Categories*. Appl Categor Struct 23, p. 725–740, doi:10.1007/s10485-014-9375-6.

[9] Bob Coecke & Ross Duncan (2011): *Interacting quantum observables: categorical algebra and diagrammatics*. New Journal of Physics 13(043016), doi:10.1088/1367-2630/13/4/043016.

[10] Bob Coecke & Raymond Lal (2013): *Causal Categories: Relativistically Interacting Processes*. Found Phys 43, p. 458–501, doi:10.1007/s10701-012-9646-8.

[11] Marc Comeau (2012): *Premonoidal *-Categories and Algebraic Quantum Field Theory*. Ph.D. thesis, University of Ottawa, doi:10.20381/ruor-5524.

[12] Brian Day (1970): *Construction of Biclosed Categories*. Ph.D. thesis, University of New South Wales, doi:10.26190/unsworks/8048.

[13] Brian Day (1970): *On closed categories of functors*. In: Reports of the Midwest Category Seminar IV, 137, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–38, doi:10.1007/BFb0060438.

[14] Brian Day (2011): *Monoidal functor categories and graphic Fourier transforms*. Theory and Applications of Categories 25(5), pp. 118–141.

[15] Pau Enrique Moliner (2021): *Space and time in monoidal categories*. Ph.D. thesis, The University of Edinburgh, doi:10.7488/era/1259.

[16] Pau Enrique Moliner, Chris Heunen & Sean Tull (2018): *Space in Monoidal Categories*. Electronic Proceedings in Theoretical Computer Science 266, pp. 399–410, doi:10.4204/eptcs.266.25.

[17] François Foltz, Christian Lair & Gregory M. Kelly (1980): *Algebraic categories with few monoidal biclosed structures or none*. Journal of Pure and Applied Algebra 17, pp. 171–177, doi:10.1016/0022-4049(80)90082-1.

[18] Robert Geroch (1970): *Domain of Dependence*. Journal of Mathematical Physics 11(2), pp. 437–449, doi:10.1063/1.1665157.

[19] Stefano Gogioso (2019): *A Process-Theoretic Church of the Larger Hilbert Space*, doi:10.48550/arXiv.1905.13117. arXiv:1905.13117.

[20] Stefano Gogioso, Maria E. Stasinou & Bob Coecke (2021): *Functorial Evolution of Quantum Fields*. Frontiers in Physics 9, doi:10.3389/fphy.2021.534265.

[21] Jonathan Gorard, Manojna Namuduri & Xerxes D. Arsiwalla (2021): *Fast Automated Reasoning over String Diagrams using Multiway Causal Structure*, doi:10.48550/arXiv.2105.04057. arXiv:2105.04057.

[22] Alessio Guglielmi (2007): *A system of interaction and structure*. ACM Transactions on Computational Logic, doi:10.1145/1182613.1182614.

[23] Aleks Kissinger & Sander Uijlen (2019): *A categorical semantics for causal structure*. Logical Methods in Computer Science, doi:10.23638/LMCS-15(3:15)2019.

[24] E. H. Kronheimer & Roger Penrose (1967): *On the structure of causal spaces*. Mathematical Proceedings of the Cambridge Philosophical Society 63(2), p. 481–501, doi:10.1017/S030500410004144X.

[25] Raymond Lal (2012): *Causal structure in categorical quantum mechanics*. Ph.D. thesis, Oxford University, UK.

[26] Fosco Loregian (2021): *(Co)end Calculus*. London Mathematical Society Lecture Note Series, Cambridge University Press, doi:10.1017/9781108778657.

[27] Fosco Loregian & Emily Riehl (2020): *Categorical notions of fibration*. Expositiones Mathematicae 38(4), pp. 496–514, doi:10.1016/j.exmath.2019.02.004.

[28] David B. Malament (1977): *The class of continuous timelike curves determines the topology of spacetime*. Journal of Mathematical Physics 18(7), pp. 1399–1404, doi:10.1063/1.523436.

[29] John Power & Edmund Robinson (1997): *Premonoidal categories and notions of computation*. Mathematical Structures in Computer Science 7(5), p. 453–468, doi:10.1017/S0960129597002375.

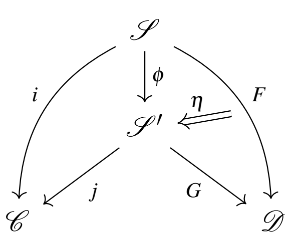
[30] Will Simmons & Aleks Kissinger (2022): *Higher-Order Causal Theories Are Models of BV-Logic*. In: 47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022), Leibniz International Proceedings in Informatics (LIPIcs) 241, pp. 80:1–80:14, doi:10.4230/LIPIcs.MFCS.2022.80.

[31] Mark Weber (2013): *Free Products of Higher Operad Algebras*. Theory and Applications of Categories 28(2).

A Partially Monoidal Categories as Promonoidal Categories

In this appendix we compare the partially monoidal categories of [10, 25, 19] to promonoidal categories. We discuss a class of partially monoidal categories that can be equivalently described as promonoidal categories which are representable wherever the presheaves are non-empty and discuss when it is possible to derive a partially monoidal category from a promonoidal one.

Definition 12 (Partial Functor [19])**.** A partial functor $\mathscr{C} \rightharpoonup \mathscr{D}$ is a span of functors $\mathscr{C} \xleftarrow{i} \mathscr{S} \xrightarrow{F} \mathscr{D}$ where i is an opisofibration, embedding S as a subcategory of $\mathscr{C}$ (so i is full, faithful and $\mathscr{S}$ is a replete subcategory of $\mathscr{C}$). Composition of partial functors is by pullback. A morphism of partial functors $(\phi,\eta) : (i,F) \to (j,G)$ is a pair of a functor $\phi : \mathscr{S} \to \mathscr{S}'$ between the apexes of the spans and a natural transformation $\eta : F \implies G\phi$,



(4)

Categories, partial functors and morphisms of partial functors form a monoidal bicategory PCat where the tensor is given pointwise by taking the product of categories and the product of the underlying functors in the spans. Note that full and faithful opisofibrations are closed under composition and stable under pullback.

Definition 13 (Partially Monoidal Category [19])**.** A category $\mathscr{C}$ is partially monoidal if it is equipped with:

- A partial tensor product functor $\boxtimes : \mathscr{C} \times \mathscr{C} \rightharpoonup \mathscr{C}$

- A unit object I

together with associativity and unit natural isomorphisms such that the triangle and pentagon equations hold.

Remark. A very concise definition of a partially monoidal category $\mathscr{C}$ is as a pseudomonoid in PCat

It may not be immediately apparent that there are connections between partially monoidal and promonoidal categories. It turns out though that there is a case where the two coincide on-the-nose.

There exists a special class of partial functors where the left leg is not only an opisofibration but a proper discrete opfibration. This makes the left leg a *cosieve* which coincides with the definition of partial functor given by [7]. Demanding that the left leg is a cosieve ensures that the subcategory on which the tensor is defined is closed under post-composition with morphisms of $\mathscr{C} \times \mathscr{C}$. This captures the following physical intuition: if $X \otimes Y$ exists and there is a morphism $X \to X'$ then $X' \otimes Y$ exists too. Thus we maintain the intuition that if one applies a local map to X then the tensor product should still exist afterwards. From a mathematical perspective, when the left leg of the tensor product partial functor is a cosieve, the partially monoidal category is equivalent to a promonoidal one. Indeed, Bénabou notes that there is an 1-1 correspondence (up to isomorphism) between partial functors with left leg a cosieve and profunctors which factorise through the representable and empty presheaves [7]. In this light the following proposition is not surprising but there is a little effort required in checking that everything works out:

Proposition 7. *A partially monoidal category $(\mathscr{C}, \boxtimes, J)$ whose left leg of the tensor product partial functor is a cosieve is a promonoidal category with representable unit and a tensor $\otimes(-,b,c)$ which is either representable or empty for each $(b,c) \in \mathscr{C} \times \mathscr{C}$.*

Proof. Proof given in appendix B.7. □

There are many examples of partially monoidal categories which are not equivalent to promonoidal ones and vice-versa. For instance, we require that the unit presheaf $J(-)$ of a promonoidal category is representable to have any hope that it is a partially monoidal category.

Conversely, one might hope (similarly to monoidal categories) that all partially monoidal categories could be turned into promonoidal ones. In general this is not possible though as taking the representable presheaves at the defined points of the partial tensor is not enough to define a profunctor $\mathscr{C} \times \mathscr{C} \to\!\!\!\!\!\mapsto \mathscr{C}$. Indeed, a promonoidal category still has a total tensor, just into the presheaf category,

It is possible though to derive partially monoidal structures from a promonoidal one with representable unit presheaf $J(-)$, by pulling back the promonoidal tensor along the Yoneda embedding whenever it is representable. There is of course a canonical "maximal" such partially monoidal structure induced by defining it everywhere it is possible to do so, i.e. everywhere the promonoidal tensor is representable.

One may wonder if there are any further connections between partial functors and profunctors - is there a category that unites them? This would allow us to place the two on equal footing and compare arbitrary partially monoidal and promonoidal categories. The key to this unification is the following result:

Theorem 8 ([7, 27])**.** *There is an equivalence of categories between profunctors $\mathscr{C} \to\!\!\!\!\!\mapsto \mathscr{D}$ and two-sided discrete fibrations* $\mathsf{DFib}(\mathscr{C}, \mathscr{D})$.

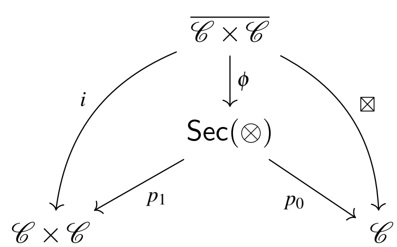
A two-sided discrete fibration is a span of functors $\mathscr{C} \xleftarrow{F} \mathscr{E} \xrightarrow{G} \mathscr{D}$ where:

- each $F(e) \to c'$ in $\mathscr{C}$ has unique lift $f : e \to e'$ in $\mathscr{E}$ such that $G(f) = 1_{G(e)}$,

- each $d \to G(e)$ in $\mathscr{D}$ has unique lift $g : e' \to e$ in $\mathscr{E}$ such that $F(g) = 1_{F(e)}$,
- for each $f : e \to e'$ in $\mathscr{E}$, the codomain of the lift of Ff equals the domain of the lift of Gf, and their composite is f.

The two-sided discrete fibration corresponding to a profunctor $P : \mathscr{C} \nrightarrow \mathscr{D}$ is given by the projections out of the category $\mathsf{Sec}(P)$ of sections of the collage of P. The objects of $\mathsf{Sec}(P)$ are the elements of the sets $P(d,c)$ for all c and d. A morphism $x \in P(d,c) \to x' \in P(d',c')$ is given by a pair of arrows f and g such that $P(g,1)(x') = P(1,f)(x)$.

Consequently, each profunctor has a canonical span and by working in the category of spans of functors one can study the partial functors and profunctors side-by-side. For instance, suppose $(\mathscr{C}, \otimes, J)$ is a promonoidal category with $J(-) \cong \mathscr{C}(-,I)$. There is a partial monoidal structure $(\boxtimes, I)$ on $\mathscr{C}$ given by pulling back $\otimes$ along the Yoneda embedding whenever it is representable - that is, whenever $\otimes(-,b,c) \cong \mathscr{C}(-,x_{bc})$ for some objects b and c, we define $b \boxtimes c := x_{bc}$. Write $\overline{\mathscr{C} \times \mathscr{C}}$ for the subcategory of $\mathscr{C} \times \mathscr{C}$ where the promonoidal tensor is representable. Then there is a 2-cell in $\mathsf{Span}(\mathsf{Cat})$ capturing the extension of the partially monoidal structure on $\mathscr{C}$ to the promonoidal structure:



where ϕ sends (b,c) to $1_{b \boxtimes c, b \boxtimes c} \in \otimes(b \boxtimes c, b, c)$ and (g,f) to $(g \boxtimes f, g, f)$.

B Proofs

B.1 Proof of Proposition 3

Proof. The projections are given by

$$\pi_0 = \mathscr{C}[X] : X \cup Y \to X$$
$$\pi_1 = \mathscr{C}[Y] : X \cup Y \to Y$$

while the coprojections are given by

$$i_0 = \mathscr{C}[X] : X \to X \cup Y$$
$$i_1 = \mathscr{C}[Y] : Y \to X \cup Y$$

Given $f : Z \to X$ and $f' : Z \to Y$, the universal arrow completing the product diagram is $\langle f, f' \rangle = f \cup f' : Z \to X \cup Y$, and given $g : X \to Z$ and $g' : Y \to Z$, the universal arrow completing the coproduct diagram is $[g, g'] = g \cup g' : X \cup Y \to Z$. Indeed, it follows that the diagrams commute because X and Y are jointly spacelike with $X \cap Y = \varnothing$ and thus $f \cap \mathscr{C}[Y] = f' \cap \mathscr{C}[X] = g \cap \mathscr{C}[Y] = g' \cap \mathscr{C}[X] = \varnothing$. $\square$

B.2 Proof of Lemma 2

Proof. Note that the following diagram commutes for any $U : Z' \to Z$

$$\begin{array}{ccc} (X \oslash Y)(Z) & \xrightarrow{(X\oslash Y)(U)} & (X \oslash Y)(Z') \\ {\scriptstyle (S\oslash Y)_Z}\downarrow & & \downarrow{\scriptstyle (S\oslash Y)_{Z'}} \\ (X' \oslash Y)(Z) & \xrightarrow[(X'\oslash Y)(U)]{} & (X' \oslash Y)(Z') \end{array}$$

because on the top path we see $C \mapsto C \cap U \mapsto (C \cap U) \cap S$ while on the bottom path $C \mapsto C \cap S \mapsto (C \cap S) \cap U$. Naturality of $(X \oslash T)$ follows similarly and checking the commutativity condition is straightforward. $\square$

B.3 Proof of Lemma 3

Proof. Firstly note that each component of $1_X \oslash 1_Y : (X \oslash Y)(-) \implies (X \oslash Y)(-)$ is just the identity. Thus it is the identity natural transformation and we conclude $1_X \oslash 1_Y = 1_{(X \oslash Y)(-)}$.

Now take $S : X \to X'$ and $S' : X' \to X''$. The arrow $(S' \oslash Y)_Z \circ (S \oslash Y)_Z$ acts as $C \mapsto (C \cap S) \cap S'$ while the arrow $((S' \circ S) \oslash Y)_Z$ acts as $C \mapsto C \cap (S' \cap S)$. Thus the components of the composite natural transformation $(S' \oslash Y) \circ (S \oslash Y)$ equal those of $((S' \circ S) \oslash Y)$.

A similar argument holds for arrows $T : Y \to Y'$ and because $(S \oslash Y)$ and $(X \oslash T)$ commute we are done. $\square$

B.4 Proof of Theorem 2

Proof. Let us begin with associativity $\oslash(\oslash \times 1) \cong \oslash(1 \times \oslash)$. Note that by Yoneda we have

$$\begin{aligned} \oslash(\oslash \times 1)(W,X,Y,Z) &= \int^{A,B} \oslash(W,A,B) \times \oslash(A,X,Y) \times \mathsf{Slice}(B,Z) \\ &\cong \int^{A} \oslash(W,A,Z) \times \oslash(A,X,Y) \end{aligned}$$

While

$$\oslash(1 \times \oslash)(W,X,Y,Z) \cong \int^{A} \oslash(W,X,A) \times \oslash(A,Y,Z)$$

Let us show there is a canonical identification $\oslash(\oslash \times 1)(W,X,Y,Z) \cong \mathscr{P}(\mathscr{C}[W,X] \cap \mathscr{C}[W,Y] \cap \mathscr{C}[W,Z]) =: \Lambda$. There are functions

$$\oslash(W,A,Z) \times \oslash(A,X,Y) \to \Lambda :: (S,T) \mapsto S \cap T$$

which form a cowedge with apex Λ. By the universal property of the coend this induces a unique function $g : \int^A \oslash(W,A,Z) \times \oslash(A,X,Y) \to \Lambda$ making the obvious cowedge diagrams commute.

We can also construct a function f by composing

$$\Lambda \xrightarrow{f'} \oslash(W,W,Z) \times \oslash(W,X,Y) \xrightarrow{\mathrm{copr}_W} \int^{A} \oslash(W,A,Z) \times \oslash(A,X,Y)$$

where f' acts as $S \mapsto (S,S)$.

The universal property of the coend implies that the composition $fg = 1$, or we can check explicitly:

$$(S,T) \mapsto S\cap T \mapsto (S\cap T, S\cap T)$$

upon which we simply need to note that we have $(S,T) = (S\cap S, T\cap T) \sim (S\cap T, S\cap T)$.

Similarly, it is straightforward to show that $gf = 1$: $S \mapsto (S,S) \mapsto S\cap S = S$. Thus $\Lambda \cong \int^A \oslash(W,A,Z)\times \oslash(A,X,Y)$ as sets.

Now note that this isomorphism is in fact natural in W,X,Y and Z. Let $w : W' \to W, x : X \to X', y : Y \to Y', z : Z \to Z'$, then we have

$$\begin{array}{ccc} (S,T) & \longmapsto & (S\cap w\cap z, T\cap x\cap y) \\ {\scriptstyle g_{WXYZ}}\downarrow & & \downarrow{\scriptstyle g_{W'X'Y'Z'}} \\ S\cap T & \longmapsto & S\cap T\cap w\cap x\cap y\cap z \end{array}$$

Thus exhibiting the desired natural isomorphism.

A similar argument shows that $\oslash(1\times\oslash)(W,X,Y,Z) \cong \Lambda$, and thus we have established the associativity natural isomorphism.

The pentagon equation is given by (writing i for the interchange and ignoring the associativity isomorphisms of profunctor composition):

$$\begin{array}{ccccc} & \oslash^a_{xe}\oslash^x_{yd}\oslash^y_{bc} & \xrightarrow{1\circ\alpha} & \oslash^a_{xe}\oslash^x_{by}\oslash^y_{cd} & \\ \swarrow{\scriptstyle \alpha\circ 1} & & & & \searrow{\scriptstyle \alpha\circ 1} \\ \oslash^a_{yx}\oslash^x_{de}\oslash^y_{bc} & & & & \oslash^a_{bx}\oslash^x_{ye}\oslash^y_{cd} \\ & \searrow{\scriptstyle (\alpha\circ 1)i} & & \swarrow{\scriptstyle 1\circ\alpha} & \\ & & \oslash^a_{bx}\oslash^x_{cy}\oslash^y_{de} & & \end{array}$$

Clockwise we have the following mapping:

$$(S,T,V) \mapsto (S, T\cap V, T\cap V) \mapsto (S\cap T\cap V, S\cap T\cap V, T\cap V) \mapsto (S\cap T\cap V, S\cap T\cap V, S\cap T\cap V)$$

while anticlockwise we have

$$(S,T,V) \mapsto (S\cap T, S\cap T, V) \mapsto (S\cap T\cap V, S\cap T, S\cap T\cap V)$$

and it clear that $(S\cap T\cap V, S\cap T, S\cap T\cap V) \sim (S\cap T\cap V, S\cap T\cap V, S\cap T\cap V)$ under the coend equivalence relation. Thus the pentagon commutes.

Now we show the existence of the unit isomorphisms $\oslash(I\times 1) \cong 1 \cong \oslash(1\times I)$.

Much of the construction is similar to the previous argument, so we leave the reader to fill in some of the details. There exist functions $\oslash(-,=,B)\times\mathscr{P}(\mathscr{C}[B,B]) \to \mathsf{Slice}(-,=)$ for each B given by sending $(S,T) \mapsto S\cap T$. These functions form a cowedge and therefore induce a unique function $\int^B \oslash(-,=,B)\times\mathscr{P}(\mathscr{C}[B,B]) \to \mathsf{Slice}(-,=)$.

The inverse of this function is given by the function $S \mapsto (S,S)$ which factorises via copr. It is straightforward to check that these give the left unit natural isomorphism, and the construction of the right unit is similar.

Writing $\mathbb{Y}$ for an application of the Yoneda lemma, the triangle equation is given by

$$\begin{array}{ccc} & \owedge^{a}_{bc} & \\ {\scriptstyle \mathbb{Y}\rho}\nearrow & & \nwarrow{\scriptstyle \mathbb{Y}\lambda} \\ \owedge^{a}_{xc}\,\owedge^{x}_{by}\,I^{y} & \xrightarrow[\alpha\circ 1]{} & \owedge^{a}_{bx} I^{y} \owedge^{x}_{yc} \end{array}$$

and it is little work to check that this commutes.

The symmetry $(X \owedge Y)(Z) \to (Y \owedge X)(Z)$ is given by the identity map for all X, Y and Z. □

B.5 Proof of Lemma 7

Proof. Take $S : X \to X'$ and $S' : X' \to X''$. Then $(S' \ovee Y)_Z (S \ovee Y)_Z$ acts as $C \mapsto C \cap (S \cup \mathscr{C}[Y]) \cap (S' \cup \mathscr{C}[Y]) = C \cap ((S \cap S') \cup \mathscr{C}[Y])$ which is precisely the same as the action of $(S'S \ovee Y)_Z$. We conclude $(S' \ovee Y)_Z (S \ovee Y)_Z = (S'S \ovee Y)_Z$.

A similar argument shows that $(X \ovee T')_Z (X \ovee T)_Z = (X \ovee T'T)_Z$ and thus we have functoriality of $(- \ovee =)$ in each component. This is enough to extend to functoriality from the funny tensor. □

B.6 Proof of Theorem 7

Proof. (Sketch). The proof is similar and uses the same methods as Theorem 2 so we only sketch the idea.

Fix a slice A. We will show that $(A \ovee -)(-)$ is a kernel.

Starting with the units we need to show that $\int^X \ovee^Z_{AX} J^X \cong J^Z$. There are functions $\ovee^Z_{AX} J^X \to J^Z$ sending $(S, T) \mapsto S \cap (T \cup \mathscr{C}[A])$. These are dinatural in X and thus form a cowedge factorising uniquely via the coend. As a result we have a function $\int^X \ovee^Z_{AX} J^X \to J^Z$. This function is an isomorphism with inverse given by $S \mapsto (S, S)$ which factorises via copr. Indeed,

$$S \mapsto (S, S) \mapsto S \cap (S \cup \mathscr{C}[A]) = S$$

and

$$\begin{aligned} (S, T) \mapsto S \cap (T \cup \mathscr{C}[A]) &\mapsto (S \cap (T \cup \mathscr{C}[A]), S \cap (T \cup \mathscr{C}[A])) \\ &\sim (S \cap (S \cup \mathscr{C}[A]), T \cap T) \\ &= (S, T) \end{aligned}$$

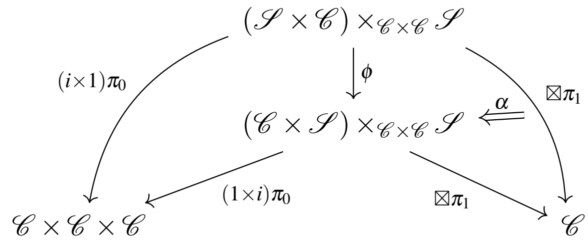
As for the multiplications we want to show $\int^Z \ovee^W_{AZ} \owedge^Z_{XY} \cong \int^{ZZ'} \ovee^Z_{AX} \ovee^{Z'}_{AY} \owedge^W_{ZZ'}$ which it is easiest to do by showing each is naturally isomorphic to $\Lambda := \mathscr{P}(\mathscr{C}[W, A] \cup (\mathscr{C}[W, X] \cap \mathscr{C}[W, Y]))$. For the former, there is a cowedge with components $(S, T) \mapsto S \cap (T \cup \mathscr{C}[A])$, with the inverse to the induced map given by $S \mapsto (S, S)$, as in the case of the units. For the latter, there is a cowedge with components $(S, T, V) \mapsto S \cap T \cap V$, with the inverse to the induced map given by $S \mapsto (S, S, S)$.

To show that all the isomorphisms are natural is little work. □

B.7 Proof of Proposition 7

Proof. In a slight abuse of notation write $\mathscr{C} \times \mathscr{C} \xleftarrow{i} \mathscr{S} \xrightarrow{\boxtimes} \mathscr{C}$ for the underlying span of the partial functor $\boxtimes : \mathscr{C} \times \mathscr{C} \rightharpoonup \mathscr{C}$, and note that $J : 1 \rightharpoonup \mathscr{C}$ is simply a normal functor $J : 1 \to \mathscr{C}$, in other words an object J of $\mathscr{C}$. Just like for monoidal categories we can define a promonoidal structure on $\mathscr{C}$ by taking $(X\otimes Y)(Z) := \mathscr{C}(Z, X\boxtimes Y)$ whenever $(X,Y)\in\mathscr{S}$ and $(X\otimes Y)(Z) := \varnothing$ otherwise. The unit is the representable presheaf at J, $\mathbb{Y}_J$.

The associativity isomorphism of a partially monoidal category induces the following arrows:

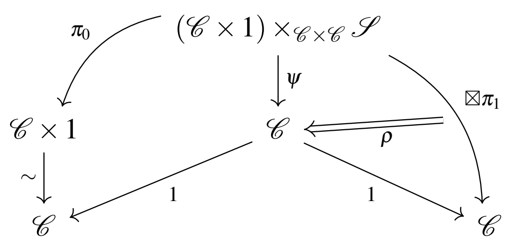


(5)

where π_0 and π_1 are the canonical projections from the pullback and α is a natural isomorphism.

Given a cospan of functors $\mathscr{C} \xrightarrow{F} \mathscr{E} \xleftarrow{G} \mathscr{D}$, the pullback $\mathscr{C}\times_{\mathscr{E}}\mathscr{D}$ is the category consisting of pairs of objects (c,d) with $Fc = Gd$ and pairs of morphisms (f,g) with $Ff = Gg$. We can think of $(\mathscr{S}\times\mathscr{C})\times_{\mathscr{C}\times\mathscr{C}}\mathscr{S}$ as the category with objects $(((a,b),c),(a\boxtimes b,c))$ where $(a,b)\in\mathscr{S}$ and $c\in\mathscr{C}$ with $(a\boxtimes b,c)\in\mathscr{S}$, while $(\mathscr{C}\times\mathscr{S})\times_{\mathscr{C}\times\mathscr{C}}\mathscr{S}$ has objects $((a,(b,c)),(a,b\boxtimes c))$ where $(b,c)\in\mathscr{S}$ and $a\in\mathscr{C}$ with $(a,b\boxtimes c)\in\mathscr{S}$. The left triangle of (5) ensures that ϕ must act to send $(((a,b),c),(a\boxtimes b,c))\mapsto((a,(b,c)),(a,b\boxtimes c))$. The right triangle of (5) then implies that the components of α have type $\alpha_{a,b,c} : (a\boxtimes b)\boxtimes c \to a\boxtimes(b\boxtimes c)$. This induces the necessary isomorphism $\otimes^a_{xd}\otimes^x_{bc} \to \otimes^a_{bx}\otimes^x_{cd}$ and checking the pentagon coherence equation now follows the same standard proof as Theorem 1.

The right unit isomorphism induces the following arrows:



the components of ρ have type $\rho_a : a\boxtimes J \to a$ as expected. A similar diagram is induced by λ and in turn one sees that this has components $\lambda_a : J\boxtimes a \to a$. Checking the triangle coherence equation follows like Theorem 1.

Now suppose we begin with a promonoidal category $\mathscr{C}$ where the unit is representable $J(-)\cong\mathscr{C}(-,I)$ and for each $(b,c)\in\mathscr{C}\times\mathscr{C}$, either $\otimes(-,b,c)\cong\mathscr{C}(-,x_{bc})$ is representable, or $\otimes(-,b,c)\cong\varnothing(-)$ is empty. Define a full subcategory $\mathscr{S}$ of $\mathscr{C}\times\mathscr{C}$ spanned by objects (b,c) where $\otimes(-,b,c)$ is representable. Suppose for a contradiction that $(b,c)\in\mathscr{S}$ and there exists a $(f,g) : (b,c)\to(b',c')$ in $\mathscr{C}\times\mathscr{C}$ but with $(b',c')\notin\mathscr{S}$. Then we would have a natural transformation $\mathscr{C}(-,x_{bc})\to\varnothing(-)$, a contradiction. Thus (f,g) cannot exist and as a result the canonical inclusion functor $\mathscr{S}\hookrightarrow\mathscr{C}\times\mathscr{C}$ is a discrete opfibration. $\square$

Open Dynamical Systems as Coalgebras for Polynomial Functors, with Application to Predictive Processing

Toby St. Clere Smithe
Topos Institute
toby@topos.institute

We present categories of open dynamical systems with general time evolution as categories of coalgebras opindexed by polynomial interfaces, and show how this extends the coalgebraic framework to capture common scientific applications such as ordinary differential equations, open Markov processes, and random dynamical systems. We then extend Spivak's operad **Org** to this setting, and construct associated monoidal categories whose morphisms represent hierarchical open systems; when their interfaces are simple, these categories supply canonical comonoid structures. We exemplify these constructions using the 'Laplace doctrine', which provides dynamical semantics for active inference, and indicate some connections to Bayesian inversion and coalgebraic logic.

1 Background

1.1 Closed dynamical systems and Markov processes

In this brief section, we recall a 'behavioural' approach to dynamical systems originally due (we believe) to Lawvere; for a pedagogical account, see [1]. These systems are 'closed' in the sense that they do not require environmental interaction for their evolution, but they nonetheless form the starting point for our categories of more open systems.

Definition 1.1. Let $(\mathbb{T},+,0)$ be a monoid, representing time. Let $X : \mathscr{E}$ be some space, called the *state space*. Then a *closed dynamical system* ϑ *with state space* X *and time* $\mathbb{T}$ is an action of $\mathbb{T}$ on X. When $\mathbb{T}$ is also an object of $\mathscr{E}$, then this amounts to a morphism $\vartheta : \mathbb{T} \times X \to X$ (or equivalently, a time-indexed family of X-endomorphisms, $\vartheta(t) : X \to X$), such that $\vartheta(0) = \mathsf{id}_X$ and $\vartheta(s+t) = \vartheta(s) \circ \vartheta(t)$.

Proposition 1.2. When time is discrete, as in the case $\mathbb{T} = \mathbb{N}$, any dynamical system ϑ is entirely determined by its action at $1 : \mathbb{T}$. That is, letting the state space be X, we have $\vartheta(t) = \vartheta(1)^{\circ t}$ where $\vartheta(1)^{\circ t}$ means "compose $\vartheta(1) : X \to X$ with itself t times".

Example 1.3. Suppose $X : U \to TU$ is a vector field on U, with a corresponding solution (integral curve) $\chi_x : \mathbb{R} \to U$ for all $x : U$; that is, $\chi'(t) = X(\chi_x(t))$ and $\chi_x(0) = x$. Then letting the point x vary, we obtain a map $\chi : \mathbb{R} \times U \to U$. This χ is a closed dynamical system with state space U and time $\mathbb{R}$.

Proposition 1.4. Closed dynamical systems with state spaces in $\mathscr{E}$ and time $\mathbb{T}$ are the objects of the functor category $\mathbf{Cat}(\mathbf{B}\mathbb{T}, \mathscr{E})$, where $\mathbf{B}\mathbb{T}$ denotes the delooping of the monoid $\mathbb{T}$. Morphisms of dynamical systems are therefore natural transformations.

We will also often be interested in dynamical systems whose evolution has 'side-effects', such as the generation (or 'mixing') of uncertainty or randomness. We will largely model such systems as Kleisli maps or coalgebras of monads modelling these side-effects. In the case of uncertainty, the monads will be so-called *probability monads*, which we will often denote by $\mathscr{P}$. Such a monad $\mathscr{P} : \mathscr{E} \to \mathscr{E}$ can often be thought of as taking each set or space $X : \mathscr{E}$ to the set (or space) $\mathscr{P}X$ of probability distributions over

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 307–330, doi:10.4204/EPTCS.380.18

X, and each morphism to the corresponding 'pushforwards' map; the monad multiplication is given by "averaging out" uncertainty, and the unit takes a point to the 'Dirac' distribution over it. With these ideas in mind, we can extend the concepts above to cover Markov chains and Markov processes.

Example 1.5 (Closed Markov chains and Markov processes)**.** A closed *Markov chain* is given by a map $X \to \mathscr{P}X$, where $\mathscr{P} : \mathscr{E} \to \mathscr{E}$ is a probability monad on $\mathscr{E}$; this is equivalently a $\mathscr{P}$-coalgebra with time $\mathbb{N}$, and an object in $\mathbf{Cat}\big(\mathbf{B}\mathbb{N}, \mathscr{K}\ell(\mathscr{P})\big)$. With more general time $\mathbb{T}$, one obtains closed *Markov processes*: objects in $\mathbf{Cat}\big(\mathbf{B}\mathbb{T}, \mathscr{K}\ell(\mathscr{P})\big)$. More explicitly, a closed Markov process is a time-indexed family of Markov kernels; that is, a morphism $\vartheta : \mathbb{T} \times X \to \mathscr{P}X$ such that, for all times $s,t : \mathbb{T}$, $\vartheta_{s+t} = \vartheta_s \bullet \vartheta_t$ as a morphism in $\mathscr{K}\ell(\mathscr{P})$. Note that composition $\bullet$ in $\mathscr{K}\ell(\mathscr{P})$ is given by the Chapman-Kolmogorov equation, so this means that

$$\vartheta_{s+t}(y|x) = \int_{x':X} \vartheta_s(y|x')\, \vartheta_t(\mathrm{d}x'|x)\,.$$

1.2 Polynomial functors

We will use *polynomial functors* to model the interfaces of our open systems, following Spivak and Niu [2]. We will assume these to be functors $\mathscr{E} \to \mathscr{E}$ for a locally Cartesian closed category $\mathscr{E}$, but we will typically assume that $\mathscr{E}$ is furthermore concrete, and often that it is in fact **Set**.

Definition 1.6. Let $\mathscr{E}$ be a locally Cartesian closed category, and denote by y^A the representable copresheaf $y^A := \mathscr{E}(A,-) : \mathscr{E} \to \mathscr{E}$. A *polynomial functor* p is a coproduct of representable functors, written $p := \sum_{i:p(1)} y^{p_i}$, where $p(1) : \mathscr{E}$ is the indexing object. The category of polynomial functors in $\mathscr{E}$ is the full subcategory $\mathbf{Poly}_{\mathscr{E}} \hookrightarrow [\mathscr{E},\mathscr{E}]$ of the $\mathscr{E}$-copresheaf category spanned by coproducts of representables. A morphism of polynomials is therefore a natural transformation.

Remark 1.7. Every polynomial functor $P : \mathscr{E} \to \mathscr{E}$ corresponds to a bundle $p : E \to B$ in $\mathscr{E}$, for which $B = P(1)$ and for each $i : P(1)$, the fibre p_i is $P(i)$. We will henceforth elide the distinction between a copresheaf P and its corresponding bundle p, writing $p(1) := B$ and $p[i] := p_i$, where $E = \sum_i p[i]$. A natural transformation $f : p \to q$ between copresheaves therefore corresponds to a map of bundles. In the case of polynomials, by the Yoneda lemma, this map is given by a 'forwards' map $f_1 : p(1) \to q(1)$ and a family of 'backwards' maps $f^\# : q[f_1(\text{-})] \to p[\text{-}]$ indexed by $p(1)$, as in the left diagram below. Given $f : p \to q$ and $g : q \to r$, their composite $g \circ f : p \to r$ is as in the right diagram below.

$$\begin{array}{ccccc} E & \xleftarrow{f^\#} & f^*F & \longrightarrow & F \\ {\scriptstyle p}\big\downarrow & & \big\downarrow & \lrcorner & \big\downarrow{\scriptstyle q} \\ B & = & B & \xrightarrow{f_1} & C \end{array} \qquad\qquad \begin{array}{ccccc} E & \xleftarrow{(gf)^\#} & f^*g^*G & \longrightarrow & G \\ {\scriptstyle p}\big\downarrow & & \big\downarrow & \lrcorner & \big\downarrow{\scriptstyle r} \\ B & = & B & \xrightarrow{g_1 \circ f_1} & D \end{array}$$

where $(gf)^\#$ is given by the $p(1)$-indexed family of composite maps $r[g_1(f_1(\text{-}))] \xrightarrow{f^*g^\#} q[f_1(\text{-})] \xrightarrow{f^\#} p[\text{-}]$.

We can interpret the type $p(1)$ to be a set or space of 'configurations' or 'positions' of a p-shaped system, and each $p[i]$ to be the available 'inputs' or 'directions' available to the system when it is in configuration/position i.

We now recall a handful of useful facts about polynomials and their morphisms, each of which is explained in Spivak and Niu [2] and summarized in Spivak [3].

Proposition 1.8. Polynomial morphisms $p \to y$ correspond to sections $p(1) \to \sum_i p[i]$ of the corresponding bundle p.

Proposition 1.9. There is an embedding of $\mathscr{E}$ into $\mathbf{Poly}_{\mathscr{E}}$ given by taking objects $X : \mathscr{E}$ to the linear polynomials $Xy : \mathbf{Poly}_{\mathscr{E}}$ and morphisms $f : X \to Y$ to morphisms $(f, \mathsf{id}_X) : Xy \to Yy$.

Proposition 1.10. There is a symmetric monoidal structure $(\otimes, y)$ on $\mathbf{Poly}_{\mathscr{E}}$ that we call tensor, and which is given on objects by $p \otimes q := \sum_{i:p(1)} \sum_{j:q(1)} y^{p[i] \times q[j]}$ and on morphisms $f := (f_1, f^\#) : p \to p'$ and $g := (g_1, g^\#) : q \to q'$ by $f \otimes g := (f_1 \times g_1, f^\# \times g^\#)$.

Proposition 1.11. $(\mathbf{Poly}_{\mathscr{E}}, \otimes, y)$ is symmetric monoidal closed, with internal hom denoted $[-,=]$. Explicitly, we have $[p,q] = \sum_{f:p \to q} y^{\sum_{i:p(1)} q[f_1(i)]}$. Given an object $A : \mathscr{E}$, we have $[Ay, y] \cong y^A$.

Proposition 1.12. The composition of polynomial functors $q \circ p : \mathscr{E} \to \mathscr{E} \to \mathscr{E}$ induces a monoidal structure on $\mathbf{Poly}_{\mathscr{E}}$, which we denote $\triangleleft$, and call 'composition' or 'substitution'. Its unit is again y. Famously, $\triangleleft$-comonoids correspond to categories and their comonoid homomorphisms are cofunctors [4]. If $\mathbb{T}$ is a monoid, then the comonoid structure on $y^{\mathbb{T}}$ corresponds witnesses it as the category $\mathbf{B}\mathbb{T}$. Monomials of the form Sy^S can be equipped with a canonical comonoid structure witnessing the codiscrete groupoid on S.

2 Open dynamical systems as polynomial coalgebras

2.1 Deterministic systems

Definition 2.1. A deterministic open dynamical system with interface p, state space S and time $\mathbb{T}$ is a polynomial morphism $\beta : Sy^S \to [\mathbb{T}y, p]$ such that, for any section $\sigma : p \to y$, the induced morphism

$$Sy^S \xrightarrow{\beta} [\mathbb{T}y, p] \xrightarrow{[\mathbb{T}y, \sigma]} [\mathbb{T}y, y] \xrightarrow{\sim} y^{\mathbb{T}}$$

is a comonoid homomorphism.

To see how such a morphism β is like an 'open' version of the closed dynamical systems introduced above, note that by the tensor-hom adjunction, β can equivalently be written with the type $\mathbb{T}y \otimes Sy^S \to p$. In turn, such a morphism corresponds to a pair (β^o, β^u), where β^o is the component 'on positions' with the type $\mathbb{T} \times S \to p(1)$, and β^u is the component 'on directions' with the type $\sum_{t:\mathbb{T}} \sum_{s:S} p[\beta^o(t,s)] \to S$. We will call the map β^o the *output map*, as it chooses an output position for each state and moment in time; and we will call the map β^u the *update map*, as it takes a state $s : S$, a quantity of time $t : \mathbb{T}$, and an 'input' in $p[\beta^o(t,s)]$, and returns a new state. We might imagine the new state as being given by evolving the system from s for time t, and the input as being given at the position corresponding to (s,t).

But it is not sufficient to consider merely such pairs $\beta = (\beta^o, \beta^u)$ to be our open dynamical systems, for we need them to be like 'open' monoid actions: evolving for time t then for time s must be equivalent to evolving for time $t+s$, given the same inputs. It is fairly easy to prove the following proposition, whose proof we defer until after establishing the categories $\mathbf{Coalg}^{\mathbb{T}}(p)$.

Proposition 2.2. Comonoid homomorphisms $Sy^S \to y^{\mathbb{T}}$ correspond bijectively with closed dynamical systems with state space $S : \mathscr{E}$, in the sense given by functors $\mathbf{B}\mathbb{T} \to \mathscr{E}$.

This establishes that seeking such a comonoid homomorphism will give us the monoid action property that we seek, and so it remains to show that a composite comonoid homomorphism of the form $[\mathbb{T}y, \sigma] \circ \beta$ is a closed dynamical system with the "right inputs". Unwinding this composite, we find that the condition that it be a comonoid homomorphism corresponds to the requirement that, for any $t : \mathbb{T}$, the *closure* $\beta^\sigma : \mathbb{T} \times S \to S$ of β by σ given by

$$\beta^\sigma(t) := S \xrightarrow{\beta^o(t)^* \sigma} \sum_{s:S} p[\beta^o(t,s)] \xrightarrow{\beta^u} S$$

constitutes a closed dynamical system on S. The idea here is that σ gives the 'context' in which we can make an open system closed, thereby formalizing the "given the same inputs" requirement above.

With this conceptual framework in mind, we are in a position to render open dynamical systems on p with time $\mathbb{T}$ into a category, which we will denote by $\mathbf{Coalg}^{\mathbb{T}}(p)$. Its objects will be pairs (S, β) with $S : \mathscr{E}$ and β an open dynamical on p with state space S; we will often write these pairs equivalently as triples (S, β^o, β^u), making explicit the output and update maps. Morphisms will be maps of state spaces that commute with the dynamics:

Proposition 2.3. Open dynamical systems over p with time $\mathbb{T}$ form a category, denoted $\mathbf{Coalg}^{\mathbb{T}}(p)$. Its morphisms are defined as follows. Let $\vartheta := (X, \vartheta^o, \vartheta^u)$ and $\psi := (Y, \psi^o, \psi^u)$ be two dynamical systems over p. A morphism $f : \vartheta \to \psi$ consists in a morphism $f : X \to Y$ such that, for any time $t : \mathbb{T}$ and global section $\sigma : p(1) \to \sum\limits_{i:p(1)} p[i]$ of p, the following naturality squares commute:

$$\begin{array}{ccccc} X & \xrightarrow{\vartheta^o(t)^*\sigma} & \sum\limits_{x:X} p[\vartheta^o(t,x)] & \xrightarrow{\vartheta^u(t)} & X \\ {\scriptstyle f}\downarrow & & & & \downarrow{\scriptstyle f} \\ Y & \xrightarrow[\psi^o(t)^*\sigma]{} & \sum\limits_{y:Y} p[\psi^o(t,y)] & \xrightarrow[\psi^u(t)]{} & Y \end{array}$$

The identity morphism id_ϑ on the dynamical system ϑ is given by the identity morphism id_X on its state space X. Composition of morphisms of dynamical systems is given by composition of the morphisms of the state spaces.

Proof. We need to check unitality and associativity of composition. This amounts to checking that the composite naturality squares commute. But this follows immediately, since the composite of two commutative diagrams along a common edge is again a commutative diagram. □

We can alternatively state Proposition 2.2 as follows, noting that the polynomial y corresponds to a trivial interface, exposing no configuration to any environment nor receiving any signals from it:

Proposition 2.4. $\mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(y)$ is equivalent to the classical category $\mathbf{Cat}(\mathbf{B}\mathbb{T}, \mathscr{E})$ of closed dynamical systems in $\mathscr{E}$ with time $\mathbb{T}$.

Proof. The trivial interface y corresponds to the trivial bundle $\mathsf{id}_1 : 1 \to 1$. Therefore, a dynamical system over y consists of a choice of state space S along with a trivial output map $\vartheta^o = \overline{\top} : \mathbb{T} \times S \to 1$ and a time-indexed update map $\vartheta^u : \mathbb{T} \times S \to S$. This therefore has the form of a classical closed dynamical system, so it remains to check the monoid action. There is only one section of id_1, which is again id_1. Pulling this back along the unique map $\vartheta^o(t) : S \to 1$ gives $\vartheta^o(t)^* \mathsf{id}_1 = \mathsf{id}_S$. Therefore the requirement that, given any section σ of y, the maps $\vartheta^u \circ \vartheta^o(t)^*\sigma$ form an action means in turn that so does $\vartheta^u : \mathbb{T} \times S \to S$. Since the pullback of the unique section id_1 along the trivial output map $\vartheta^o(t) = \overline{\top} : S \to 1$ of any dynamical system in $\mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(y)$ is the identity of the corresponding state space id_S, a morphism $f : (\vartheta(*), \vartheta^u, \overline{\top}) \to (\psi(*), \psi^u, \overline{\top})$ in $\mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(y)$ amounts precisely to a map $f : \vartheta(*) \to \psi(*)$ on the state spaces in $\mathscr{E}$ such that the naturality condition $f \circ \vartheta^u(t) = \psi^u(t) \circ f$ of Proposition 1.4 is satisfied, and every morphism in $\mathbf{Cat}(\mathbf{B}\mathbb{T}, \mathscr{E})$ corresponds to a morphism in $\mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(y)$ in this way. □

Now that we know that our concept of open dynamical system subsumes closed systems, let us consider some more examples.

Example 2.5. Consider a dynamical system $(S, \vartheta^o, \vartheta^u)$ with outputs but no inputs. Such a system has a 'linear' interface $p := Iy$ for some $I : \mathscr{E}$; alternatively, we can write its interface p as the 'bundle' $\mathsf{id}_I : I \to I$. A section of this bundle must again be id_I, and so $\vartheta^o(t)^* \mathsf{id}_I = \mathsf{id}_S$. Once again, the update maps collect into to a closed dynamical system in $\mathbf{Cat}(\mathbf{B}\mathbb{T}, \mathscr{E})$; just now we have outputs $\vartheta^o : \mathbb{T} \times S \to p(1) = I$ exposed to the environment.

Proposition 2.6. When time is discrete, as with $\mathbb{T} = \mathbb{N}$, any open dynamical system $(X, \vartheta^o, \vartheta^u)$ over p is entirely determined by its components at $1 : \mathbb{T}$. That is, we have $\vartheta^o(t) = \vartheta^o(1) : X \to p(1)$ and $\vartheta^u(t) = \vartheta^u(1) : \sum_{x:X} p[\vartheta^o(x)] \to X$. A discrete-time open dynamical system is therefore a triple $(X, \vartheta^o, \vartheta^u)$, where the two maps have types $\vartheta^o : X \to p(1)$ and $\vartheta^u : \sum_{x:X} p[\vartheta^o(x)] \to X$.

Proof. Suppose σ is a section of p. We require each closure ϑ^σ to satisfy the flow conditions, that $\vartheta^\sigma(0) = \mathsf{id}_X$ and $\vartheta^\sigma(t+s) = \vartheta^\sigma(t) \circ \vartheta^\sigma(s)$. In particular, we must have $\vartheta^\sigma(t+1) = \vartheta^\sigma(t) \circ \vartheta^\sigma(1)$. By induction, this means that we must have $\vartheta^\sigma(t) = \vartheta^\sigma(1)^{\circ t}$ (compare Proposition 1.2). Therefore we must in general have $\vartheta^o(t) = \vartheta^o(1)$ and $\vartheta^u(t) = \vartheta^u(1)$. □

Example 2.7. Suppose $\dot{x} = f(x, a)$ and $b = g(x)$, with $f : X \times A \to TX$ and $g : X \to B$. Then, as for the 'closed' vector fields of Example 1.3, this induces an open dynamical system $(X, \int f, g) : \mathbf{Coalg}^{\mathbb{R}}(By^A)$, where $\int f : \mathbb{R} \times X \times A \to X$ returns the (X, A)-indexed solutions of f.

Example 2.8. The preceding example is easily extended to the case of a general polynomial interface. Suppose similarly that $\dot{x} = f(x, a_x)$ and $b = g(x)$, now with $f : \sum_{x:X} p[g(x)] \to TX$ and $g : X \to p(1)$. Then we obtain an open dynamical system $(X, \int f, g) : \mathbf{Coalg}^{\mathbb{R}}_{\mathsf{id}}(p)$, where now $\int f : \mathbb{R} \times \sum_{x:X} p[g(x)] \to X$ is the 'update' and $g : X \to p(1)$ the 'output' map.

It is quite straightforward to extend the construction of $\mathbf{Coalg}^{\mathbb{T}}(p)$ to an opindexed category $\mathbf{Coalg}^{\mathbb{T}}$; we unravel this opindexing explicitly in the appendix (Proposition A.1).

Proposition 2.9. $\mathbf{Coalg}^{\mathbb{T}}$ extends to an opindexed category $\mathbf{Coalg}^{\mathbb{T}} : \mathbf{Poly}_{\mathscr{E}} \to \mathbf{Cat}$. On objects (polynomials), it returns the categories above. On morphisms of polynomials, we simply post-compose: given $\varphi : p \to q$ and $\beta : Sy^S \to [\mathbb{T}y, p]$, obtain $Sy^S \to [\mathbb{T}y, p] \to [\mathbb{T}y, q]$ in the obvious way.

At this point, the reader may be wondering in what sense these open dynamical systems are coalgebras. To see this, observe that a polynomial morphism $Sy^S \to q$ is equivalently a map $S \to q(S)$: that is to say, a q-coalgebra. By setting $q = [\mathbb{T}y, p]$, we see the connection immediately; to make it clear, in Proposition A.2, we spell it out for the case $\mathbb{T} = \mathbb{N}$.

2.2 Open Markov processes via stochastic polynomials

Just as coalgebras $S \to pS$ correspond to discrete-time deterministic open dynamical systems, coalgebras $S \to p\mathscr{P}S$ correspond to discrete-time *stochastic* dynamical systems when $\mathscr{P}$ is a probability monad as introduced above. We have already seen that 'closed' Markov chains correspond to maps $S \to \mathscr{P}S$, and that Markov processes in general time correspond to functors $\mathbf{B}\mathbb{T} \to \mathscr{K}\ell(\mathscr{P})$. Our task in this section is therefore to connect these two perspectives, extending the categories of deterministic coalgebras $\mathbf{Coalg}^{\mathbb{T}}(p)$.

Working concretely, it is not hard to spot the relevant adjustment. We therefore make the following definition.

Definition 2.10. Let $M : \mathscr{E} \to \mathscr{E}$ be a monad on the category $\mathscr{E}$, and let $p : \mathbf{Poly}_{\mathscr{E}}$ be a polynomial in $\mathscr{E}$. Let $(\mathbb{T}, +, 0)$ be a monoid in $\mathscr{E}$, representing time. Then a *pM-coalgebra with time* $\mathbb{T}$ consists in a triple $\vartheta :=$

$(S, \vartheta^o, \vartheta^u)$ of a state space $S : \mathscr{E}$ and two morphisms $\vartheta^o : \mathbb{T} \times S \to p(1)$ and $\vartheta^u : \sum_{t:\mathbb{T}} \sum_{s:\mathbb{S}} p[\vartheta^o(t,s)] \to MS$, such that, for any section $\sigma : p(1) \to \sum_{i:p(1)} p[i]$ of p, the maps $\vartheta^\sigma : \mathbb{T} \times S \to MS$ given by

$$\sum_{t:\mathbb{T}} S \xrightarrow{\vartheta^o(-)^*\sigma} \sum_{t:\mathbb{T}} \sum_{s:S} p[\vartheta^o(-,s)] \xrightarrow{\vartheta^u} MS$$

constitute an object in the functor category $\mathbf{Cat}\big(\mathbf{B}\mathbb{T}, \mathscr{K}\ell(T)\big)$, where $\mathbf{B}\mathbb{T}$ is the delooping of $\mathbb{T}$ and $\mathscr{K}\ell(T)$ is the Kleisli category of T. Once more, we call the closed system ϑ^σ, induced by a section σ of p, the closure of ϑ by σ.

As before, such pM-coalgebras form a category; and these categories in turn are opindexed by polynomials.

Proposition 2.11. *pM-coalgebras with time $\mathbb{T}$ form a category, denoted $\mathbf{Coalg}^{\mathbb{T}}_M(p)$. Its morphisms are defined as follows. Let $\vartheta := (X, \vartheta^o, \vartheta^u)$ and $\psi := (Y, \psi^o, \psi^u)$ be two pM-coalgebras. A morphism $f : \vartheta \to \psi$ consists in a morphism $f : X \to Y$ such that, for any time $t : \mathbb{T}$ and global section $\sigma : p(1) \to \sum_{i:p(1)} p[i]$ of p, the following naturality squares commute:*

$$\begin{array}{ccccc} X & \xrightarrow{\vartheta^o(t)^*\sigma} & \sum_{x:X} p[\vartheta^o(t,x)] & \xrightarrow{\vartheta^u(t)} & MX \\ {\scriptstyle f}\downarrow & & & & \downarrow{\scriptstyle Mf} \\ Y & \xrightarrow[\psi^o(t)^*\sigma]{} & \sum_{y:Y} p[\psi^o(t,y)] & \xrightarrow[\psi^u(t)]{} & MY \end{array}$$

The identity morphism id_ϑ on the pM-coalgebra ϑ is given by the identity morphism id_X on its state space X. Composition of morphisms of pM-coalgebras is given by composition of the morphisms of the state spaces.

Proposition 2.12. *$\mathbf{Coalg}^{\mathbb{T}}_M(p)$ extends to an opindexed category, $\mathbf{Coalg}^{\mathbb{T}}_M(-) : \mathbf{Poly}_{\mathscr{E}} \to \mathbf{Cat}$. Suppose $\varphi : p \to q$ is a morphism of polynomials. We define a corresponding functor $\mathbf{Coalg}^{\mathbb{T}}_M(\varphi) : \mathbf{Coalg}^{\mathbb{T}}_M(p) \to \mathbf{Coalg}^{\mathbb{T}}_M(q)$ as follows. Suppose $(X, \vartheta^o, \vartheta^u) : \mathbf{Coalg}^{\mathbb{T}}_M(p)$ is an object (pM-coalgebra) in $\mathbf{Coalg}^{\mathbb{T}}_M(p)$. Then $\mathbf{Coalg}^{\mathbb{T}}_M(\varphi)(X, \vartheta^o, \vartheta^u)$ is defined as the triple $(X, \varphi_1 \circ \vartheta^o, \vartheta^u \circ \vartheta^{o*}\varphi^\#) : \mathbf{Coalg}^{\mathbb{T}}_M(q)$, where the two maps are explicitly the following composites:*

$$\mathbb{T} \times X \xrightarrow{\vartheta^o} p(1) \xrightarrow{\varphi_1} q(1), \qquad \sum_{t:\mathbb{T}} \sum_{x:X} q[\varphi_1 \circ \vartheta^o(t,x)] \xrightarrow{\vartheta^{o*}\varphi^\#} \sum_{t:\mathbb{T}} \sum_{x:X} p[\vartheta^o(t,x)] \xrightarrow{\vartheta^u} MX\,.$$

On morphisms, $\mathbf{Coalg}^{\mathbb{T}}_M(\varphi)(f) : \mathbf{Coalg}^{\mathbb{T}}_M(\varphi)(X, \vartheta^o, \vartheta^u) \to \mathbf{Coalg}^{\mathbb{T}}_M(\varphi)(Y, \psi^o, \psi^u)$ is given by the same underlying map $f : X \to Y$ of state spaces.

The opindexed category $\mathbf{Coalg}^{\mathbb{T}}_M(p)$ clearly generalizes $\mathbf{Coalg}^{\mathbb{T}}$, since we can always take $M = \mathsf{id}_{\mathscr{E}}$. Yet these concrete definitions obscure the more elegant representation of the objects of $\mathbf{Coalg}^{\mathbb{T}}$ as morphisms $Sy^S \to [\mathbb{T}y, p]$. Our task is therefore to find a setting in which a similar representation is possible; to do so, we generalize $\mathbf{Poly}_{\mathscr{E}}$ so that the backwards components of its morphisms may incorporate 'side-effects' modelled by M. We will call the corresponding category $\mathbf{Poly}_M$, and will find that instantiating $\mathbf{Coalg}^{\mathbb{T}}$ in $\mathbf{Poly}_M$ recovers $\mathbf{Coalg}^{\mathbb{T}}_M(p)$.

We begin by recalling that $\mathbf{Poly}_{\mathscr{E}}$ is equivalent to the category of Grothendieck lenses for the self-indexing [2, 5]: $\mathbf{Poly}_{\mathscr{E}} \cong \int \mathscr{E}/-^{\mathrm{op}}$, where the opposite is taken pointwise on each $\mathscr{E}/B$. We will define

$\mathbf{Poly}_M$ by analogy, using the following indexed category. Suppose M is a commutative monad on $\mathscr{E}$ and let ι denote the identity-on-objects inclusion $\mathscr{E} \hookrightarrow \mathscr{K}\ell(M)$ given on morphisms by post-composing with the unit η of the monad structure. For ease of exposition in this short paper, we will assume here that $\mathscr{E} = \mathbf{Set}$.

Definition 2.13. Define the indexed category $\mathscr{E}_M/- : \mathscr{E}^{\,\mathrm{op}} \to \mathbf{Cat}$ as follows. On objects $B : \mathscr{E}$, we define $\mathscr{E}_M/B$ to be the full subcategory of $\mathscr{K}\ell(M)/B$ on those objects $\iota p : E \twoheadrightarrow B$ which correspond to maps $E \xrightarrow{p} B \xrightarrow{\eta_B} MB$ in the image of ι. Now suppose $f : C \to B$ is a map in $\mathscr{E}$. We define $\mathscr{E}_M/f : \mathscr{E}_M/B \to \mathscr{E}_M/C$ as follows. The functor $\mathscr{E}_M/f$ takes objects $\iota p : E \twoheadrightarrow B$ to $\iota(f^*p) : f^*E \twoheadrightarrow C$ where f^*p is the pullback of p along f in $\mathscr{E}$, included into $\mathscr{K}\ell(M)$ by ι.

To define the action of $\mathscr{E}_M/f$ on morphisms $\alpha : (E, \iota p : E \twoheadrightarrow B) \to (F, \iota q : F \twoheadrightarrow B)$, note that since we must have $\iota q \bullet \alpha = \iota p$, α must correspond to a family of maps $\alpha_x : p[x] \to Mq[x]$ for $x : B$. Then we can define $(\mathscr{E}_M/f)(\alpha)$ pointwise as $(\mathscr{E}_M/f)(\alpha)_y := \alpha_{f(y)} : p[f(y)] \to Mq[f(y)]$ for $y : C$.

Definition 2.14. We define $\mathbf{Poly}_M$ to be the category of Grothendieck lenses for $\mathscr{E}_M/-$. That is, $\mathbf{Poly}_M := \int \mathscr{E}_M/-^{\,\mathrm{op}}$, where the opposite is again taken pointwise.

Unwinding this definition, we find that the objects of $\mathbf{Poly}_M$ are the same polynomial functors as constitute the objects of $\mathbf{Poly}_{\mathscr{E}}$. The morphisms $f : p \to q$ are pairs $(f_1, f^\#)$, where $f_1 : B \to C$ is a map in $\mathscr{E}$ and $f^\#$ is a family of morphisms $q[f_1(x)] \twoheadrightarrow p[x]$ in $\mathscr{K}\ell(M)$, making the following diagram commute:

$$\begin{array}{ccccc}
\sum_{x:B} Mp[x] & \xleftarrow{f^\#} & \sum_{b:B} q[f_1(x)] & \longrightarrow & \sum_{y:C} q[y] \\
{\scriptstyle \eta_B * p}\downarrow & & \downarrow & \lrcorner & \downarrow{\scriptstyle q} \\
B & = = = & B & \xrightarrow{f_1} & C
\end{array}$$

Remark 2.15. Note that the tensor $\otimes$ extends to $\mathbf{Poly}_M$: on objects, it is defined identically to the tensor on $\mathbf{Poly}_{\mathscr{E}}$. On morphisms $f := (f_1, f^\#) : p \to q$ and $g := (g_1, g^\#) : p' \to q'$, we define the tensor $f \otimes g$ to have forwards component $f_1 \times g_1$ as before, and the backwards components are defined by $(f \otimes g)^\#_{(x,x')} := q[f_1(x)] \times q'[g_1(x')] \to Mp[x] \times Mp'[x'] \to M\big(p[x] \times p'[x']\big)$, where the second arrow is given by the commutativity of the monad M. On the other hand, we only get an internal hom satisfying the adjunction $\mathbf{Poly}_M(p \otimes q, r) \cong \mathbf{Poly}_M(p, [q, r])$ when the backwards components of morphisms $p \otimes q \to r$ are 'uncorrelated' between p and q.

Remark 2.16. For $\mathbf{Poly}_M$ to behave faithfully like the category $\mathbf{Poly}_{\mathscr{E}}$ of polynomial functors and their morphisms, we should want the substitution functors $\mathscr{E}_M/f : \mathscr{E}_M/C \to \mathscr{E}_M/B$ to have left and right adjoints. Although we do not spell it out here, it is quite straightforward to exhibit the left adjoints. On the other hand, writing f^* as shorthand for $\mathscr{E}_M/f$, we can see that a right adjoint only obtains in restricted circumstances. Denote the putative right adjoint by $\Pi_f : \mathscr{E}_M/B \to \mathscr{E}_M/C$, and for $\iota p : E \twoheadrightarrow B$ suppose that $(\Pi_f E)[y]$ is given by the set of 'partial sections' $\sigma : f^{-1}\{y\} \to TE$ of p over $f^{-1}\{y\}$ as in the commutative diagram:

$$\begin{array}{ccccc}
 & \overset{\sigma}{\swarrow} & f^{-1}\{y\} & \longrightarrow & \{y\} \\
 & & \downarrow & \lrcorner & \downarrow \\
TE & \xrightarrow{\eta_B * p} & B & \xrightarrow{f} & C
\end{array}$$

Then we would need to exhibit a natural isomorphism $\mathscr{E}_M/B(f^*D, E) \cong \mathscr{E}_M/C(D, \Pi_f E)$. But this will only obtain when the 'backwards' components $h^\#_y : D[y] \to M(\Pi_f E)[y]$ are in the image of ι—otherwise, it is not generally possible to pull $f^{-1}\{y\}$ out of M.

Despite these restrictions, we do have enough structure at hand to instantiate $\mathbf{Coalg}^{\mathbb{T}}$ in $\mathbf{Poly}_M$. The only piece remaining is the composition product $\triangleleft$, but for our purposes it suffices to define its action on objects, which is identical to its action on objects in $\mathbf{Poly}_{\mathscr{E}}$[1], and then consider $\triangleleft$-comonoids in $\mathbf{Poly}_M$. The comonoid laws force the structure maps to be deterministic (*i.e.*, in the image of ι), and so $\triangleleft$-comonoids in $\mathbf{Poly}_M$ are just $\triangleleft$-comonoids in $\mathbf{Poly}_{\mathscr{E}}$.

Finally, we note that we can define morphisms $\beta : Sy^S \to [\mathbb{T}y, p]$: these again just correspond to morphisms $\mathbb{T}y \otimes Sy^S \to p$, and the condition that the backwards maps be uncorrelated between $\mathbb{T}y$ and p is satisfied because $\mathbb{T}y$ has a trivial exponent. Unwinding such a β according to the definition of $\mathbf{Poly}_M$ indeed gives precisely a pair (β^o, β^u) of the requisite types; and a comonoid homomorphism $Sy^S \to y^{\mathbb{T}}$ in $\mathbf{Poly}_M$ is precisely a functor $\mathbf{B}\mathbb{T} \to \mathscr{K}\ell(M)$, thereby establishing equivalence between the objects of $\mathbf{Coalg}^{\mathbb{T}}(p)$ established in $\mathbf{Poly}_M$ and the objects of $\mathbf{Coalg}^{\mathbb{T}}_M(p)$. The equivalence between the hom-sets is established by a similar unwinding. All told, in this section, we have sketched the proof of the following theorem:

Theorem 2.17. Constructing $\mathbf{Coalg}^{\mathbb{T}}(p)$ in $\mathbf{Poly}_M$ yields a category equivalent to $\mathbf{Coalg}^{\mathbb{T}}_M(p)$.

2.3 Random dynamical systems and bundle systems

In the analysis of stochastic systems, it is often fruitful to consider two perspectives: on one side, one considers explicitly the evolution of the distribution of the states of the system, by following (for instance) a Markov process, or Fokker-Planck equation. On the other side, one considers the system as if it were a deterministic system, perturbed by noisy inputs, giving rise to the frameworks of stochastic differential equations and associated *random dynamical systems*.

Whereas a (closed) Markov process is typically given by the action of a 'time' monoid on an object in a Kleisli category of a probability monad, a (closed) random dynamical system is given by a *bundle* of closed dynamical systems, where the base system is equipped with a probability measure which it preserves: the idea being that a random dynamical system can be thought of as a 'random' choice of dynamical system on the total space at each moment in time, with the base measure-preserving system being the source of the randomness [6].

This idea corresponds in non-dynamical settings to the notion of *randomness pushback* [7, Def. 11.19], by which a stochastic map $f : A \to \mathscr{P}B$ can be presented as a deterministic map $f^\flat : \Omega \times A \to B$ where (Ω, ω) is a probability space such that, for any $a : A$, pushing ω forward through $f^\flat(\text{-}, a)$ gives the state $f(a)$; that is, ω induces a random choice of map $f^\flat(\omega, \text{-}) : A \to B$. Similarly, under 'nice' conditions, random dynamical systems and Markov processes do coincide, although they have different suitability in applications.

In this section, we sketch how the generalized-coalgebraic structures developed above extend also to random dynamical systems, though with most details deferred to the Appendix. By observing that we can also 'open up' the base system of a random dynamical system, we obtain furthermore a notion of *open bundle system*: a bundle of dynamical systems that is coherently 'open' over polynomials both in the total space and the base space.

Definition 2.18. Suppose $\mathscr{E}$ is a category equipped with a probability monad $\mathscr{P} : \mathscr{E} \to \mathscr{E}$ and a terminal object $1 : \mathscr{E}$. A probability space in $\mathscr{E}$ is an object of the slice $1/\mathscr{K}\ell(\mathscr{P})$ of the Kleisli category of the probability monad under 1.

Remark 2.19. In order to consider polynomials in $\mathscr{E}$, we will later assume again that it is locally Cartesian closed. A simple example of a locally Cartesian closed category equipped with a probability monad is

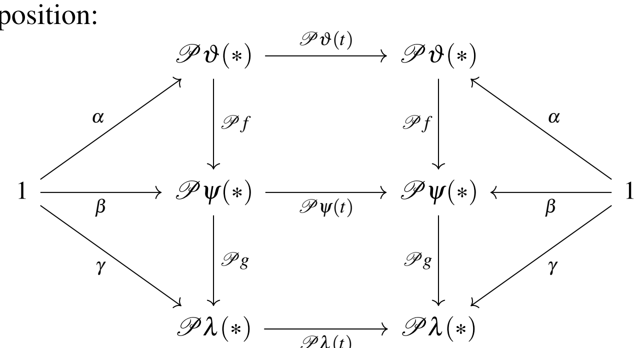
[1] We leave the full exposition of $\triangleleft$ in $\mathbf{Poly}_M$ to the forthcoming extended version of this paper.

the category **Set** equipped with the monad $\mathscr{D}$ taking each set to the set of finitely-supported probability distributions upon it.

Proposition 2.20. There is a forgetful functor $1/\mathscr{K}\ell(\mathscr{P}) \to \mathscr{E}$ taking probability spaces (B,β) to the underlying spaces B and their morphisms $f : (A,\alpha) \to (B,\beta)$ to the underlying maps $f : A \to \mathscr{P}B$. We will write B to refer to the space in $\mathscr{E}$ underlying a probability space (B,β), in the image of this forgetful functor.

Definition 2.21. Let (B,β) be a probability space in $\mathscr{E}$. A closed metric or measure-preserving dynamical system (ϑ,β) on (B,β) with time $\mathbb{T}$ is a closed dynamical system ϑ with state space $B : \mathscr{E}$ such that, for all $t : \mathbb{T}$, $\mathscr{P}\vartheta(t) \circ \beta = \beta$; that is, each $\vartheta(t)$ is a (B,β)-endomorphism in $1/\mathscr{K}\ell(\mathscr{P})$.

Proposition 2.22. Closed measure-preserving dynamical systems in $\mathscr{E}$ with time $\mathbb{T}$ form the objects of a category $\mathbf{Cat}(\mathbf{B}\mathbb{T},\mathscr{E})_{\mathscr{P}}$ whose morphisms $f : (\vartheta,\alpha) \to (\psi,\beta)$ are maps $f : \vartheta(*) \to \psi(*)$ in $\mathscr{E}$ between the state spaces that preserve both flow and measure, as in the following commutative diagram, which also indicates their composition:



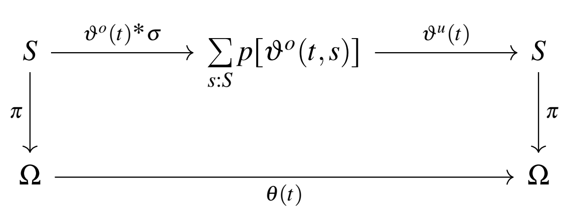
Definition 2.23. Let (ϑ,β) be a closed measure-preserving dynamical system. A closed random dynamical system over (ϑ,β) is an object of the slice category $\mathbf{Cat}(\mathbf{B}\mathbb{T},\mathscr{E})/\vartheta$; it is therefore a bundle of the corresponding functors.

Example 2.24. The solutions $X(t,\omega;x_0) : \mathbb{R}_+ \times \Omega \times M \to M$ to a stochastic differential equation $\mathrm{d}X_t = f(t,X_t)\mathrm{d}t + \sigma(t,X_t)\mathrm{d}W_t$, where $W : \mathbb{R}_+ \times \Omega \to M$ is a Wiener process in M, define a random dynamical system $\mathbb{R}_+ \times \Omega \times M \to M : (t,\omega,x) \mapsto X(t,\omega;x_0)$ over the Wiener base flow $\theta : \mathbb{R}_+ \times \Omega \to \Omega : (t,\omega) \mapsto W(s+t,\omega) - W(t,\omega)$ for any $s : \mathbb{R}_+$.

Definition 2.25. Let (θ,β) be a closed measure-preserving dynamical system in $\mathscr{E}$ with time $\mathbb{T}$, and let $p : \mathbf{Poly}_{\mathscr{E}}$ be a polynomial in $\mathscr{E}$. Write $\Omega := \theta(*)$ for the state space of θ, and let $\pi : S \to \Omega$ be an object (bundle) in $\mathscr{E}/\Omega$. An open random dynamical system over (θ,β) on the interface p with state space $\pi : S \to \Omega$ and time $\mathbb{T}$ consists in a pair of morphisms $\vartheta^o : \mathbb{T} \times S \to p(1)$ and $\vartheta^u : \sum_{t:\mathbb{T}} \sum_{s:S} p[\vartheta^o(t,s)] \to S$, such that, for any global section $\sigma : p(1) \to \sum_{i:p(1)} p[i]$ of p, the maps $\vartheta^\sigma : \mathbb{T} \times S \to S$ defined as

$$\sum_{t:\mathbb{T}} S \xrightarrow{\vartheta^o(-)^*\sigma} \sum_{t:\mathbb{T}} \sum_{s:S} p[\vartheta^o(-,s)] \xrightarrow{\vartheta^u} S$$

form a closed random dynamical system in $\mathbf{Cat}(\mathbf{B}\mathbb{T},\mathscr{E})/\theta$, in the sense that, for all $t : \mathbb{T}$ and sections σ, the following diagram commutes:



Proposition 2.26. Let (θ,β) be a closed measure-preserving dynamical system in $\mathscr{E}$ with time $\mathbb{T}$, and let $p : \mathbf{Poly}_{\mathscr{E}}$ be a polynomial in $\mathscr{E}$. Open random dynamical systems over (θ,β) on the interface p form the objects of a category $\mathbf{RDyn}^{\mathbb{T}}(p,\theta)$. See Definition A.3 in the Appendix for details.

Proposition 2.27. The categories $\mathbf{RDyn}^{\mathbb{T}}(p,\theta)$ collect into a doubly-indexed category of the form $\mathbf{RDyn}^{\mathbb{T}} : \mathbf{Poly}_{\mathscr{E}} \times \mathbf{Cat}(\mathbf{B}\mathbb{T},\mathscr{E})_{\mathscr{P}} \to \mathbf{Cat}$. See Proposition A.4 in the Appendix for details.

By allowing the base systems of open random dynamical systems instead to be arbitrary dynamical systems, and then by opening them up similarly, one obtains notions of *open bundle dynamical system*, and correspondingly doubly-opindexed categories over pairs of polynomials. Representing these categories concisely, as we did for the categories $\mathbf{Coalg}^{\mathbb{T}}_M(p)$, is the subject of on-going work, and so we defer the details to the Appendix, in Definition A.5, and Propositions A.6, A.7, and A.8.

3 Hierarchical systems via generalized Org

In order to exhibit the main example of this paper, we will need to construct, from the opindexed categories of $p\mathscr{P}$-coalgebras introduced above, monoidal categories whose objects represent the interfaces of hierarchical systems and whose morphisms represent the hierarchical systems themselves. Informally put, we will think of a morphism $p \to q$ in such a category as "a q-shaped system with a p-shaped hole". In order to achieve this, we will in turn adopt and generalize the operad **Org** introduced by Spivak [8].

Definition 3.1 (Following Spivak [8, Def. 2.19])**.** We define a (category-enriched, symmetric, coloured) operad, $\mathbf{Org}^{\mathbb{T}}_M$. Its objects are polynomials, and for any tuple of polynomials $(p_1,\dots,p_k;p')$ of at least length 2, the hom category $\mathbf{Org}^{\mathbb{T}}_M(p_1,\dots,p_k;p')$ is given by $\mathbf{Coalg}^{\mathbb{T}}_M([p_1\otimes\cdots\otimes p_k,p'])$. Note that $\{y \to [\mathbb{T}y,[p,p]]\} \cong \{\mathbb{T}y \to [p,p]\}$. On any given interface p, the identity coalgebra is therefore given by the morphism $\mathbb{T}y \to [p,p]$ that constantly outputs id_p and has trivial backwards component. To define composition, we use the canonical maps $[p,q]\otimes[q,r] \to [p,r]$ and $[p,q]\otimes[p',q'] \to [p\otimes p',q\otimes q']$, the pseudofunctoriality of $\mathbf{Coalg}^{\mathbb{T}}_M(-)$, and the laxators $\mathbf{Coalg}^{\mathbb{T}}_M(p)\times\mathbf{Coalg}^{\mathbb{T}}_M(q) \to \mathbf{Coalg}^{\mathbb{T}}_M(p\otimes q)$; since each of these components is associative and unital, the composition is well-defined.

Remark 3.2. Spivak's original definition of **Org** corresponds to the case where $M = \mathsf{id}_{\mathscr{E}}$ and $\mathbb{T} = \mathbb{N}$.

For our present purposes, all that is required is to obtain from **Org** a (monoidal) (bi)category[2]. We therefore restrict $\mathbf{Org}^{\mathbb{T}}_M$ to a bicategory **Hier** whose objects are again polynomials and whose hom-categories from p to q are given by $\mathbf{Org}^{\mathbb{T}}_M(p,q)$; it inherits a monoidal structure from the monoidal category associated to the symmetric operad $\mathbf{Org}^{\mathbb{T}}_M$. We will write $\mathbf{Hier}|_{\mathscr{E}}$ to denote the restriction of **Hier** to the linear polynomials Ay.

To bring things a little down to earth, first consider a general system $\beta : p \to q$ in **Hier**. Recall that $[p,q] = \sum_{f:p\to q} y^{\sum_{i:p(1)} q[f_1(i)]}$. β is therefore given by a choice of state space X along with a pair of maps $\beta^o : \mathbb{T}\times X \to \mathbf{Poly}_M(p,q)$ and $\beta^u : \sum_{t:\mathbb{T}}\sum_{x:X}\sum_{i:p(1)} q[\beta^o(t,s)_1(i)] \to MX$.

To make this a little more comprehensible again, suppose $p = Ay^S$ and $q = By^T$. Then $\mathbf{Poly}_M(p,q) = \mathscr{E}(A,B)\times\mathscr{E}(A\times T,S)$, and so by the universal property of the product, β^o is equivalently given by a pair of maps: a 'forwards' output map $\beta^o_1 : \mathbb{T}\times X\times A \to B$ and a 'backwards' output map $\beta^o_2 : \mathbb{T}\times X\times A\times T \to S$; if this reminds you of a category of lenses, then this is no surprise: the subcategory of $\mathbf{Poly}_{\mathscr{E}}$ on the monomials Ay^S is indeed the category of bimorphic lenses in $\mathscr{E}$. Finally, the update map simplifies to $\beta^u : \mathbb{T}\times X\times A\times T \to MX$, which updates the state given 'forwards' inputs in A and 'backwards' inputs in T. We might denote the subcategory of **Hier** on such linear polynomials as **HiBi**, to indicate 'hierarchical bidirectional' systems.

[2] and in fact, we won't even really need to make use of the monoidal or bicategorical structures here!

Taking one further step down the ladder of complexity, we briefly consider systems $\beta : Ay \to By$ in $\mathbf{Hier}|_{\mathscr{E}}$: these are just hierarchical bidirectional systems where $S = T = 1$. Therefore, in this case, the backwards output map becomes trivial, leaving only a forwards output map $\beta^o : \mathbb{T} \times X \times A \to B$ and an update map taking inputs in A, $\beta^u : \mathbb{T} \times X \times A \to MX$. By filling in the A-inputs, we get a system with B-outputs, corresponding to the informal intuition with which we opened this section: we have a B-shaped system with an A-shaped hole. Composition of these systems corresponds to placing systems in parallel using $\otimes$ and plugging interfaces into holes of the matching shape.

We end this section by briefly sketching the canonical $\otimes$-comonoid structure on $\mathbf{Hier}|_{\mathscr{E}}$, making $\mathbf{Hier}|_{\mathscr{E}}$ into a 'semi-Markov' [7] or 'copy-discard' [9] category. Note that, if a system has the trivial state space 1, then (i) tensoring with it is a no-op, and (ii) it has a trivial update map (assuming that $M1 \cong 1$[3]). Thus, for each object Ay, we obtain a discarding system $\top_A : Ay \to 1y$ by taking the trivial state space, trivial update map, and trivial output map. The copying system $\curlyvee_A : Ay \to (A \times A)y$ again has trivial state space and update map, but now the output map $\curlyvee_A^o : \mathbb{T} \times A \to A \times A$ is given by the constant copying map $(t, a) \mapsto (a, a)$. It is then straightforward to check the comonoid laws.

4 Dynamical Bayesian inversion

One consequence of $\mathbf{Hier}|_{\mathscr{E}}$ being a copy-discard category is that we can instantiate an abstract form of Bayes' rule there, giving rise to a notion of when one $p\mathscr{P}$-coalgebraic system can be seen to be 'predicting' or 'inverting' another. In general, Bayes' rule is expressed as an equality between morphisms, but this is too strong for dynamical systems, which 'black-box' their state spaces: that is to say, we should consider two morphisms (systems) 'equal' when they are observationally equivalent—or, more precisely, when they are related by a (quasi-)bisimulation.

Definition 4.1. We define a family of relations $\sim$ that we collectively call *quasi-bisimilarity*. Given systems $\vartheta := (X, \vartheta^o, \vartheta^u)$ and $\psi := (Y, \psi^o, \psi^u)$ in $\mathbf{Coalg}^{\mathbb{T}}_{\mathscr{P}}(p)$ and a section σ of p, we first define the *trace*[4] or *trajectory* of ϑ given σ as the morphism

$$\mathsf{tr}(\theta, \sigma) := \mathbb{T} \times X \xrightarrow{\vartheta^o(-)^*\sigma} \sum_{t:\mathbb{T}} \sum_{x:X} p[\vartheta^o(t,x)] \xrightarrow{(\vartheta^u)^{\rhd \mathbb{T}}} \mathbb{T} \times \mathscr{P}X \xrightarrow{\mathscr{P}\vartheta^o} \mathscr{P}p(1)\,.$$

Supposing $\alpha : 1 \to \mathscr{P}X$ and $\beta : 1 \to \mathscr{P}Y$ to be corresponding initial states, we define $\vartheta \overset{\alpha,\beta}{\sim} \psi$ as the relation

$$\vartheta \overset{\alpha,\beta}{\sim} \psi \iff \forall \sigma : \Gamma(p).\, \forall t : \mathbb{T}.\, \mathsf{tr}(\vartheta, \sigma)(t) \bullet \alpha = \mathsf{tr}(\psi, \sigma)(t) \bullet \beta\,,$$

where we write $g \bullet f$ to indicate Kleisli composition $g \bullet f = \mu \circ \mathscr{P}g \circ f$ (where μ is the multiplication of the monad $\mathscr{P}$). We write $\vartheta \overset{\exists,\exists}{\sim} \psi$ when there exists some α, β such that $\vartheta \overset{\alpha,\beta}{\sim} \psi$, and likewise for $\vartheta \overset{\forall,\forall}{\sim} \psi$, $\vartheta \overset{\forall,\exists}{\sim} \psi$, and $\vartheta \overset{\exists,\forall}{\sim} \psi$.

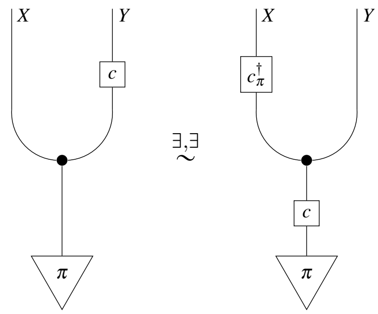
In light of this definition, we can define an appropriate notion of Bayesian inversion for $\mathbf{Hier}|_{\mathscr{E}}$:

---

[3]This condition is satisfied when M is a probability monad like the finite-support distribution monad, for instance.

[4]Note that this is in analogy with the *coalgebraic trace*, not the trace of *traced monoidal categories*.

Definition 4.2. We say that a system $c : Xy \to Yy$ in $\mathbf{Hier}|_{\mathscr{E}}$ *admits Bayesian inversion* with respect to $\pi : y \to Xy$, if there exists a system $c^\dagger_\pi : Yy \to Xy$ satisfying the equation [9, eq. 5]:



We call $c^\dagger_\pi$ the *Bayesian inversion* of c with respect to π, and call the defining relation the *dynamical Bayes' rule*.

5 The Laplace doctrine of predictive processing

In real-world systems, however, even such quasi-bisimulation is too strong. In the setting of computational neuroscience, it is proposed [10, 11] that certain neural circuits implement *approximate* Bayesian inference by optimizing certain statistical games [12]. A statistical game consists of a Bayesian lens—a pair of a 'forwards' stochastic channel $A \to \mathscr{P}B$ and a 'backwards' inversion $\mathscr{P}A \times B \to \mathscr{P}A$—equipped with a loss function to evaluate the systems predictive performance. Embodied predictive systems such as brains then realize these games as dynamical systems. Here we sketch this functorial semantics, using a category of 'hierarchical bidirectional Stat-systems', following [12, 13].

We noted above that the category **HiBi** resembles a category of lenses, but it does not sufficiently resemble the category of *Bayesian* lenses: notice that the backwards maps of the latter have codomains of the form $\mathscr{P}A \times T \to \mathscr{P}S$ rather than $A \times T \to S$. For this reason, **HiBi** makes for an inadequate semantic category for predictive processing. However, all is not lost, for we can define a modification of **HiBi** by analogy to the definition of Bayesian lenses as Grothendieck lenses for the indexed category Stat of state-dependent maps [13].

Definition 5.1. Denote by $\mathbf{HiBi}_{\mathscr{P}}$ the following (semi-)(bi)category. Its objects are pairs of objects in $\mathscr{E}$, and its hom-categories $\mathbf{HiBi}_{\mathscr{P}}((A,S),(B,T))$ are given by $\mathbf{Org}^{\mathbb{T}}_{\mathscr{P}}(\mathscr{P}Ay^S, By^T)$. Composition is given by the following family of composite maps:

$$
\begin{aligned}
&\mathbf{HiBi}_{\mathscr{P}}((A,S),(B,T)) \times \mathbf{HiBi}_{\mathscr{P}}((B,T),(C,U)) \\
&= \mathbf{Org}^{\mathbb{T}}_{\mathscr{P}}(\mathscr{P}Ay^S, By^T) \times \mathbf{Org}^{\mathbb{T}}_{\mathscr{P}}(\mathscr{P}By^T, Cy^U) \\
&= \mathbf{Coalg}^{\mathbb{T}}_{\mathscr{P}}([\mathscr{P}Ay^S, By^T]) \times \mathbf{Coalg}^{\mathbb{T}}_{\mathscr{P}}([\mathscr{P}By^T, Cy^U]) \\
&\to \mathbf{Coalg}^{\mathbb{T}}_{\mathscr{P}}([\mathscr{P}Ay^S, By^T] \otimes [\mathscr{P}By^T, Cy^U]) \\
&\to \mathbf{Coalg}^{\mathbb{T}}_{\mathscr{P}}([\mathscr{P}Ay^S, \mathscr{P}By^T] \otimes [\mathscr{P}By^T, Cy^U]) \\
&\to \mathbf{Coalg}^{\mathbb{T}}_{\mathscr{P}}([\mathscr{P}Ay^S, Cy^U]) \\
&= \mathbf{HiBi}_{\mathscr{P}}((A,S),(C,U))
\end{aligned}
$$

where the fourth line is generated from the monadic unit $\eta_B : B \to \mathscr{P}B$ by $\mathbf{Coalg}([\mathscr{P}y^S, (\eta_B)y^T])$.

Remark 5.2. Note that we say 'semi-'(bi)category: this is because $\mathbf{HiBi}_{\mathscr{P}}$ does not have identities. This is not problematic for our work here; and of course $\mathbf{Org}^{\mathbb{T}}_{\mathscr{P}}$ itself does have identities.

We are now in a position to sketch the 'Laplace doctrine' of dynamical semantics for approximate inference. We first recall the notion of D-Bayesian inference game [12]:

Definition 5.3 (Bayesian inference)**.** Let $D : \mathscr{K}\ell(\mathscr{P})(I,X) \times \mathscr{K}\ell(\mathscr{P})(1,X) \to \mathbb{R}$ be a measure of divergence between states on X. Then a (simple) *D-Bayesian inference* game is a statistical game $(X,X) \to (Y,Y)$ with fitness function $\phi : \mathscr{K}\ell(\mathscr{P})(1,X) \times \mathscr{K}\ell(\mathscr{P})(Y,X) \to \mathbb{R}$ given by

$$\phi(\pi,k) = \mathop{\mathbb{E}}_{y\sim k\bullet c\bullet\pi}\left[D\left(c'_\pi(y), c^\dagger_\pi(y)\right)\right]$$

where (c,c') constitutes the lens part of the game and $c^\dagger_\pi$ is the exact inversion of c with respect to π.

Write D_{KL} for the Kullback-Leibler divergence. Given a D_{KL}-Bayesian inference game (γ,ρ,ϕ) : $(X,X) \to (Y,Y)$ where X and Y are Euclidean spaces and whose forward and backward channels are constrained to output Gaussian distributions, the Laplace doctrine returns a hierarchical bidirectional Stat-system minimizing an upper bound on the divergence between each approximate posterior ρ_π and the 'true' posterior $\gamma^\dagger_\pi$, for any Gaussian state $\pi : \mathscr{P}X$.

Remark 5.4. Note that the statistical properties of the system are not the focus of this paper: this doctrine is merely being used to illustrate the coalgebraic framework.

The Laplace doctrine hinges on the following approximation, whose proof we defer to A.9.

Lemma 5.5 (Laplace approximation)**.** Given a D_{KL}-Bayesian inference game $(\gamma,\rho,\phi) : (X,X) \to (Y,Y)$ with forwards channel $\gamma : X \to \mathscr{P}Y$ constrained to emit Gaussian distributions, write $\mu_\gamma(x) : \mathbb{R}^{|Y|}$ for the mean of $\gamma(x)$ and $\Sigma_\gamma(x) : \mathbb{R}^{|Y|\times|Y|}$ for its covariance matrix, and assume that for all $y : Y$, the eigenvalues of $\Sigma_{\rho_\pi}(y)$ are small.

Then the loss $\phi : \mathscr{K}\ell(\mathscr{P})(1,X) \times \mathscr{K}\ell(\mathscr{P})(Y,X) \to \mathbb{R}$ is approximately bounded from above by

$$\begin{aligned}\phi(\pi,k) &= \mathop{\mathbb{E}}_{y\sim k\bullet\gamma\bullet\pi}\left[D\left(\rho_\pi(y), \gamma^\dagger_\pi(y)\right)\right] \\ &\leqslant \mathop{\mathbb{E}}_{y\sim k\bullet\gamma\bullet\pi}\left[D\left(\rho_\pi(y), \gamma^\dagger_\pi(y)\right) - \log p_{\gamma\bullet\pi}(y)\right] \\ &= \mathop{\mathbb{E}}_{y\sim k\bullet\gamma\bullet\pi}\left[\mathscr{F}(y)\right] \approx \mathop{\mathbb{E}}_{y\sim k\bullet\gamma\bullet\pi}\left[\mathscr{F}^L(y)\right]\end{aligned}$$

where $\mathscr{F}$ is called the *free energy* and where $\mathscr{F}^L$ is its *Laplace approximation*,

$$\begin{aligned}\mathscr{F}^L(y) &= E_{(\pi,\gamma)}\left(\mu_{\rho_\pi}(y), y\right) - S_X\left[\rho_\pi(y)\right] \\ &= -\log p_\gamma(y|\mu_{\rho_\pi}(y)) - \log p_\pi(\mu_{\rho_\pi}(y)) - S_X\left[\rho_\pi(y)\right]\end{aligned} \tag{1}$$

where $S_x[\rho_\pi(y)] = \mathbb{E}_{x\sim\rho_\pi(y)}[-\log p_{\rho_\pi}(x|y)]$ is the Shannon entropy of $\rho_\pi(y)$, and $p_\gamma : Y \times X \to [0,1]$, $p_\pi : X \to [0,1]$, and $p_{\rho_\pi} : X \times Y \to [0,1]$ are density functions for γ, π, and ρ_π respectively. The approximation is valid when Σ_{ρ_π} satisfies

$$\Sigma_{\rho_\pi}(y) = \left(\partial^2_x E_{(\pi,\gamma)}\right)\left(\mu_{\rho_\pi}(y), y\right)^{-1}. \tag{2}$$

With this approximation in hand, and given such a statistical game (γ,ρ,ϕ), we will construct a hierarchical bidirectional Stat-system $\mathsf{Laplace}(\gamma,\rho,\phi)$ performing approximate stochastic gradient descent on the loss function, with respect to the statistical parameters of the inversions ρ_π. We will work in discrete time, $\mathbb{T} = \mathbb{N}$, although all of what follows can be done in continuous time, $\mathbb{T} = \mathbb{R}_+$, by replacing the discrete update steps by stochastic differential equations.

Since the entropy $S_X[\rho_\pi(y)]$ depends only on the variance $\Sigma_{\rho_\pi}(y)$, to optimize the mean $\mu_{\rho_\pi}(y)$ it suffices to consider only the energy $E_{(\pi,\gamma)}(\mu_{\rho_\pi}(y), y)$. We have

$$\begin{aligned} E_{(\pi,\gamma)}(x,y) &= -\log p_\gamma(y|x) - \log p_\pi(x) \\ &= -\frac{1}{2}\left\langle \varepsilon_\gamma(y,x), \Sigma_\gamma(x)^{-1}\varepsilon_\gamma(y,x)\right\rangle - \frac{1}{2}\left\langle \varepsilon_\pi(x), {\Sigma_\pi}^{-1}\varepsilon_\pi(x)\right\rangle \\ &\quad + \log\sqrt{(2\pi)^{|Y|}\det\Sigma_\gamma(x)} + \log\sqrt{(2\pi)^{|X|}\det\Sigma_\pi} \end{aligned}$$

and a straightforward computation shows that

$$\partial_x E_{(\pi,\gamma)}(x,y) = -\partial_x \mu_\gamma(x)^T \Sigma_\gamma(x)^{-1}\varepsilon_\gamma(y,x) + {\Sigma_\pi}^{-1}\varepsilon_\pi(x)\,.$$

Let $\eta_\gamma(y,x) := \Sigma_\gamma(x)^{-1}\varepsilon_\gamma(y,x)$ and $\eta_\pi(x) := {\Sigma_\pi}^{-1}\varepsilon_\pi(x)$, so that

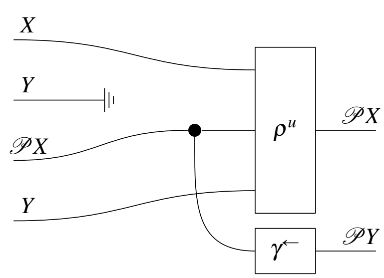
$$\partial_x E_{(\pi,\gamma)}(x,y) = -\partial_x \mu_\gamma(x)^T \eta_\gamma(y,x) + \eta_\pi(x)\,. \tag{3}$$

Note that $E_{(\pi,\gamma)}$ defines a function $X \times Y \to \mathbb{R}$. We will use the domain $X \times Y$ of this function as the state space of our system. To avoid ambiguity, we will write $\overrightarrow{X}$ to indicate the space X when it is used as an input in the 'forwards' direction, and $\overleftarrow{Y}$ to indicate the space Y when it is used as an input in the 'backwards' direction.

Our system $\mathsf{Laplace}(\gamma, \rho, \phi)$ will therefore have the type

$$\begin{aligned} (X \times Y, \quad & \beta_1^o : X \times Y \times \mathscr{P}\overrightarrow{X} \to \overrightarrow{Y}, \\ & \beta_2^o : X \times Y \times \mathscr{P}\overrightarrow{X} \times \overleftarrow{Y} \to \overleftarrow{X}, \\ & \beta^u : X \times Y \times \mathscr{P}\overrightarrow{X} \times \overleftarrow{Y} \to \mathscr{P}(X \times Y)). \end{aligned}$$

We define β_1^o to be the projection of the second factor Y of the state space onto Y, and β_2^o to be the projection of the first factor X onto X. The update map $\beta^u : X \times Y \times \mathscr{P}\overrightarrow{X} \times \overleftarrow{Y} \to \mathscr{P}(X \times Y)$ is then given by composing the commutativity (or 'double strength') of the monad $\mathscr{P}$, $\mathsf{dst} : \mathscr{P}X \times \mathscr{P}Y \to \mathscr{P}(X \times Y)$, after the following map (represented as a string diagram in $\mathscr{E}$):



where $(-)^{\leftarrow} := \mu^{\mathscr{P}} \circ \mathscr{P}(-)$ denotes Kleisli extension (for $\mu^{\mathscr{P}}$ the multiplication of the monad $\mathscr{P}$), so that $\gamma^{\leftarrow} := \mu_Y^{\mathscr{P}} \circ \mathscr{P}(\gamma) : \mathscr{P}X \to \mathscr{P}Y$.

In turn, the map $\rho^u : X \times \mathscr{P}X \times Y \to \mathscr{P}X$ is defined by

$$\begin{aligned} \rho^u : X \times \mathscr{P}X \times Y &\to \mathbb{R}^{|X|} \times \mathbb{R}^{|X|\times|X|} \hookrightarrow \mathscr{P}X \\ (x, \pi, y) &\mapsto \Big(x - \lambda\, \partial_x E_{(\pi,\gamma)}(x,y), \Sigma^*_{\rho_\pi}(y)\Big) \end{aligned}$$

where the inclusion into $\mathscr{P}X$ picks the Gaussian state with the given statistical parameters, where $\lambda : \mathbb{R}_+$ is some choice of "learning rate", where $\Sigma^*_{\rho_\pi}(y)$ is as above and in Equation (2), and where $\partial_x E_{(\pi,\gamma)}(x,y)$ is as in Equation (3).

Observe that the factor ρ^u performs approximate stochastic gradient descent on the free energy: for a given input $y : Y$, the mean trajectory of the system follows the update law $\mu_\rho \mapsto \mu_\rho - \lambda \partial_{\mu_\rho} E_{(\pi,\gamma)}(\mu_\rho, y)$, and, when $\Sigma_\rho(y) = \Sigma^*_\rho(y)$, we have $\partial_{\mu_\rho} E_{(\pi,\gamma)}(\mu_\rho, y) \approx \partial_{\mu_\rho} \mathscr{F}(y)$. Note also that the update map ρ^u depends on a prior, just as the inversion map ρ of the lens (γ, ρ) does.

A full treatment of the Laplace doctrine will appear in a forthcoming sequeal to the author's [12].

6 Conclusions; current and future work

In this work we have sketched a framework for treating open dynamical systems of a general nature as coalgebras for certain polynomial functors or—in the case of systems with side-effects such as randomness—certain generalizations thereof. Although we have attempted to give a wide overview of the applicability of these structures, with a particular focus on the adaptive systems of primary interest to the author, we are aware that we have barely scratched the surface of their use and relationships. Here, we briefly list some avenues of current and future work.

Our current principal focus is on exploring the connections between these structures and other compositional treatments of dynamical systems. In particular, relating our categories to the respective frameworks of Myers [14], Libkind [15] and Baez and colleagues (*e.g.*, [16]). Evidently, the structures presented here are most closely in line with the approaches explored by Spivak [8, 17], and are particularly interested in generalizing his topos-theoretic perspective: given that the category of discrete-time deterministic systems over a polynomial p forms a topos, we suspect that so too does $\mathbf{Coalg}^{\mathbb{T}}(p)$. We are also seeking the connections between these putative topoi and the topoi of behaviour types [17] as well as with coalgebraic logic [18], particularly in its modal forms. We hope that we can further develop the theory of $\mathbf{Poly}_M$ to support some of these methods, too.

Finally, there are a number of ways in which this framework should be made more elegant. In particular, we hope to cast a number of properties instead as structures, including the comonoid-homomorphism property of our main definition, and the explicit definitions of random and bundle dynamical systems. With particular respect to the latter, we expect there to be an inductive story of nested parameterization, which appears to the author to have an opetopic shape closely connected to the **Para** construction [19].

7 References

[1] F. W. Lawvere and S. H. Schnauel. *Conceptual Mathematics : A First Introduction to Categories*. Cambridge, UK New York: Cambridge University Press, 2009. ISBN: 978-0-511-80419-9. DOI: 10.1017/CBO9780511804199.

[2] David I Spivak and Nelson Niu. *Polynomial Functors: A General Theory of Interaction*. (In press), 2021. URL: https://raw.githubusercontent.com/ToposInstitute/poly/main/Book-Poly.pdf.

[3] David I. Spivak. *A Reference for Categorical Structures on Poly*. 02/01/2022. arXiv: 2202.00534. preprint.

[4] Danel Ahman and Tarmo Uustalu. "Directed Containers as Categories". In: *EPTCS 207, 2016, pp. 89-98* (04/05/2016). DOI: 10.4204/EPTCS.207.5. arXiv: 1604.01187.

[5] David I. Spivak. *Poly: An Abundant Categorical Setting for Mode-Dependent Dynamics*. 05/05/2020. arXiv: 2005.01894. preprint.

[6] Ludwig Arnold. *Random Dynamical Systems*. Springer Berlin Heidelberg, 1998. 612 pp. ISBN: 978-3-540-63758-5. DOI: 10.1007/978-3-662-12878-7. URL: https://doi.org/10.1007/978-3-662-12878-7.

[7] Tobias Fritz. "A Synthetic Approach to Markov Kernels, Conditional Independence and Theorems on Sufficient Statistics". In: *Advances in Mathematics* 370.107239 (08/19/2019). DOI: 10.1016/j.aim.2020.107239. arXiv: 1908.07021v3.

[8] David I. Spivak. "Learners' Languages". In: *Electronic Proceedings in Theoretical Computer Science* 372 (03/01/2021), pp. 14–28. DOI: 10.4204/EPTCS.372.2. arXiv: 2103.01189.

[9] Kenta Cho and Bart Jacobs. "Disintegration and Bayesian Inversion via String Diagrams". In: *Math. Struct. Comp. Sci. 29 (2019) 938-971* (08/29/2017). DOI: 10.1017/S0960129518000488. arXiv: 1709.00322v3.

[10] K. Friston. "A Theory of Cortical Responses". In: *Philos. Trans. R. Soc. Lond., B, Biol. Sci.* 360.1456 (04/2005), pp. 815–836. DOI: 10.1098/rstb.2005.1622. pmid: 15937014.

[11] Christopher L Buckley et al. "The Free Energy Principle for Action and Perception: A Mathematical Review". In: *Journal of Mathematical Psychology* 81 (05/24/2017), pp. 55–79. DOI: 10.1016/j.jmp.2017.09.004. arXiv: 1705.09156v1.

[12] Toby St Clere Smithe. *Compositional Active Inference I: Bayesian Lenses. Statistical Games*. 09/09/2021. arXiv: 2109.04461. preprint.

[13] Toby St Clere Smithe. *Bayesian Updates Compose Optically*. 05/31/2020. arXiv: 2006.01631. preprint.

[14] David Jaz Myers. "Double Categories of Open Dynamical Systems (Extended Abstract)". In: *Electronic Proceedings in Theoretical Computer Science* 333 (02/08/2021), pp. 154–167. ISSN: 2075-2180. DOI: 10.4204/EPTCS.333.11. arXiv: 2005.05956. URL: http://arxiv.org/abs/2005.05956 (visited on 11/16/2022).

[15] Sophie Libkind. *An Algebra of Resource Sharing Machines*. 07/28/2020. arXiv: 2007.14442. preprint.

[16] John C. Baez, Kenny Courser, and Christina Vasilakopoulou. "Structured versus Decorated Cospans". In: *Compositionality* 4.3 (09/01/2022). DOI: 10.32408/compositionality-4-3. arXiv: 2101.09363.

[17] Patrick Schultz, David I Spivak, and Christina Vasilakopoulou. "Dynamical Systems and Sheaves". In: *Applied Categorical Structures* 28 (2020), pp. 1–57. DOI: 10.1007/s10485-019-09565-x. arXiv: 1609.08086.

[18] Bart Jacobs. *Introduction to Coalgebra*. Vol. 59. Cambridge University Press, 2017. ISBN: 978-1-107-17789-5. URL: https://www.cambridge.org/gb/academic/subjects/mathematics/logic-categories-and-sets/introduction-coalgebra-towards-mathematics-states-and-observation.

[19] Matteo Capucci, Bruno Gavranović, and Toby St. Clere Smithe. "Parameterized Categories and Categories by Proxy". In: *Category Theory 2021*. Category Theory. Genoa, 2021. URL: https://www.youtube.com/watch?v=lYs6Bs6JK9Q.

A Extra proofs and structures

Proposition A.1. $\mathbf{Coalg}^{\mathbb{T}}(p)$ extends to a polynomially-indexed category, $\mathbf{Coalg}^{\mathbb{T}} : \mathbf{Poly}_{\mathscr{E}} \to \mathbf{Cat}$. Suppose $\varphi : p \to q$ is a morphism of polynomials. We define a corresponding functor $\mathbf{Coalg}^{\mathbb{T}}(\varphi) : \mathbf{Coalg}^{\mathbb{T}}(p) \to \mathbf{Coalg}^{\mathbb{T}}(q)$ as follows. Suppose $(X, \vartheta^o, \vartheta^u) : \mathbf{Coalg}^{\mathbb{T}}(p)$ is an object (dynamical system) in $\mathbf{Coalg}^{\mathbb{T}}(p)$. Then $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(X, \vartheta^o, \vartheta^u)$ is defined as the triple $(X, \varphi_1 \circ \vartheta^o, \vartheta^u \circ \vartheta^{o*}\varphi^\#) : \mathbf{Coalg}^{\mathbb{T}}(q)$, where the two maps are explicitly the following composites:

$$\mathbb{T} \times X \xrightarrow{\vartheta^o} p(1) \xrightarrow{\varphi_1} q(1)\,, \qquad \sum_{t:\mathbb{T}} \sum_{x:X} q[\varphi_1 \circ \vartheta^o(t,x)] \xrightarrow{\vartheta^{o*}\varphi^\#} \sum_{t:\mathbb{T}} \sum_{x:X} p[\vartheta^o(t,x)] \xrightarrow{\vartheta^u} X\,.$$

On morphisms, $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(f) : \mathbf{Coalg}^{\mathbb{T}}(\varphi)(X, \vartheta^o, \vartheta^u) \to \mathbf{Coalg}^{\mathbb{T}}(\varphi)(Y, \psi^o, \psi^u)$ is given by the same underlying map $f : X \to Y$ of state spaces.

Proof. We need to check that $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(X, \vartheta^o, \vartheta^u)$ satisfies the flow conditions of Definition 2.1, that $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(f)$ satisfies the naturality condition of Proposition 2.3, and that $\mathbf{Coalg}^{\mathbb{T}}$ is functorial with respect to polynomials. We begin with the flow condition. Given a section $\tau : q(1) \to \sum_{j:q(1)} q[j]$ of q, we require the closures $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)^\tau : \mathbb{T} \times X \to X$ given by

$$\sum_{t:\mathbb{T}} X \xrightarrow{\vartheta^o(-)^*\tau} \sum_{t:\mathbb{T}} \sum_{x:X} q[\varphi_1 \circ \vartheta^o(t,x)] \xrightarrow{\vartheta^{o*}\varphi^\#} \sum_{t:\mathbb{T}} \sum_{x:X} p[\vartheta^o(t,x)] \xrightarrow{\vartheta^u} X$$

to satisfy $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)^\tau(0) = \mathsf{id}_X$ and $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)^\tau(s+t) = \mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)^\tau(s) \circ \mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)^\tau(t)$. Note that the following diagram commutes, by the definition of $\varphi^\#$,

$$\begin{array}{ccccc}
\sum_{i:p(1)} p[i] & \xleftarrow{\varphi^\#} & \sum_{i:p(1)} q[\varphi_1(i)] & \xleftarrow{\varphi_1^*\tau} & p(1) \\
\Big\downarrow p & & \Big\downarrow \varphi_1^* q & \diagup\!\!\diagup & \\
p(1) & = & p(1) & &
\end{array}$$

so that $\varphi^\# \circ \varphi_1^*\tau$ is a section of p. Therefore, letting $\sigma := \varphi^\# \circ \varphi_1^*\tau$, for $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)^\tau$ to satisfy the flow condition for τ reduces to ϑ^σ satisfying the flow condition for σ. But this is given *ex hypothesi* by Definition 2.1, for any such section σ, so $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)^\tau$ satisfies the flow condition for τ. And since τ was any section, we see that $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(\vartheta)$ satisfies the flow condition generally.

The proof that $\mathbf{Coalg}^{\mathbb{T}}(\varphi)(f)$ satisfies the naturality condition of Proposition 2.3 proceeds similarly. Supposing again that τ is any section of q, we require the following diagram to commute for any time $t : \mathbb{T}$:

$$\begin{array}{ccccccc}
X & \xrightarrow{\vartheta^o(t)^*\varphi_1^*\tau} & \sum_{x:X} q[\varphi_1 \circ \vartheta^o(t,x)] & \xrightarrow{\vartheta^o(t)^*\varphi^\#} & \sum_{x:X} p[\vartheta^o(t,x)] & \xrightarrow{\vartheta^u(t)} & X \\
\Big\downarrow f & & & & & & \Big\downarrow f \\
Y & \xrightarrow{\psi^o(t)^*\varphi_1^*\tau} & \sum_{y:Y} q[\varphi_1 \circ \psi^o(t,x)] & \xrightarrow{\vartheta^o(t)^*\varphi^\#} & \sum_{y:Y} p[\psi^o(t,x)] & \xrightarrow{\psi^u(t)} & Y
\end{array}$$

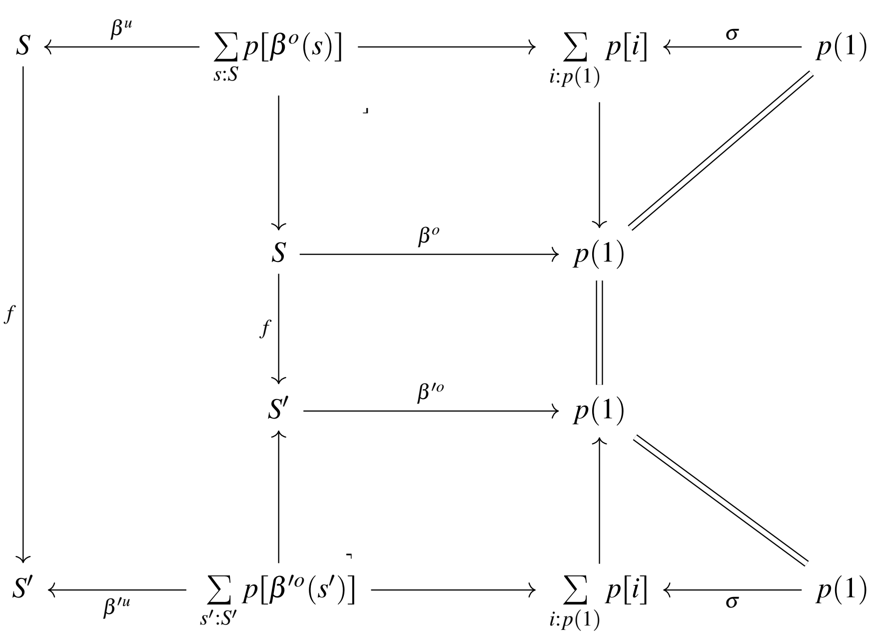
Again letting $\sigma := \varphi^\# \circ \varphi_1^* \tau$, we see that this diagram reduces exactly to the diagram in Proposition 2.3 by the functoriality of pullback, and since f makes that diagram commute, it must also make this diagram commute.

Finally, to show that $\mathbf{Coalg}^{\mathbb{T}}$ is functorial with respect to polynomials amounts to checking that composition and pullback are functorial; but this is a basic result of category theory. □

Proposition A.2. When $\mathbb{T} = \mathbb{N}$, the category $\mathbf{Coalg}^{\mathbb{N}}(p)$ of open dynamical systems over p with time $\mathbb{N}$ is equivalent to the topos p-**Coalg** of p-coalgebras [8].

Proof. p-**Coalg** has as objects pairs (S, β) where $S : \mathscr{E}$ is an object in $\mathscr{E}$, $\beta : S \to p \triangleleft S$ is a morphism of polynomials (interpreting S as the constant copresheaf on the set S), and $\triangleleft$ denotes the composition monoidal product in $\mathbf{Poly}_{\mathscr{E}}$ (*i.e.*, composing the corresponding copresheaves $\mathscr{E} \to \mathscr{E}$). A straightforward computation shows that, interpreted as an object in $\mathscr{E}$, $p \triangleleft S$ corresponds to $\sum_{i:p(1)} S^{p[i]}$. By the universal property of the dependent sum, a morphism $\beta : S \to \sum_{i:p(1)} S^{p[i]}$ therefore corresponds bijectively to a pair of maps $\beta^o : S \to p(1)$ and $\beta^u : \sum_{s:S} p[\beta^o(s)] \to X$. By Proposition 2.6, such a pair is equivalently a discrete-time open dynamical system over p with state space S: that is, the objects of p-**Coalg** are in bijection with those of $\mathbf{Coalg}^{\mathbb{N}}(p)$.

Next, we show that the hom-sets $p\text{-}\mathbf{Coalg}\big((S,\beta),(S',\beta')\big)$ and $\mathbf{Coalg}^{\mathbb{N}}(p)\big((S,\beta^o,\beta^u),(S',\beta'^o,\beta'^u)\big)$ are in bijection. A morphism $f : (S,\beta) \to (S',\beta')$ of p-coalgebras is a morphism $f : S \to S'$ between the state spaces such that $\beta' \circ f = (p \triangleleft f) \circ \beta$. Unpacking this, we find that this means the following diagram in $\mathscr{E}$ must commute for any section σ of p:



Pulling the arbitrary section σ back along the 'output' maps β^o and β'^o means that the following commutes:

$$\begin{array}{ccccc}
S & \xleftarrow{\ \beta^u\ } & \sum_{s:S} p[\beta^o(s)] & \xleftarrow{\ \beta^{o*}\sigma\ } & S \\
 & & \big\downarrow & // & \\
\Big\downarrow{\scriptstyle f} & & S & & \\
 & & \big\downarrow{\scriptstyle f} & & \\
 & & S' & & \\
 & & \big\uparrow & \backslash\backslash & \\
S' & \xleftarrow[\ \beta'^u\ ]{} & \sum_{s':S'} p[\beta'^o(s')] & \xleftarrow[\ \beta'^{o*}\sigma\ ]{} & S'
\end{array}$$

Forgetting the vertical projections out of the pullbacks gives:

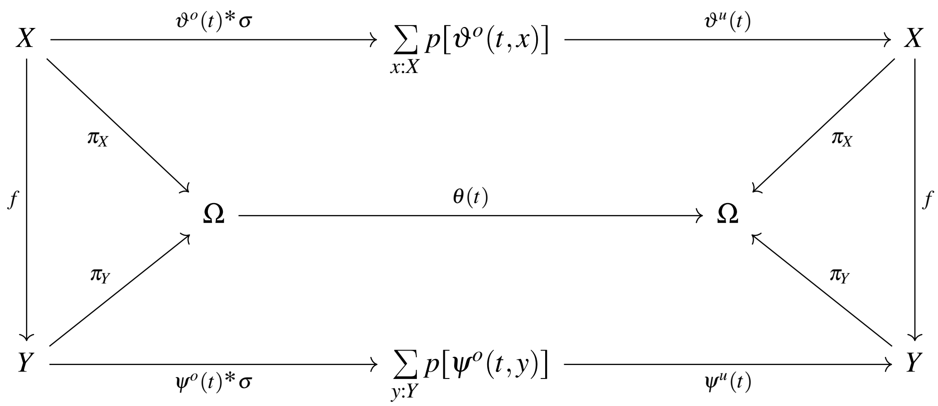
$$\begin{array}{ccccc}
S & \xleftarrow{\ \beta^u\ } & \sum_{s:S} p[\beta^o(s)] & \xleftarrow{\ \beta^{o*}\sigma\ } & S \\
 & & & & \Vert \\
 & & & & S \\
\Big\downarrow{\scriptstyle f} & & & & \big\downarrow{\scriptstyle f} \\
 & & & & S' \\
 & & & & \Vert \\
S' & \xleftarrow[\ \beta'^u\ ]{} & \sum_{s':S'} p[\beta'^o(s')] & \xleftarrow[\ \beta'^{o*}\sigma\ ]{} & S'
\end{array}$$

Finally, by collapsing the identity maps and reflecting the diagram horizontally, we obtain

$$\begin{array}{ccccc}
S & \xrightarrow{\ \beta^{o*}\sigma\ } & \sum_{s:S} p[\beta^o(s)] & \xrightarrow{\ \beta^u\ } & S \\
\big\downarrow{\scriptstyle f} & & & & \big\downarrow{\scriptstyle f} \\
S' & \xrightarrow[\ \beta'^{o*}\sigma\ ]{} & \sum_{s':S'} p[\beta'^o(s')] & \xrightarrow[\ \beta'^u\ ]{} & S'
\end{array}$$

which we recognize from Proposition 2.3 as the defining characteristic of a morphism in $\mathbf{Coalg}^{\mathbb{N}}(p)$. Finally, we note that each of these steps is bijective, and so we have the desired bijection of hom-sets. $\square$

Definition A.3 (Category of open random dynamical systems over p)**.** Writing $\vartheta := (\pi_X, \vartheta^o, \vartheta^u)$ and $\psi := (\pi_Y, \psi^o, \psi^u)$, a morphism $f : \vartheta \to \psi$ is a map $f : X \to Y$ in $\mathscr{E}$ making the following diagram commute for all times $t : \mathbb{T}$ and sections σ of p:



Identities are given by the identity maps on state-spaces. Composition is given by pasting of diagrams.

Proposition A.4 (Opindexed category of open random dynamical systems over polynomials)**.** By the universal property of the product $\times$ in **Cat**, it suffices to define the actions of $\mathbf{RDyn}^{\mathbb{T}}$ separately on morphisms of polynomials and on morphisms of closed measure-preserving systems.

Suppose therefore that $\varphi : p \to q$ is a morphism of polynomials. Then, for each measure-preserving system $(\theta, \beta) : \mathbf{Cat}(\mathbf{B}\mathbb{T}, \mathscr{E})_{\mathscr{P}}$, we define the functor $\mathbf{RDyn}^{\mathbb{T}}(\varphi, \theta) : \mathbf{RDyn}^{\mathbb{T}}(p, \theta) \to \mathbf{RDyn}^{\mathbb{T}}(q, \theta)$ as follows. Let $\vartheta := (\pi_X : X \to \Omega, \vartheta^o, \vartheta^u) : \mathbf{RDyn}^{\mathbb{T}}(p, \theta)$ be an object (open random dynamical system) in $\mathbf{RDyn}^{\mathbb{T}}(p, \theta)$. Then $\mathbf{RDyn}^{\mathbb{T}}(\varphi, \theta)(\vartheta)$ is defined as the triple $(\pi_X, \varphi_1 \circ \vartheta^o, \vartheta^u \circ \varphi^{o*}\varphi^{\#}) : \mathbf{RDyn}^{\mathbb{T}}(q, \theta)$, where the two maps are explicitly the following composites:

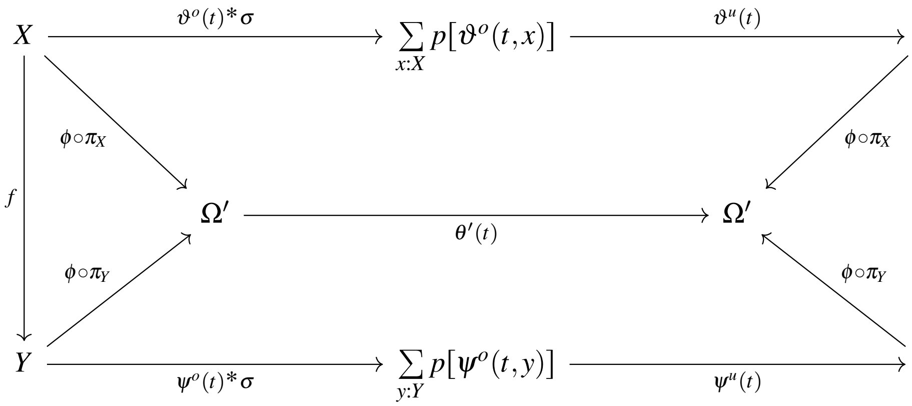
$$\mathbb{T} \times X \xrightarrow{\vartheta^o} p(1) \xrightarrow{\varphi_1} q(1)\,, \qquad \sum_{t:\mathbb{T}} \sum_{x:X} q[\varphi_1 \circ \vartheta^o(t,x)] \xrightarrow{\vartheta^{o*}\varphi^{\#}} \sum_{t:\mathbb{T}} \sum_{x:X} p[\vartheta^o(t,x)] \xrightarrow{\vartheta^u} X\,.$$

On morphisms $f : (\pi_X : X \to \Omega, \vartheta^o, \vartheta^u) \to (\pi_Y : Y \to \Omega, \psi^o, \psi^u)$, the image

$$\mathbf{RDyn}^{\mathbb{T}}(\varphi, \theta)(f) : \mathbf{RDyn}^{\mathbb{T}}(\varphi, \theta)(\pi_X, \vartheta^o, \vartheta^u) \to \mathbf{RDyn}^{\mathbb{T}}(\varphi, \theta)(\pi_Y, \psi^o, \psi^u)$$

is given by the same underlying map $f : X \to Y$ of state spaces.

Next, suppose that $\phi : (\theta, \beta) \to (\theta', \beta')$ is a morphism of closed measure-preserving dynamical systems, and let $\Omega' := \theta'(*)$ be the state space of the system θ'. By Proposition 2.22, the morphism ϕ corresponds to a map $\phi : \Omega \to \Omega'$ on the state spaces that preserves both flow and measure. Therefore, for each polynomial $p : \mathbf{Poly}_{\mathscr{E}}$, we define the functor $\mathbf{RDyn}^{\mathbb{T}}(p, \phi) : \mathbf{RDyn}^{\mathbb{T}}(p, \theta) \to \mathbf{RDyn}^{\mathbb{T}}(p, \theta')$ by post-composition. That is, suppose given open random dynamical systems and morphisms over (p, θ) as in the diagram of Proposition 2.26. Then $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)$ returns the following diagram:



That is, $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)(\vartheta) := (\phi\circ\pi_X, \vartheta^o, \vartheta^u)$ and $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)(f)$ is given by the same underlying map $f : X \to Y$ on state spaces.

Proof. We need to check: the naturality condition of Definition 2.25 for both $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(\vartheta)$ and $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)(\vartheta)$; functoriality of $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)$ and $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)$; and (pseudo)functoriality of $\mathbf{RDyn}^{\mathbb{T}}$ with respect to both morphisms of polynomials and of closed measure-preserving systems.

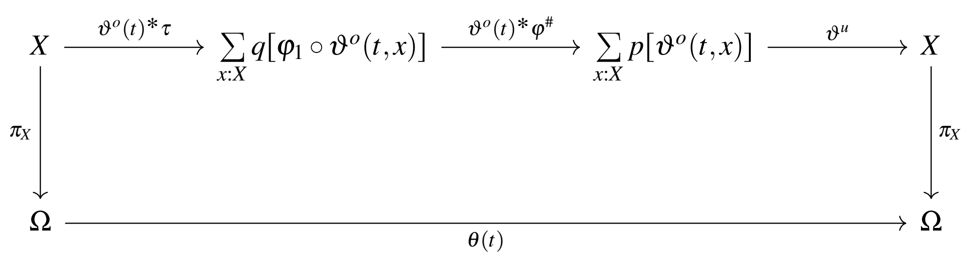
We begin by checking that the conditions of Definition 2.25 are satisfied by the objects

$$\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(\pi_X,\vartheta^o,\vartheta^u) : \mathbf{RDyn}^{\mathbb{T}}(q,\theta)$$

and morphisms

$$\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(f) : \mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(\pi_X,\vartheta^o,\vartheta^u) \to \mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(\pi_Y,\psi^o,\psi^u)$$

in the image of $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)$. Given a section $\tau : q(1) \to \sum_{j:q(1)} q[j]$ of q, we need to check that the closure $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(\vartheta)^{\tau}$ forms a closed random dynamical system in $\mathbf{Cat}(\mathbf{B}\mathbb{T},\mathscr{E})/\theta$. That is to say, for all $t:\mathbb{T}$ and sections τ, we need to check that the following naturality square commutes:



As before, we find that $\varphi^\#\circ\varphi_1^*\tau$ is a section of p, so that commutativity of the diagram above reduces to commutativity of the diagram in Definition 2.25. Similarly, given a morphism $f : (\pi_X,\vartheta^o,\vartheta^u) \to (\pi_Y,\psi^o,\psi^u)$, we need to check that the diagram in Proposition 2.26 induced for $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(f)$ commutes for all times $t:\mathbb{T}$ and sections τ of q. But given such a section τ, the diagram for $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)(f)$ reduces to that for f and the section $\varphi^\#\circ\varphi_1^*\tau$ of p, which commutes *ex hypothesi*; and functoriality of $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)$ follows immediately.

Next, we check that the conditions of Definition 2.25 are satisfied in the image of $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)$. It is clear by the definition of the action of $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)$ that the condition that the diagram in Proposition A.1 commutes is satisfied, from which it follows by pasting that $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)$ is functorial. We therefore just have to check the induced diagram in Definition 2.25 commutes. Consider the following diagram:

$$
\begin{array}{ccccc}
X & \xrightarrow{\vartheta^o(t)^*\sigma} & \sum_{x:X} p[\vartheta^o(t,x)] & \xrightarrow{\vartheta^u(t)} & X \\
{\scriptstyle \pi_X}\downarrow & & & & \downarrow{\scriptstyle \pi_X} \\
\Omega & & \xrightarrow{\theta(t)} & & \Omega \\
{\scriptstyle \phi}\downarrow & & & & \downarrow{\scriptstyle \phi} \\
\Omega' & & \xrightarrow{\theta'(t)} & & \Omega'
\end{array}
$$

The top square commutes *ex hypothesi*, the bottom square commutes by the definition of morphism of closed measure-preserving dynamical systems (Proposition 2.22), and the outer square is the induced diagram we need to check, which therefore commutes by the pasting of commuting squares.

Finally, we check that $\mathbf{RDyn}^{\mathbb{T}}$ is functorial with respect to morphisms of polynomials and morphisms of closed measure-preserving dynamical systems. These reduce to checking that pullback and composition are functorial, which we again leave to the dedicated reader. □

Definition A.5 (Open bundle dynamical system)**.** Let $p, b : \mathbf{Poly}_{\mathscr{E}}$ be polynomials in $\mathscr{E}$, and let $\theta := (\theta(*), \theta^o, \theta^u) : \mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(b)$ be an open dynamical system over b. An open bundle dynamical system over (p, b, θ) is a pair $(\pi_{\vartheta\theta}, \vartheta)$ where $\vartheta := (\vartheta(*), \vartheta^o, \vartheta^u) : \mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(p)$ is an open dynamical system over p and $\pi_{\vartheta\theta} : \vartheta(*) \to \theta(*)$ is a bundle in $\mathscr{E}$, such that, for all time $t : \mathbb{T}$ and sections σ of p and ς of b, the following diagrams commute, thereby inducing a bundle of closed dynamical systems $\pi^{\sigma\varsigma}_{\vartheta\theta} : \vartheta^{\sigma} \to \theta^{\varsigma}$ in $\mathbf{Cat}(\mathbf{B}\mathbb{T}, \mathscr{E})$:

$$
\begin{array}{ccccc}
\vartheta(*) & \xrightarrow{\vartheta^o(t)^*\sigma} & \sum_{w:\vartheta(*)} p[\vartheta^o(t,w)] & \xrightarrow{\vartheta^u(t)} & \vartheta(*) \\
{\scriptstyle \pi_{\vartheta\theta}}\downarrow & & & & \downarrow{\scriptstyle \pi_{\vartheta\theta}} \\
\theta(*) & \xrightarrow{\theta^o(t)^*\varsigma} & \sum_{x:\theta(*)} b[\theta^o(t,x)] & \xrightarrow{\theta^u(t)} & \theta(*)
\end{array}
$$

Proposition A.6 (Category of open bundle dynamical systems over (p,b))**.** Let $p, b : \mathbf{Poly}_{\mathscr{E}}$ be polynomials in $\mathscr{E}$, and let $\theta := (\theta(*), \theta^o, \theta^u) : \mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(b)$ be an open dynamical system over b. Open bundle dynamical systems over (p, b, θ) form the objects of a category $\mathbf{BunDyn}^{\mathbb{T}}(p, b, \theta)$. Morphisms $f : (\pi_{\vartheta\theta}, \vartheta) \to (\pi_{\rho\theta}, \rho)$ are maps $f : \vartheta(*) \to \rho(*)$ in $\mathscr{E}$ making the following diagram commute for all times $t : \mathbb{T}$ and sections σ of p and ς of b:

$$\begin{tikzcd}
\vartheta(*) \arrow[rr, "\vartheta^o(t)^*\sigma"] \arrow[dd, "f"'] \arrow[dr, "\pi_{\vartheta\theta}"] & & \sum\limits_{w:\vartheta(*)} p[\vartheta^o(t,w)] \arrow[rr, "\vartheta^u(t)"] & & \vartheta(*) \arrow[dd, "f"] \arrow[dl, "\pi_{\vartheta\theta}"'] \\
& \theta(*) \arrow[r, "\theta^o(t)^*\varsigma"] & \sum\limits_{x:\theta(*)} b[\theta^o(t,x)] \arrow[r, "\theta^u(t)"] & \theta(*) & \\
\rho(*) \arrow[rr, "\rho^o(t)^*\sigma"'] \arrow[ur, "\pi_{\rho\theta}"] & & \sum\limits_{y:\rho(*)} p[\rho^o(t,y)] \arrow[rr, "\rho^u(t)"'] & & \rho(*) \arrow[ul, "\pi_{\rho\theta}"']
\end{tikzcd}$$

That is, f is a map on the state spaces that induces a morphism $(\pi_{\vartheta\theta}, \vartheta^{\sigma}) \to (\pi_{\rho\theta}, \rho^{\sigma})$ in $\mathbf{Cat}(\mathbf{B}\mathbb{T}, \mathscr{E})/\theta^{\varsigma}$ of bundles of the closures. Identity morphisms are the corresponding identity maps, and composition is by pasting.

Proposition A.7 (Opindexed category of open bundle dynamical systems)**.** Varying the polynomials p in $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta)$ induces an opindexed category $\mathbf{BunDyn}^{\mathbb{T}}(-,b,\theta) : \mathbf{Poly}_{\mathscr{E}} \to \mathbf{Cat}$. On polynomials p, it returns the categories $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta)$ of Proposition A.6. On morphisms $\varphi : p \to q$ of polynomials, define the functors $\mathbf{BunDyn}^{\mathbb{T}}(\varphi,b,\theta) : \mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta) \to \mathbf{BunDyn}^{\mathbb{T}}(q,b,\theta)$ as in Proposition 2.27. That is, suppose $(\pi_{\vartheta\theta}, \vartheta) : \mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta)$ is object (open bundle dynamical system) in $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta)$, where $\vartheta := (\vartheta(*), \vartheta^o, \vartheta^u)$. Then its image $\mathbf{BunDyn}^{\mathbb{T}}(\varphi,b,\theta)(\pi_{\vartheta\theta}, \vartheta)$ is defined as the pair $(\pi_{\vartheta\theta}, \varphi\vartheta)$, where $\varphi\vartheta := (\vartheta(*), \phi_1 \circ \vartheta^o, \vartheta^u \circ \vartheta^{o*}\varphi^{\#})$. On morphisms $f : (\pi_{\vartheta\theta}, \vartheta) \to (\pi_{\rho\theta}, \rho)$, $\mathbf{BunDyn}^{\mathbb{T}}(\varphi,b,\theta)(f)$ is again given by the same underlying map $f : \vartheta(*) \to \rho(*)$ of state spaces.

Proof. The proof amounts to the proof for Proposition 2.27 that $\mathbf{RDyn}^{\mathbb{T}}(\varphi,\theta)$ constitutes an indexed category, except that the closed base dynamical system θ of that Proposition is here replaced, for any section ς of b, by the closure θ^{ς} by ς of the open dynamical system $\theta : \mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(b)$ of the present Proposition. The proof goes through accordingly, since the relevant diagrams are guaranteed to commute for any such ς by the conditions in Definition A.5 and Proposition A.6. □

Proposition A.8 (Doubly-opindexed category of open bundle dynamical systems)**.** Letting the base system θ also vary induces a doubly-opindexed category $\mathbf{BunDyn}^{\mathbb{T}}(-,b,=) : \mathbf{Poly}_{\mathscr{E}} \times \mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(b) \to \mathbf{Cat}$. Given a polynomial $p : \mathbf{Poly}_{\mathscr{E}}$ and morphism $\phi : \theta \to \rho$ in $\mathbf{Coalg}^{\mathbb{T}}_{\mathsf{id}}(b)$, the functor $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\phi) : \mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta) \to \mathbf{BunDyn}^{\mathbb{T}}(p,b,\rho)$ is defined by post-composition, as in Proposition 2.27 for the action of $\mathbf{RDyn}^{\mathbb{T}}$ on morphisms of the base systems there. More explicitly, such a morphism ϕ corresponds to a map $\phi : \theta(*) \to \rho(*)$ of state spaces in $\mathscr{E}$. Given an object $(\pi_{\vartheta\theta}, \vartheta)$ of $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta)$, we define $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\phi)(\pi_{\vartheta\theta}, \vartheta) := (\phi \circ \pi_{\vartheta\theta}, \vartheta)$. Given a morphism $f : (\pi_{\vartheta\theta}, \vartheta) \to (\pi_{\rho\theta}, \rho)$ in $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\theta)$, its image $\mathbf{BunDyn}^{\mathbb{T}}(p,b,\phi)(f) : (\phi \circ \pi_{\vartheta\theta}, \vartheta) \to (\phi \circ \pi_{\rho\theta}, \rho)$ is given by the same underlying map $f : \vartheta(*) \to \rho(*)$ of state spaces.

Proof. As for Proposition A.7, the proof here amounts to the proof for Proposition 2.27 that $\mathbf{RDyn}^{\mathbb{T}}(p,\phi)$ constitutes an indexed category, except again the closed systems are replaced by (the appropriate closures of) open ones, and the measure-preserving structure is forgotten. □

Proof A.9 (Proof of the Laplace approximation). First note that the KL divergence is bounded from above by the free energy since $\log p_{\gamma \bullet \pi}(y)$ is always negative.

Next, we can write the density functions as:

$$\log p_\gamma(y|x) = \frac{1}{2}\left\langle \varepsilon_\gamma, \Sigma_\gamma{}^{-1}\varepsilon_\gamma \right\rangle - \log\sqrt{(2\pi)^{|Y|}\det\Sigma_\gamma}$$
$$\log p_{\rho_\pi}(x|y) = \frac{1}{2}\left\langle \varepsilon_{\rho_\pi}, \Sigma_{\rho_\pi}{}^{-1}\varepsilon_{\rho_\pi} \right\rangle - \log\sqrt{(2\pi)^{|X|}\det\Sigma_{\rho_\pi}}$$
$$\log p_\pi(x) = \frac{1}{2}\left\langle \varepsilon_\pi, \Sigma_\pi{}^{-1}\varepsilon_\pi \right\rangle - \log\sqrt{(2\pi)^{|X|}\det\Sigma_\pi}$$

where for clarity we have omitted the dependence of Σ_γ on x and Σ_{ρ_π} on y, and where

$$\varepsilon_\gamma : Y \times X \to Y : (y,x) \mapsto y - \mu_\gamma(x)\,,$$
$$\varepsilon_{\rho_\pi} : X \times Y \to X : (x,y) \mapsto x - \mu_{\rho_\pi}(y)\,,$$
$$\varepsilon_\pi : X \times 1 \to X : (x,*) \mapsto x - \mu_\pi\,.$$

Then, note that we can write the free energy $\mathscr{F}(y)$ as the difference between expected energy and entropy:

$$\begin{aligned}\mathscr{F}(y) &= \mathop{\mathbb{E}}_{x\sim\rho_\pi(y)}\left[\log\frac{p_{\rho_\pi}(x|y)}{p_\gamma(y|x)\cdot p_\pi(x)}\right] \\ &= \mathop{\mathbb{E}}_{x\sim\rho_\pi(y)}\left[-\log p_\gamma(y|x) - \log p_\pi(x)\right] - S_X\left[\rho_\pi(y)\right] \\ &= \mathop{\mathbb{E}}_{x\sim\rho_\pi(y)}\left[E_{(\pi,\gamma)}(x,y)\right] - S_X\left[\rho_\pi(y)\right]\end{aligned}$$

Next, since the eigenvalues of $\Sigma_{\rho_\pi}(y)$ are small for all $y : Y$, we can approximate the expected energy by its second-order Taylor expansion around the mean $\mu_{\rho_\pi}(y)$:

$$\begin{aligned}\mathscr{F}(y) \approx E_{(\pi,\gamma)}(\mu_{\rho_\pi}(y),y) + \frac{1}{2}\Big\langle \varepsilon_{\rho_\pi}\left(\mu_{\rho_\pi}(y),y\right), \left(\partial_x^2 E_{(\pi,\gamma)}\right)\left(\mu_{\rho_\pi}(y),y\right)\cdot\varepsilon_{\rho_\pi}\left(\mu_{\rho_\pi}(y),y\right)\Big\rangle \\ - S_X\big[\rho_\pi(y)\big]\,.\end{aligned}$$

where $\left(\partial_x^2 E_{(\pi,\gamma)}\right)\left(\mu_{\rho_\pi}(y),y\right)$ is the Hessian of $E_{(\pi,\gamma)}$ with respect to x evaluated at $(\mu_{\rho_\pi}(y),y)$.

Note that

$$\Big\langle \varepsilon_{\rho_\pi}\left(\mu_{\rho_\pi}(y),y\right), \left(\partial_x^2 E_{(\pi,\gamma)}\right)\left(\mu_{\rho_\pi}(y),y\right)\cdot\varepsilon_{\rho_\pi}\left(\mu_{\rho_\pi}(y),y\right)\Big\rangle = \operatorname{tr}\Big[\left(\partial_x^2 E_{(\pi,\gamma)}\right)\left(\mu_{\rho_\pi}(y),y\right)\Sigma_{\rho_\pi}(y)\Big]\,, \qquad (4)$$

that the entropy of a Gaussian measure depends only on its covariance,

$$S_X\big[\rho_\pi(y)\big] = \frac{1}{2}\log\det\left(2\pi e\,\Sigma_{\rho_\pi}(y)\right)\,,$$

and that the energy $E_{(\pi,\gamma)}(\mu_{\rho_\pi}(y),y)$ does not depend on $\Sigma_{\rho_\pi}(y)$. We can therefore write down directly the covariance $\Sigma^*_{\rho_\pi}(y)$ minimizing $\mathscr{F}(y)$ as a function of y. We have

$$\partial_{\Sigma_{\rho_\pi}}\mathscr{F}(y) \approx \frac{1}{2}\left(\partial_x^2 E_{(\pi,\gamma)}\right)\left(\mu_{\rho_\pi}(y),y\right) + \frac{1}{2}\Sigma_{\rho_\pi}{}^{-1}\,.$$

Setting $\partial_{\Sigma_{\rho_\pi}}\mathscr{F}(y) = 0$, we find the optimum

$$\Sigma^*_{\rho_\pi}(y) = \left(\partial_x^2 E_{(\pi,\gamma)}\right)\left(\mu_{\rho_\pi}(y),y\right)^{-1}\,.$$

Finally, on substituting $\Sigma^*_{\rho_\pi}(y)$ in equation (4), we obtain the desired expression

$$\mathscr{F}(y) \approx E_{(\pi,\gamma)}\left(\mu_{\rho_\pi}(y),y\right) - S_X\left[\rho_\pi(y)\right] =: \mathscr{F}^L(y)\,.$$

□

Polynomial Functors and Shannon Entropy

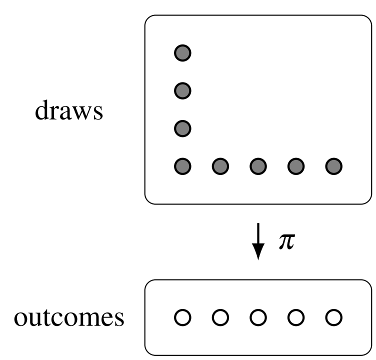
David I. Spivak

Past work shows that one can associate a notion of Shannon entropy to a Dirichlet polynomial, regarded as an empirical distribution. Indeed, entropy can be extracted from any $d \in \text{Dir}$ by a two-step process, where the first step is a rig homomorphism out of Dir, the *set* of Dirichlet polynomials, with rig structure given by standard addition and multiplication. In this short note, we show that this rig homomorphism can be upgraded to a rig *functor*, when we replace the set of Dirichlet polynomials by the *category* of ordinary (Cartesian) polynomials.

In the Cartesian case, the process has three steps. The first step is a rig functor $\mathbf{Poly}^{\mathbf{Cart}} \to \mathbf{Poly}$ sending a polynomial p to $\dot{p}y$, where $\dot{p}$ is the derivative of p. The second is a rig functor $\mathbf{Poly} \to \mathbf{Set} \times \mathbf{Set}^{\text{op}}$, sending a polynomial q to the pair $(q(1), \Gamma(q))$, where $\Gamma(q) = \mathbf{Poly}(q, y)$ can be interpreted as the global sections of q viewed as a bundle, and $q(1)$ as its base. To make this precise we define what appears to be a new distributive monoidal structure on $\mathbf{Set} \times \mathbf{Set}^{\text{op}}$, which can be understood geometrically in terms of rectangles. The last step, as for Dirichlet polynomials, is simply to extract the entropy as a real number from a pair of sets (A, B); it is given by $\log A - \log \sqrt[A]{B}$ and can be thought of as the log aspect ratio of the rectangle.

1 Introduction

In practice, a probability distribution on a set of *outcomes* arises from considering finite samples. A sample consists of a set of observations, or *draws*, each corresponding to one of the outcomes. For example, the following is a sample with five (5) outcomes and eight (8) draws:



(1)

This corresponds to the probability distribution $P = (\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$. But the sample itself can be encoded in the form of a polynomial, namely $p := y^4 + 4y$. Note that $p(1) = 5$ is the number of outcomes and that $\dot{p}(1) = 8$ is the number of draws, where $\dot{p} = 4y^3 + 4$ is the derivative of p. The map π itself is somehow inherent in p: one of its summands has an exponent of 4, whereas its other four summands each have an exponent of 1. Yet one may wonder: is this polynomial encoding really meaningful, or is it just a bizarre packaging of the sample? Our goal in this paper is to show that it is meaningful, at least when it comes to considering the Shannon entropy $H(P)$.

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 331–343, doi:10.4204/EPTCS.380.19

The Shannon entropy of a distribution [Sha48] is a measure of how much information is transmitted when outcomes are selected according to the distribution. For example, if one repeatedly chooses an element of the 8 draws in diagram (1) uniformly at random but only reports the outcome, then the first outcome will show up four-times more often than any other. As we will explain, Shannon's information theory says that this distribution has entropy $H(P) = 2$, i.e. it transmits the same amount of information as if it were a uniform distribution on only 4 outcomes.

In this paper we will give a category-theoretic account of the Shannon entropy of the probability distribution corresponding to a sample encoded as a polynomial p, or more precisely a *polynomial functor* $p \in \mathbf{Poly}$. Polynomial functors are ubiquitous: they show up in type theory [ACH19; AN18], dynamical systems theory [Spi20; SN22], database theory [SW15; Spi21], programming language theory [BD96; AAG03], and higher category theory [TCM19; Sha21].

The category **Poly** of polynomial functors in one variable has an enormous amount of structure. For example, it has at least eight distinct monoidal structures, of which two will be relevant to us. One is the coproduct: given two polynomials p,q, we may add them to form $p+q$. In terms of samples, this operation simply takes the disjoint union of two samples: both the sets of outcomes and the sets of draws. The other is the *Dirichlet product*, denoted $\otimes$. We will give the precise formula for $p \otimes q$ in Section 2.3, but the idea is that it runs the two samples independently: an outcome in $p \otimes q$ is a pair consisting of an outcome from p and an outcome from q, and a draw is also a pair consisting of a draw from p and a draw from q.

These two operations make **Poly** a *distributive monoidal category*, because $\otimes$ distributes over $+$. The goal of this paper is to show that most of the process for taking the Shannon entropy of a sample is fully category-theoretic. Indeed, we will factor the process into three stages, the first two of which are completely categorical, and the last of which extracts a real number that will be the entropy.

The first stage is to define a rig functor $T\colon \mathbf{Poly}^{\mathbf{Cart}} \to \mathbf{Poly}$, which sends a polynomial p to $T(p) := \dot{p}y$, where $\dot{p}$ is the derivative of p. The second stage is to define a rig functor $R\colon \mathbf{Poly} \to \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$, which sends a polynomial q to $R(q) := (q(1), \Gamma(q))$, where $\Gamma(p) = \mathbf{Poly}(p, y)$ can be construed as the set of global sections of p, viewed as a bundle.

The fact that both T and R are rig functors means that each preserves both the coproduct and the $\otimes$-product, a surprising amount of structure. But to say this, we need to define what appears to be a novel symmetric monoidal product $\otimes$ on $\mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$. It is given by

$$(A_1, B_1) \otimes (A_2, B_2) := \left(A_1 A_2\,,\, B_1^{A_2} B_2^{A_1}\right).$$

This monoidal product $\otimes$ distributes over the coproduct, which is given by

$$(A_1, B_1) + (A_2, B_2) := (A_1 + A_2\,,\, B_1 \times B_2)\,,$$

hence making $\mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ a distributive monoidal category, and in particular a *rig category*. We will explain these two rig functors T and R in Section 3. We denote their composite—the result of the first and second stages—by

$$\hbar := (R \circ T)\colon \mathbf{Poly}^{\mathbf{Cart}} \to \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}.$$

It contains the categorical aspect of the entropy in a given sample $p \in \mathbf{Poly}^{\mathbf{Cart}}$.

Before we discuss the third stage, we need a bit of intuition. Namely, we can think of an object $(A,B) \in \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ as encoding a rectangle that has length A and width $\sqrt[A]{B}$. The coproduct of two rectangles is given by adding their lengths and taking the geometric mean of their widths. The $\otimes$-product of two rectangles is given by multiplying both their lengths and their widths. It is in these terms that we can understand the third and final stage, which is simply to take the *log aspect ratio* (the log of the quotient of length divided by width) of a given rectangle:

$$L(A,B) = \log A - \log \sqrt[A]{B}.$$

That is, we will prove that for any polynomial p with an associated probability distribution P, the Shannon entropy $H(P)$ can be computed by first applying the rig-functorial operation to obtain $\hbar(p) \in \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$, and then by extracting the log aspect ratio:

$$H(P) = L(\hbar(p)).$$

We will conclude by returning to our original example, after giving the full composite: the function that takes a polynomial p and returns the entropy of the corresponding empirical distribution is given by

$$L(\hbar(p)) \coloneqq \log \dot{p}(1) - \frac{\log \Gamma(\dot{p}y)}{\dot{p}(1)}$$

Note that $\log \sqrt[A]{B} = \frac{\log B}{A}$.

So consider again the polynomial $p = y^4 + 4y$, depicted in (1). Then we calculate

$$\dot{p}y = 4y^4 + 4y, \quad \dot{p}(1) = 8, \quad \Gamma(\dot{p}y) = 4^4 * 1^4 = 2^8, \quad \text{and} \quad L(\hbar(p)) = \log 8 - \frac{\log 2^8}{8} = 2$$

which agrees with our former calculation: its entropy is $H(P) = L(\hbar(p)) = 2$.

The remainder of this note is divided into two sections: Section 2 gives background on polynomial functors, including the definition of $\mathbf{Poly}^{\mathbf{Cart}} \subseteq \mathbf{Poly}$ as well as the $+$ and $\otimes$ structures. Section 3 gives the main results: explaining the seemingly novel distributive monoidal structure on $\mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$, providing a rig monoidal functor $\hbar\colon \mathbf{Poly}^{\mathbf{Cart}} \to \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$, showing how to extract the entropy via a partial function $L\colon \mathrm{Ob}(\mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}) \to \mathbb{R}$, and finally proving the main theorem: that $H(P) = L(\hbar(p))$.

There have been other categorical approaches to entropy, most notably [BFL11], [BF14], [Lei21], and [Par22]. Our presentation here has almost nothing in common with those.

However, this work is closely aligned with [SH21]. There, the authors—myself and Tim Hosgood—use Dirichlet polynomials rather than ordinary (Cartesian)[1] polynomials. At the time, we seemed to have a choice of whether to use Dirichlet or Cartesian polynomials, and the Dirichlet route seemed cleaner and more intuitive for talking about the bundles. However, we were missing a few key ideas at the time. Whereas there we only factored out from H a rig homomorphism (a function) Dir $\to$ Rect to a somewhat ad hoc rig we called Rect, the presentation here factors out from H a rig *functor* $\mathbf{Poly}^{\mathbf{Cart}} \to \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$. Thus it is a significant categorical upgrade.

[1] René Decartes at least invented the notation, e.g. $y^2 + 3y + 2$, for polynomials; hence we refer to them as *Cartesian polynomials* when we need to distinguish them from Dirichlet polynomials.

Acknowledgments

Thanks to Valeria de Paiva for interesting conversations, and thanks to the referees of ACT2022 for suggestions, e.g. leading to Remark 3.12.

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0348.

2 Background on polynomial functors

Readers familiar with the rig category $(\mathbf{Poly}, 0, +, y, \otimes)$ should skip to Section 3.1.

2.1 Basics

The main purpose of this section is to fix notation and provide a brief overview of polynomial functors in one variable. More extensive background material can be found in [SN22] and [GK12].

Definition 2.1 (Polynomial functor)**.** Given a set S, we denote the corresponding representable functor by

$$y^S := \mathbf{Set}(S, -)\colon \mathbf{Set} \to \mathbf{Set},$$

e.g. $y^S(X) := X^S$. In particular $y = y^1$ is the identity and $y^0 = 1$ is constant singleton.

A *polynomial functor* is a functor $p\colon \mathbf{Set} \to \mathbf{Set}$ that is isomorphic to a sum of representables, i.e. for which there exists a set T, a set $p[t] \in \mathbf{Set}$ for each $t \in T$, and an isomorphism of functors

$$p \cong \sum_{t \in T} y^{p[t]}.$$

We refer to T as the set of *p-types*, and for each type $t \in T$ we refer to $p[t]$ as the set of *p-terms of type t*.

A *morphism* $\varphi\colon p \to p'$ of polynomial functors is simply a natural transformation between them. It is called *cartesian* if for every map of sets $f\colon S \to S'$, the naturality square

$$\begin{array}{ccc} p(S) & \xrightarrow{p(f)} & p(S) \\ {\scriptstyle \varphi(S)}\downarrow & \lrcorner & \downarrow{\scriptstyle \varphi(S')} \\ p'(S') & \xrightarrow[p'(f)]{} & p'(S') \end{array}$$

is a pullback of sets. We denote the category of polynomial functors by **Poly** and the wide subcategory of polynomials and cartesian maps by $\mathbf{Poly}^{\mathbf{Cart}} \subseteq \mathbf{Poly}$. ◊

For any polynomial $p = \sum_{t \in T} y^{p[t]}$, we have a canonical isomorphism $p(1) \cong T$; hence from now on we will denote p by

$$p = \sum_{I \in p(1)} y^{p[I]} \tag{2}$$

so that each p-types is written with an upper-case letter, e.g. $I \in p(1)$, and its terms are written with corresponding lower-case letters, e.g. $i \in p[I]$.

Remark 2.2. Using the Yoneda lemma, the fact that a morphism in **Poly** is just a natural transformation, and the fact that a polynomial is a coproduct of representables, we derive

$$\begin{aligned}\mathbf{Poly}(p,q) &= \mathbf{Poly}\left(\sum_{I\in p(1)} y^{p[I]}, \sum_{J\in p(1)} y^{q[J]}\right)\\ &\cong \prod_{I\in p(1)} \mathbf{Poly}\left(y^{p[I]}, \sum_{J\in p(1)} y^{q[J]}\right)\\ &\cong \prod_{I\in p(1)} \sum_{J\in q(1)} \mathbf{Set}(q[J], p[I]).\end{aligned}$$

Thus we can understand a morphism $p\to q$ in **Poly** to consist of two parts $(\varphi_1, \varphi^\sharp)$ as follows:

$$\varphi_1\colon p(1)\to q(1) \qquad \text{and} \qquad \varphi_I^\sharp\colon q[J]\to p[I], \tag{3}$$

where $J := \varphi_1(I)$. That is, φ_1 is a function from p-types to q-types, and $\varphi_i^\sharp$ is a function on terms that *depends on a choice of position* $I\in p(1)$. We refer to φ_1 as the *on-types function* and to $\varphi^\sharp$ as the *backwards on-terms* function.

One can check that a map $\varphi\colon p\to q$ is cartesian iff the backwards-on-terms function $\varphi_I^\sharp$ is a bijection $p[I]\cong q[\varphi_1 I]$ for each type $I\in p(1)$. ♢

Example 2.3 (Types and global sections, $p(1)$ and $\Gamma(p)$)**.** For any polynomial p, we will be particularly interested in two sorts of maps: $y\to p$ and $p\to y$. The former is easy: a map $y\to p$ is given on types by choosing a single type $I\in p(1)$ to be the image of the unique type $!\in y(1)$ and given backward on terms using the unique choice of function $p[I]\to 1 = y[!]$. Thus we have $p(1)\cong \mathbf{Poly}(y,p)$.

More interesting are the maps $\gamma\colon p\to y$. This time γ is trivial on types: each type $I\in p(1)$ is sent to the unique type $!\in y(1)$. However on terms, we need a map $\varphi_I^\sharp\colon 1\to p[I]$ for each I, meaning a choice of term $i\in p[I]$ for each $I\in p(1)$. In other words, writing $\Gamma(p) := \mathbf{Poly}(p,y)$, we have

$$\Gamma(p)\cong \prod_{I\in p(1)} p[I]. \tag{4}$$

We refer to $\Gamma(p)$ as the set of *global sections* of p, as is justified by the bundle terminology the next section.

Note that $-(1)\colon \mathbf{Poly}\to\mathbf{Set}$ and $\Gamma\colon \mathbf{Poly}\to\mathbf{Set}^{\mathrm{op}}$ are functorial, as they are represented and corepresented by $y\in\mathbf{Poly}$. We will be very interested in the functor

$$R\colon \mathbf{Poly}\to\mathbf{Set}\times\mathbf{Set}^{\mathrm{op}} \tag{5}$$

given by $R(p) := (p(1), \Gamma(p))$. In fact, R is a left adjoint, but we do not need that for this paper. In Remark 3.6 we will explain that $R(p)$ can be viewed as the *rectangular aspect* of the polynomial p, hence the name R. ♢

2.2 Derivatives and bundles

We can understand polynomial functors in terms of bundles, using the derivative. For any polynomial p, its derivative $\dot{p}$ is defined as follows:

$$\dot{p} := \sum_{I\in p(1)} \sum_{i\in p[I]} y^{p[I]-\{i\}} \tag{6}$$

where $p[I]-\{i\}$ denotes the set-difference. Note that $\dot{p}(1)\cong\sum_{I\in p(1)}p[I]$ is the set of all p-terms, and it comes with a map $\dot{p}(1)\to p(1)$ to the set of p-types. Often in the literature, this map of sets—which we call a bundle—is taken to be the polynomial itself. A map of polynomials $\varphi\colon p\to q$ can be written in terms of these bundles:

$$
\begin{array}{ccccc}
\dot{p}(1) & \xleftarrow{\varphi^\sharp} & p(1)\times_{q(1)}\dot{q}(1) & \longrightarrow & \dot{q}(1)\\
\downarrow & & \downarrow \quad \lrcorner & & \downarrow\\
p(1) & = & p(1) & \xrightarrow[\varphi_1]{} & q(1)
\end{array}
$$

Just as in Remark 2.2, one provides a forward map on types $\varphi_1\colon p(1)\to q(1)$, at which point one takes the pullback of that map with $\dot{q}(1)\to q(1)$, and then one provides a backward map $\varphi^\sharp\colon p(1)\times_{q(1)}\dot{q}(1)\to\dot{p}(1)$ on directions. Again, φ is cartesian iff $\varphi^\sharp$ is a bijection.

We write $pq=p\times q$ for the usual product of two polynomials, e.g. $\dot{p}y=\dot{p}\times y$.

Proposition 2.4. *The assignment $p\mapsto\dot{p}y$ is a functor $\mathbf{Poly}^{\mathbf{Cart}}\to\mathbf{Poly}^{\mathbf{Cart}}$.*

Proof. We can think of $\dot{p}y$ as follows:

$$\dot{p}y\cong\sum_{I\in p(1)}\sum_{i\in p[I]}y^{p[I]} \tag{7}$$

Given a cartesian map $\varphi\colon p\to q$, the bijection $\varphi^\sharp\colon q[\varphi_1(I)]\cong p[I]$ lets us define a map $\dot{p}y\to\dot{q}y$ in an obvious way. □

Remark 2.5. In fact, the assignment $(p\mapsto\dot{p}y)\colon\mathbf{Poly}^{\mathbf{Cart}}\to\mathbf{Poly}^{\mathbf{Cart}}$ extends to a comonad on $\mathbf{Poly}^{\mathbf{Cart}}$. The counit map $\varepsilon_p\colon\dot{p}y\to p$ is cartesian and is given on types by $(I,i)\mapsto I$. The comultiplication $\delta_p\colon\dot{p}y\to\ddot{p}y^2+\dot{p}y$ is given by the coproduct inclusion.

A coalgebra for this comonad is a polynomial p equipped with a map $\gamma\colon p\to\dot{p}y$ such that $\varepsilon_p\circ\gamma=\mathrm{id}_p$; it is not hard to check that the other condition holds for free. Hence a coalgebra structure on p can be identified with a choice a global section $p\to y$, i.e. an element $\gamma\in\Gamma(p)$. Of course the map $p\to y$ is not cartesian in general, so the only way it can be encoded in $\mathbf{Poly}^{\mathbf{Cart}}$ is via this coalgebra structure. A map of coalgebras is a cartesian map $\varphi\colon p\to p'$ that commutes with the global sections: $\Gamma(\varphi)(\gamma')=\gamma$.

The above is intriguing in that $\Gamma(p)$ is a major player in the story of this paper, but we currently know of no further connection between entropy and this comonad. ◊

2.3 Rig monoidal structure on Poly

The category **Poly** has coproducts $p+q$ and products $p\times q$ given by usual polynomial arithmetic. We will be more interested in the former:[2] coproducts constitute a symmetric monoidal product with unit 0. A type in $p+q$ is a type in p or disjointly a type in q, and a term of that type is as specified in p or q, respectively.

[2]The only reason we introduce $\times$ for **Poly** is to explain that the polynomial product $\dot{p}y$ is in fact the categorical product $\dot{p}y\cong\dot{p}\times y$.

We will also be interested in another monoidal product called *Dirichlet product* and denoted $-\otimes-$; the types and terms of $p\otimes q$ are given by the following formula:

$$\left(\sum_{I\in p(1)} y^{p[I]}\right)\otimes\left(\sum_{J\in q(1)} y^{q[J]}\right) := \sum_{(I,J)\in p(1)\times q(1)} y^{p[I]\times q[J]}. \tag{8}$$

This gives a symmetric monoidal structure $(\mathbf{Poly}, y, \otimes)$. A type in $p\otimes q$ is just a pair of types $(I,J)\in p(1)\times q(1)$ and a term of it is just a pair of terms $(i,j)\in p[I]\times q[J]$.

In the language of bundles, $p+q$ and $p\otimes q$ are respectively given by

$$\begin{array}{ccc} \dot{p}(1)+\dot{q}(1) & \qquad & \dot{p}(1)\times\dot{q}(1) \\ \downarrow & & \downarrow \\ p(1)+q(1) & & p(1)\times q(1) \end{array}$$

i.e. $(p\dot{+}q)(1)\cong\dot{p}(1)+\dot{q}(1)$ and $(p\dot{\otimes}q)(1)\cong\dot{p}(1)\times\dot{q}(1)$.

The $\otimes$-structure distributes over the $+$ structure:

$$p\otimes(q_1+q_2)\cong(p\otimes q_1)+(p\otimes q_2),$$

thus making $(\mathbf{Poly}, 0, +, y, \otimes)$ a distributive monoidal category, and in particular a rig monoidal category.

Remark 2.6 (Leibniz and chain rules)**.** Some readers may be interested in the Leibniz rule and chain rule, that

$$\begin{aligned} (p\dot{\times}q) &\cong \dot{p}\times q+p\times\dot{q} \\ (p\dot{\triangleleft}q) &\cong (\dot{p}\triangleleft q)\times\dot{q} \end{aligned}$$

where $\times$ is the categorical product and $\triangleleft$ is the composition product in **Poly**. These hold, but we will not need them in this paper. ◊

3 Main results

We divide this section into two parts. Section 3.1 is the category theory part, in which we provide what seems to be a novel symmetric monoidal structure on $\mathbf{Set}\times\mathbf{Set}^{\mathrm{op}}$ and show that both $p\mapsto\dot{p}y$ and $q\mapsto(q(1),\Gamma(q))$ are rig functors. At the end of this section, we will have a rig functor $\hbar\colon \mathbf{Poly}^{\mathbf{Cart}}\to\mathbf{Set}\times\mathbf{Set}^{\mathrm{op}}$ that does the categorical work of Shannon entropy.

Section 3.2 is the finishing step, providing a function $\mathrm{Ob}(\mathbf{Set}\times\mathbf{Set}^{\mathrm{op}})\to\mathbb{R}$ and showing that when it is combined with the above, the map $H\colon \mathrm{Ob}(\mathbf{Poly}^{\mathbf{Cart}})\to\mathbb{R}$ sends an appropriately finite polynomial p to the Shannon entropy of the empirical distribution defined by p.

3.1 Categorical entropy of a polynomial

Below we will often denote products of sets by juxtaposition, $AB := A\times B$. Recall the functor $p\mapsto\dot{p}y$ from Proposition 2.4.

Proposition 3.1. *The functor $p \mapsto \dot{p}y$ is a rig functor $\textbf{Poly}^{\textbf{Cart}} \to \textbf{Poly}^{\textbf{Cart}}$.*

Proof. Clearly $\dot{0} = 0$ and $(p \dot{+} q) \cong \dot{p} + \dot{q}$, and by multiplying both sides by y we see that the functor $p \mapsto py$ preserves the coproduct structure. There is an isomorphism $\dot{y}y \cong y$, and for any $p, q \in \textbf{Poly}^{\textbf{Cart}}$ there is also an isomorphism $(p \dot{\otimes} q)y \cong (\dot{p}y) \otimes (\dot{q}y)$, as follows from (7) and (8); thus $p \mapsto \dot{p}y$ preserves the $\otimes$-structure. All of these isomorphisms are natural in $p, q \in \textbf{Poly}^{\textbf{Cart}}$, completing the proof. □

The following corollary is straightforward, since $\textbf{Poly}^{\textbf{Cart}}$ inherits $+$ and $\otimes$ from the forgetful functor $\textbf{Poly}^{\textbf{Cart}} \to \textbf{Poly}$.

Corollary 3.2. *The functor $T(p) := \dot{p}y$ is a rig functor $T\colon \textbf{Poly}^{\textbf{Cart}} \to \textbf{Poly}$.*

Remark 3.3 (Total polynomial)**.** Note that for any p we have $(\dot{p}y)(1) \cong \dot{p}(1)$. We think of $\dot{p}y$ as the *total polynomial* of p, akin to the total space of a bundle, where p is playing the role of the base. To justify this intuition, note that $\dot{p}y$ comes with a "projection" map $\varepsilon\colon \dot{p}y \to p$ and that a section $p \to \dot{p}y$ of ε can be identified with a section $\gamma \in \Gamma(p)$ of p as a bundle; see Remark **??**. ◊

Example 3.4. For any polynomial p, we have

$$\Gamma(\dot{p}y) \cong \prod_{I \in p(1)} p[I]^{p[I]}.$$

This formula—which follows directly from Eq. **??**—will be relevant when connecting the category theory to Shannon entropy later on. ◊

Proposition 3.5. *The category $\textbf{Set} \times \textbf{Set}^{\mathrm{op}}$ has a distributive monoidal structure:*

$$(A_1, B_1) + (A_2, B_2) := (A_1 + A_2\,, B_1 B_2) \tag{9}$$

$$(A_1, B_1) \otimes (A_2, B_2) := (A_1 A_2\,, B_1^{A_2} B_2^{A_1}) \tag{10}$$

The units are $(0,1)$ and $(1,1)$ respectively.

Proof. Coproducts in $\textbf{Set}^{\mathrm{op}}$ are products in $\textbf{Set}$, justifying the first line; these clearly form a symmetric monoidal structure. For the $\otimes$-monoidal structure, note that the formula is functorial in $A \in \textbf{Set}$ and $B \in \textbf{Set}^{\mathrm{op}}$. It is also symmetric as well as unital: $(1,1) \otimes (A_2, B_2) \cong (A_2, B_2)$. Associativity is justified as follows:

$$\begin{aligned}(A_1, B_1) \otimes ((A_2, B_2) \otimes (A_3, B_3)) &\cong (A_1 A_2 A_3, B_1^{A_2 A_3} B_2^{A_1 A_3} B_3^{A_1 A_2}) \\ &\cong ((A_1, B_1) \otimes (A_2, B_2)) \otimes (A_3, B_3).\end{aligned}$$

There is an absorption map $(0,1) \otimes (A, B) \cong (0, B) \to (0,1)$, and the distributivity of $\otimes$ over $+$ is justified as follows:

$$\begin{aligned}(A, B) \otimes \big((A_1, B_1) + (A_2, B_2)\big) &\cong \big(A(A_1 + A_2), B^{A_1 + A_2}(B_1 B_2)^A\big) \\ &\cong \big(AA_1 + AA_2, B^{A_1} B^{A_2} B_1^A B_2^A\big) \\ &\cong \big((A, B) \otimes (A_1, B_1)\big) + \big((A, B) \otimes (A_2, B_2)\big).\end{aligned}$$

We leave the remaining details to the interested reader. □

Remark 3.6 (Formal roots and rectangular aspect)**.** One can think of an object $(A,B) \in \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ as formally representing the Ath root of B, i.e. the number $\sqrt[A]{B} = B^{\frac{1}{A}}$, keeping track of the base A as well. It is helpful to think of (A,B) as a rectangle with length A and width $\sqrt[A]{B}$. From this perspective, the sum from (9) adds the lengths and takes the geometric mean of the widths, and the monoidal product from (10) takes the product of both lengths and widths:

$$(B_1B_2)^{\frac{1}{A_1+A_2}} = \left(\left(\sqrt[A_1]{B_1}\right)^{A_1} \times \left(\sqrt[A_2]{B_2}\right)^{A_2}\right)^{\frac{1}{A_1+A_2}} \qquad \text{and} \qquad (B_1^{A_2}B_2^{A_1})^{\frac{1}{A_1A_2}} = \sqrt[A_1]{B_1}\,\sqrt[A_2]{B_2}.$$

For any polynomial p, the functor $R(p) := (p(1), \Gamma(p))$ from (5) is consonant with this interpretation. We may say that $R(p)$ denotes the *rectangular aspect* of p in the sense that $p(1)$ represents the length and $\sqrt[p(1)]{\Gamma(p)}$, the geometric mean of the fiber cardinalities, represents the width. For example, the polynomial $p = y^4 + 4y$, depicted in Diagram (1), has length $p(1) = 5$ and width $\sqrt[5]{4} \approx 1.3$. ◊

Remark 3.7. The $\otimes$ operation (10) on $\mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ in fact has a closure

$$[(A_1,B_1),(A_2,B_2)] := \left(A_2^{A_1}B_1^{B_2}, A_1B_2\right).$$

We will not need this, but it is interesting that $\mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ has so much structure. ◊

Proposition 3.8. *The functor $R\colon \mathbf{Poly} \to \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ from* (5) *is a rig functor.*

Proof. Recall from (5) that $R(p) := (p(1), \Gamma(p))$. Clearly $0(1) = 0$ and $(p+q)(1) \cong p(1) + q(1)$. Also $\Gamma(0) = 1$ and $\Gamma(p+q) \cong \Gamma(p) \times \Gamma(q)$; hence R preserves the $(0,+)$ monoidal structure. Moreover, we have $y(1) = 1$ and $(p \otimes q)(1) \cong p(1) \times q(1)$ and $\Gamma(y) = 1$, so to show that R preserves the $(y, \otimes)$ monoidal structure, it remains only to provide an isomorphism

$$\Gamma(p \otimes q) \cong \Gamma(p)^{q(1)} \times \Gamma(q)^{p(1)}.$$

It is given as follows:

$$\begin{aligned}
\Gamma(p \otimes q) &\cong \prod_{(I,J) \in p(1) \times q(1)} p[I]q[J] \\
&\cong \left(\prod_{(I,J) \in p(1) \times q(1)} p[I]\right) \times \left(\prod_{(I,J) \in p(1) \times q(1)} q[J]\right) \\
&\cong \prod_{J \in q(1)} \prod_{I \in p(1)} p[I] \times \prod_{I \in p(1)} \prod_{J \in q(1)} q[J] \\
&\cong \Gamma(p)^{q(1)} \times \Gamma(q)^{p(1)}
\end{aligned}$$

□

We summarize the above section before we go on to the final one. Namely, the functors $T\colon \mathbf{Poly}^{\mathbf{Cart}} \to \mathbf{Poly}$ and $R\colon \mathbf{Poly} \to \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ from Corollary **??** compose to form a rig functor $\hbar := R \circ T$ given by

$$\begin{aligned}
\mathbf{Poly}^{\mathbf{Cart}} &\xrightarrow{\hbar} \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}} \\
p &\mapsto (\dot{p}(1), \Gamma(\dot{p}y)).
\end{aligned} \tag{11}$$

We refer to $\hbar(p) \in \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ as *the categorical entropy* of the polynomial p. This pair of sets leaves behind any semblance of the probability distribution associated with p, but it retains the data necessary to compute p's entropy—as we'll see in Theorem 3.13—and it is rig-functorial in p.

3.2 Shannon entropy

Writing log to denote $\log_2$, we define a partial function $L\colon \operatorname{Ob}(\mathbf{Set}\times\mathbf{Set}^{\mathrm{op}})\to\mathbb{R}$ by

$$(A,B)\mapsto \log A-\frac{\log B}{A}. \tag{12}$$

Equivalently, $L(A,B)=\log A-\log\sqrt[A]{B}$. When $A=0$ and $B=1$, we define this function to be $L(0,1):=0$; for all cases where $A=0$, or $B=0$, or either A or B is infinite, we leave $L(A,B)$ undefined. We will be only interested in this map when it is composed with the categorical entropy $\hbar$ from (11), and Lemma 3.9 below says that we do not need to worry about the undefined cases.

Lemma 3.9. *Let $p\in\mathbf{Poly}^{\mathbf{Cart}}$ with categorical entropy $(A,B):=\hbar(p)$, and suppose that $^{\#}\dot{p}(1)<\infty$. Then we have that*

i. $B\neq 0$,
ii. if $A=0$ then $B=1$, and
iii. both A and B are finite.

Proof. By definition of $\hbar$, we have that $A:=\dot{p}(1)$ and $B:=\Gamma(\dot{p}y)$.

i. One easily checks using (4) that for any $q\in\mathbf{Poly}$, the set $\Gamma(qy)\neq 0$ is nonempty since every (qy)-type has at least one term.
ii. If $\dot{p}(1)=0$ then $p\in\mathbf{Set}$ is constant, so $\dot{p}y=0$ as well, and $\Gamma(0)=1$ by (4).
iii. By assumption $^{\#}A={}^{\#}\dot{p}(1)<\infty$. For B, note that there are only a finite number of $I\in p(1)$ for which $p[I]$ is nonempty, so by (4) and (6) the set $\Gamma(\dot{p}y)$ is finite. □

Remark 3.10 (Log aspect ratio)**.** With the interpretation of an object $(A,B)\in\mathbf{Set}\times\mathbf{Set}^{\mathrm{op}}$ as a rectangle with length A and width $\sqrt[A]{B}$, as in Remark 3.6, we can think of $L(A,B)=\log A-\log\sqrt[A]{B}$ as its *log aspect ratio*, the log of its length divided by its width. This is a quantity that has come up in the study of vision [TGH11; Dic+17], though we're making no claim about whether this connection is meaningful. ♢

Definition 3.11 (Empirical distribution)**.** Let $p\neq 0$ be a nonzero polynomial and suppose that the cardinality of $\dot{p}(1)\in\mathbf{Set}$ is finite, $^{\#}\dot{p}(1)<\infty$. We define the *empirical distribution defined by p* to be the following function $P\colon p(1)\to[0,1]$:

$$P(I):=\frac{^{\#}p[I]}{^{\#}\dot{p}(1)}$$

Note that $1=\sum_{I\in p(1)}P(I)$, so P is indeed a probability distribution. ♢

Remark 3.12. One may ask how to view **Poly**'s monoidal structures, especially $+$ and $\otimes$, under the correspondence from Definition 3.11. Suppose given polynomials $p,q\in\mathbf{Poly}$ with associated probability distributions P_p and P_q. For Dirichlet product we have

$$P_{p\otimes q}=P_p\otimes P_q$$

where the left-hand side is the probability distribution associated to $p\otimes q$ and the right-hand side is the usual tensor (independent) product of probability distributions. For sums we have

$$P_{p+q}=\frac{\dot{p}(1)}{\dot{p}(1)+\dot{q}(1)}P_p+\frac{\dot{q}(1)}{\dot{p}(1)+\dot{q}(1)}P_q$$

the convex combination of P_p and Q_q, weighted according to the relative number of draws $\dot{p}(1)$ and $\dot{q}(1)$ in each. $\Diamond$

Recall that the Shannon entropy $H(P)$ of a probability distribution $P\colon X \to [0,1]$ is given by

$$H(P) := -\sum_{x\in X} P(x)\log P(x).$$

The following theorem could be summarized as follows: "thinking of $p \in \mathbf{Poly}^{\mathbf{Cart}}$ as a statistical sample, the entropy $H(P)$ of the corresponding probability distribution P is equal to the log ratio of the rectangular aspect of p's total polynomial"; see Remarks 3.3, 3.6, and 3.10.

Theorem 3.13. *Let $p \neq 0$ be a nonzero polynomial with ${}^{\#}\dot{p}(1) < \infty$, and let P be the empirical distribution defined by p. Then the following equation holds*

$$H(P) = L(\hbar(p))$$

where H is the Shannon entropy and $L, \hbar$ are as defined in Eqs. (11) *and* (12)*.*

Proof. We need to show that the following holds:

$$H(P) = \log\dot{p}(1) - \frac{\log\Gamma(\dot{p}y)}{\dot{p}(1)}.$$

With the fact $\Gamma(\dot{p}y) \cong \prod_{I\in p(1)} p[I]^{p[I]}$ from Example 3.4 in hand, this is a routine calculation:

$$\begin{aligned}
H(P) &:= -\sum_{I\in p(1)} \frac{{}^{\#}p[I]}{{}^{\#}\dot{p}(1)} \log\frac{{}^{\#}p[I]}{{}^{\#}\dot{p}(1)} \\
&= \frac{1}{{}^{\#}\dot{p}(1)} \sum_{I\in p(1)} {}^{\#}p[I]\big(\log{}^{\#}\dot{p}(1) - \log{}^{\#}p[I]\big) \\
&= \frac{1}{{}^{\#}\dot{p}(1)} \left({}^{\#}\dot{p}(1)\log{}^{\#}\dot{p}(1) - \log\prod_{I\in p(1)} {}^{\#}p[I]^{{}^{\#}p[I]}\right) \\
&= \log{}^{\#}\dot{p}(1) - \frac{\log\Gamma(\dot{p}y)}{\dot{p}(1)}
\end{aligned}$$

$\square$

Example 3.14 (Entropy of a uniform distribution)**.** It is well-known and easy to calculate that if P is a uniform distribution on A elements, then $H(P) = \log(A)$. There are many samples that correspond to P; what differs are their sample sizes. The sample in which AB-many observations are taken—each outcome occurring B-many times—corresponds to the polynomial Ay^B.

Our formula for entropy needs to agree, and it does. The rectangular aspect of the total polynomial is $\hbar(p) \cong (AB, B^{AB})$: length AB and width $B = \sqrt[AB]{B^{AB}}$, so its log aspect ratio is

$$L(\hbar(p)) = \log(AB) - \frac{\log(B^{AB})}{AB} = \log A. \qquad \Diamond$$

References

[AAG03] Michael Gordon Abbott, Thorsten Altenkirch, and Neil Ghani. “Categories of Containers”. In: *FoSSaCS*. 2003. DOI: https://doi.org/10.1007/3-540-36576-1_2 (cit. on p. 332).

[ACH19] Jeremy Avigad, Mario Carneiro, and Simon Hudon. “Data types as quotients of polynomial functors”. In: *10th International Conference on Interactive Theorem Proving (ITP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2019. DOI: https://doi.org/10.4230/LIPIcs.ITP.2019.6 (cit. on p. 332).

[AN18] Steve Awodey and Clive Newstead. *Polynomial pseudomonads and dependent type theory*. 2018. DOI: https://doi.org/10.48550/arXiv.1802.00997. arXiv: 1802.00997 [math.CT] (cit. on p. 332).

[BD96] Richard Bird and Oege De Moor. “The algebra of programming”. In: *NATO ASI DPD*. 1996, pp. 167–203. DOI: https://doi.org/10.1017/S095679689922326X (cit. on p. 332).

[BF14] John C. Baez and Tobias Fritz. “A Bayesian characterization of relative entropy”. In: *Theory and Applications of Categories* 29 (2014), Paper No. 16, 421–456. DOI: https://doi.org/10.48550/arXiv.1402.3067 (cit. on p. 333).

[BFL11] John C. Baez, Tobias Fritz, and Tom Leinster. “A Characterization of Entropy in Terms of Information Loss”. In: *Entropy* 13.11 (2011), pp. 1945–1957. URL: https://www.mdpi.com/1099-4300/13/11/1945 (cit. on p. 333).

[Dic+17] J Edwin Dickinson, Sarah K Morgan, Matthew F Tang, and David R Badcock. “Separate banks of information channels encode size and aspect ratio”. In: *Journal of Vision* 17.3 (2017), pp. 27–27. DOI: https://doi.org/10.1167/17.3.27 (cit. on p. 340).

[GK12] Nicola Gambino and Joachim Kock. “Polynomial functors and polynomial monads”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 154.1 (September 2012), pp. 153 –192. DOI: https://doi.org/10.1017/S0305004112000394 (cit. on p. 334).

[Lei21] Tom Leinster. *Entropy and Diversity: The Axiomatic Approach*. Cambridge University Press, 2021. DOI: https://doi.org/10.1017/9781108963558 (cit. on p. 333).

[Par22] Arthur Parzygnat. “A functorial characterization of von Neumann entropy”. In: *Cahiers* 63 (2022). DOI: https://doi.org/10.48550/arXiv.2009.07125 (cit. on p. 333).

[SH21] David I. Spivak and Timothy Hosgood. “Dirichlet Polynomials and Entropy”. In: *Entropy* 23.8 (2021). DOI: https://doi.org/10.3390/e23081085. URL: https://www.mdpi.com/1099-4300/23/8/1085 (cit. on p. 333).

[Sha21] Brandon Shapiro. “Familial Monads as Higher Category Theories”. In: *arXiv:2111.14796* (2021). DOI: https://doi.org/10.48550/arXiv.2111.14796 (cit. on p. 332).

[Sha48] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x (cit. on p. 332).

[SN22] David I. Spivak and Nelson Niu. *Polynomial functors: a general theory of interaction. In preparation*. 2022 (cit. on pp. 332, 334).

[Spi20] David I. Spivak. "Poly: An abundant categorical setting for mode-dependent dynamics". In: (2020). DOI: `https://doi.org/10.48550/arXiv.2005.01894`. eprint: `arXiv:2005.01894` (cit. on p. 332).

[Spi21] David I. Spivak. *Functorial aggregation*. 2021. DOI: `https://doi.org/10.48550/arXiv.2111.10968`. arXiv: `2111.10968 [math.CT]` (cit. on p. 332).

[SW15] David I. Spivak and Ryan Wisnesky. "Relational Foundations for Functorial Data Migration". In: *Proceedings of the 15th Symposium on Database Programming Languages*. DBPL. Pittsburgh, PA: ACM, 2015, pp. 21–28. DOI: `https://doi.org/10.1145/2815072.2815075` (cit. on p. 332).

[TCM19] Cédric Ho Thanh, Pierre-Louis Curien, and Samuel Mimram. "A sequent calculus for opetopes". In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE. 2019, pp. 1–12. DOI: `https://doi.org/10.1109/LICS.2019.8785667` (cit. on p. 332).

[TGH11] Justin Talbot, John Gerth, and Pat Hanrahan. "Arc length-based aspect ratio selection". In: *IEEE transactions on visualization and computer graphics* 17.12 (2011), pp. 2276–2282. DOI: `https://doi.org/10.1109/TVCG.2011.167` (cit. on p. 340).

Promonads and String Diagrams for Effectful Categories

Mario Román
Tallinn University of Technology
mroman@ttu.ee

Premonoidal and Freyd categories are both generalized by non-cartesian Freyd categories: effectful categories. We construct string diagrams for effectful categories in terms of the string diagrams for a monoidal category with a freely added object. We show that effectful categories are pseudomonoids in a monoidal bicategory of promonads with a suitable tensor product.

1 Introduction

Category theory has two sucessful applications that are rarely combined: monoidal string diagrams [23] and functional programming semantics [28]. We use string diagrams to talk about quantum transformations [1], relational queries [6], and even computability [31]; at the same time, proof nets and the geometry of interaction [13, 5] have been widely applied in computer science [2, 18]. On the other hand, we traditionally use monads and comonads, Kleisli categories and premonoidal categories to explain effectful functional programming [19, 20, 28, 34, 42]. Even if we traditionally employ Freyd categories with a cartesian base [32], we can also consider non-cartesian Freyd categories [40], which we call *effectful categories*.

Contributions. These applications are well-known. However, some foundational results in the intersection between string diagrams, premonoidal categories and effectful categories are missing in the literature. This manuscript contributes two such results.

- We introduce string diagrams for effectful categories. Jeffrey [22] was the first to preformally employ string diagrams of premonoidal categories. His technique consists in introducing an extra wire – which we call the *runtime* – that prevents some morphisms from interchanging. We promote this preformal technique into a result about the construction of free premonoidal, Freyd and effectful categories: the free premonoidal category can be constructed in terms of the free monoidal category with an extra wire.

  Our slogan, which constitutes the statement of Theorem 2.14, is

  "Premonoidal categories are Monoidal categories with a Runtime."

- We prove that effectful categories are promonad pseudomonoids. Promonads are the profunctorial counterpart of monads; they are used to encode effects in functional programming (where they are given extra properties and called *arrows* [19]). We claim that, in the same way that monoidal categories are pseudomonoids in the bicategory of categories [9], premonoidal effectful categories are pseudomonoids in a monoidal bicategory of promonads. This result justifies the role of effectful categories as a foundational object.

1.1 Synopsis

Sections 2.1 and 2.2 contain mostly preliminary material on premonoidal, Freyd and effectful categories. Our first original contribution is in Section 2.3; we prove that premonoidal categories are monoidal

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 344–361, doi:10.4204/EPTCS.380.20

categories with runtime (Theorem 2.14). Section 3 makes explicit the well-known theory of profunctors, promonads and identity-on-objects functors. In Section 4, we introduce the pure tensor of promonads. We use it in Section 5 to prove our second main contribution (Theorem 5.3).

2 Premonoidal and Effectful Categories

2.1 Premonoidal categories

Premonoidal categories are monoidal categories without the *interchange law*, $(f \otimes \mathrm{id}) \fatsemi (\mathrm{id} \otimes g) \neq (\mathrm{id} \otimes g) \fatsemi (f \otimes \mathrm{id})$. This means that we cannot tensor any two arbitrary morphisms, $(f \otimes g)$, without explicitly stating which one is to be composed first, $(f \otimes \mathrm{id}) \fatsemi (\mathrm{id} \otimes g)$ or $(\mathrm{id} \otimes g) \fatsemi (f \otimes \mathrm{id})$, and the two compositions are not equivalent (Figure 1).

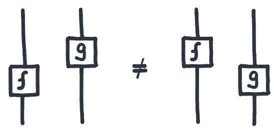


Figure 1: The interchange law does not hold in a premonoidal category.

In technical terms, the tensor of a premonoidal category $(\otimes) \colon \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ is not a functor, but only what is called a *sesquifunctor*: independently functorial on each variable. Tensoring with any identity is itself a functor $(\bullet \otimes \mathrm{id}) \colon \mathbb{C} \to \mathbb{C}$, but there is no functor $(\bullet \otimes \bullet) \colon \mathbb{C} \times \mathbb{C} \to \mathbb{C}$.

A good motivation for dropping the interchange law can be found when describing transformations that affect some global state. These effectful processes should not interchange in general, because the order in which we modify the global state is meaningful. For instance, in the Kleisli category of the *writer monad*, $(\Sigma^* \times \bullet) \colon \mathsf{Set} \to \mathsf{Set}$ for some alphabet $\Sigma \in \mathsf{Set}$, we can consider the function $\mathsf{print} \colon \Sigma^* \to \Sigma^* \times 1$. The order in which we "print" does matter (Figure 2).

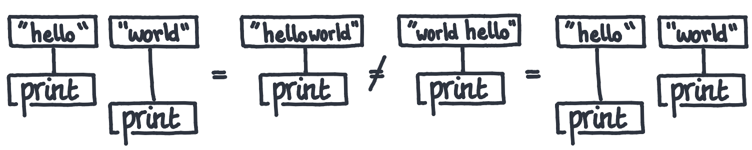


Figure 2: Writing does not interchange.

Not surprisingly, the paradigmatic examples of premonoidal categories are the Kleisli categories of Set-based monads $T \colon \mathsf{Set} \to \mathsf{Set}$ (more generally, of strong monads), which fail to be monoidal unless the monad itself is commutative [15, 33, 34, 16]. Intuitively, the morphisms are "effectful", and these effects do not always commute.

However, we may still want to allow some morphisms to interchange. For instance, apart from asking the same associators and unitors of monoidal categories to exist, we ask them to be *central*: that means that they interchange with any other morphism. This notion of centrality forces us to write the definition of premonoidal category in two different steps: first, we introduce the minimal setting in which centrality can be considered (*binoidal* categories [34]) and then we use that setting to bootstrap the full definition of premonoidal category with central coherence morphisms.

Definition 2.1 (Binoidal category)**.** A *binoidal category* is a category $\mathbb{C}$ endowed with an object $I \in \mathbb{C}$ and an object $A \otimes B$ for each $A \in \mathbb{C}$ and $B \in \mathbb{C}$. There are functors $(A \otimes \bullet) \colon \mathbb{C} \to \mathbb{C}$, and $(\bullet \otimes B) \colon \mathbb{C} \to \mathbb{C}$ that coincide on $(A \otimes B)$, even if $(\bullet \otimes \bullet)$ is not itself a functor.

Again, this means that we can tensor with identities (whiskering), functorially; but we cannot tensor two arbitrary morphisms: the interchange law stops being true in general. The *centre*, $\mathscr{Z}(\mathbb{C})$, is the wide subcategory of morphisms that do satisfy the interchange law with any other morphism. That is, $f\colon A \to B$ is *central* if, for each $g\colon A' \to B'$,

$$(f \otimes \mathrm{id}_{A'}) \mathbin{⨟} (\mathrm{id}_B \otimes g) = (\mathrm{id}_A \otimes g) \mathbin{⨟} (f \otimes \mathrm{id}_{B'}), \text{ and } (\mathrm{id}_{A'} \otimes f) \mathbin{⨟} (g \otimes \mathrm{id}_B) = (g \otimes \mathrm{id}_A) \mathbin{⨟} (\mathrm{id}_{B'} \otimes f).$$

Definition 2.2. A *premonoidal category* is a binoidal category $(\mathbb{C}, \otimes, I)$ together with the following coherence isomorphisms $\alpha_{A,B,C}\colon A \otimes (B \otimes C) \to (A \otimes B) \otimes C$, $\rho_A\colon A \otimes I \to A$ and $\lambda_A\colon I \otimes A \to A$ which are central, natural *separately at each given component*, and satisfy the pentagon and triangle equations.

A premonoidal category is *strict* when these coherence morphisms are identities. A premonoidal category is moreover *symmetric* when it is endowed with a coherence isomorphism $\sigma_{A,B}\colon A \otimes B \to B \otimes A$ that is central and natural at each given component, and satisfies the symmetry condition and hexagon equations.

Remark 2.3. The coherence theorem of monoidal categories still holds for premonoidal categories: every premonoidal is equivalent to a strict one. We will construct the free strict premonoidal category using string diagrams. However, the usual string diagrams for monoidal categories need to be restricted: in premonoidal categories, we cannot consider two morphisms in parallel unless any of the two is *central*.

2.2 Effectful and Freyd categories

Premonoidal categories immediately present a problem: what are the strong premonoidal functors? If we want them to compose, they should preserve centrality of the coherence morphisms (so that the central coherence morphisms of $F \mathbin{⨟} G$ are these of F after applying G), but naively asking them to preserve all central morphisms rules out important examples [40]. The solution is to explicitly choose some central morphisms that represent "pure" computations. These do not need to form the whole centre: it could be that some morphisms considered *effectful* just "happen" to fall in the centre of the category, while we do not ask our functors to preserve them. This is the well-studied notion of a *non-cartesian Freyd category*, which we shorten to *effectful monoidal category* or *effectful category*.[1]

Effectful categories are premonoidal categories endowed with a chosen family of central morphisms. These central morphisms are called pure morphisms, constrasting with the general, non-central, morphisms that fall outside this family, which we call effectful.

Definition 2.4. An *effectful category* is an identity-on-objects functor $\mathbb{V} \to \mathbb{C}$ from a monoidal category $\mathbb{V}$ (the pure morphisms, or "values") to a premonoidal category $\mathbb{C}$ (the effectful morphisms, or "computations"), that strictly preserves all of the premonoidal structure and whose image is central. It is *strict* when both are. A *Freyd category* [24] is an effectful category where the pure morphisms form a cartesian monoidal category.

Effectful categories solve the problem of defining premonoidal functors: a functor between effectful categories needs to preserve only the pure morphisms. We are not losing expressivity: premonoidal categories are effectful with their centre, $\mathscr{Z}(\mathbb{C}) \to \mathbb{C}$. From now on, we study effectful categories.

[1] The name "Freyd category" sometimes assumes cartesianity of the pure morphisms, but it is also used for the general case. Choosing to call "effectful categories" to the general case and reserving the name "Freyd categories" for the cartesian ones avoids this clash of nomenclature. There exists also the more fine-grained notion of "Cartesian effect category" [12], which generalizes Freyd categories and may further justify calling "effectful category" to the general case.

Definition 2.5 (Effectful functor)**.** Let $\mathbb{V} \to \mathbb{C}$ and $\mathbb{W} \to \mathbb{D}$ be effectful categories. An *effectful functor* is a quadruple $(F, F_0, \varepsilon, \mu)$ consisting of a functor $F\colon \mathbb{C} \to \mathbb{D}$ and a functor $F_0\colon \mathbb{V} \to \mathbb{W}$ making the square commute, and two natural and pure isomorphisms $\varepsilon\colon J \cong F(I)$ and $\mu\colon F(A \otimes B) \cong F(A) \otimes F(B)$ such that they make F_0 a monoidal functor. It is *strict* if these are identities.

When drawing string diagrams in an effectful category, we shall use two different colours to declare if we are depicting either a value or a computation (Figure 3).

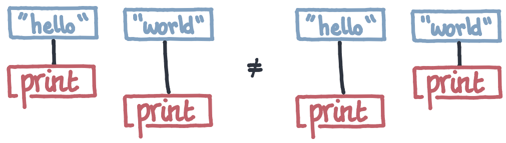


Figure 3: "Hello world" is not "world hello".

Here, the values "hello" and "world" satisfy the interchange law as in an ordinary monoidal category. However, the effectful computation "print" does not need to satisfy the interchange law. String diagrams like these can be found in the work of Alan Jeffrey [22]. Jeffrey presents a clever mechanism to graphically depict the failure of interchange: all effectful morphisms need to have a control wire as an input and output. This control wire needs to be passed around to all the computations in order, and it prevents them from interchanging.

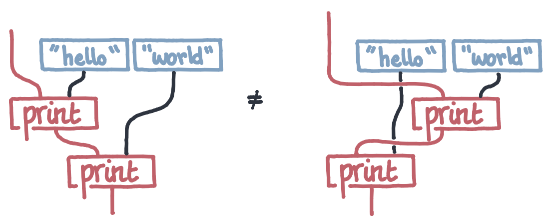


Figure 4: An extra wire prevents interchange.

A common interpretation of monoidal categories is as theories of resources. We can interpret premonoidal categories as monoidal categories with an extra resource – the "runtime" – that needs to be passed to all computations. The next section promotes Jeffrey's observation into a theorem.

2.3 Premonoidals are monoidals with runtime

String diagrams rely on the fact that the morphisms of the monoidal category freely generated over a polygraph of generators are string diagrams on these generators, quotiented by topological deformations [23]. We justify string diagrams for premonoidal categories by proving that the freely generated effectful category over a pair of polygraphs (for pure and effectful generators, respectively) can be constructed as the freely generated monoidal category over a particular polygraph that includes an extra wire.

Definition 2.6. A *polygraph* $\mathcal{G}$ (analogue of a *multigraph* [38]) is given by a set of objects, $\mathcal{G}_{\text{obj}}$, and a set of arrows $\mathcal{G}(A_0, \dots, A_n; B_0, \dots, B_m)$ for any two sequences of objects $A_0, \dots, A_n$ and $B_0, \dots, B_m$. A morphism of polygraphs $f\colon \mathcal{G} \to \mathcal{H}$ is a function between their object sets, $f_{\text{obj}}\colon \mathcal{G}_{\text{obj}} \to \mathcal{H}_{\text{obj}}$, and a function between their corresponding morphism sets,

$$f_{A_0,\dots,A_n;B_0,\dots,B_n}\colon \mathcal{G}(A_0, \dots, A_n; B_0, \dots, B_m) \to \mathcal{H}(f_{\text{obj}}(A_0), \dots, f_{\text{obj}}(A_n); f_{\text{obj}}(B_0), \dots, f_{\text{obj}}(B_m)).$$

A *polygraph couple* is a pair of polygraphs $(\mathscr{V},\mathscr{G})$ sharing the same objects, $\mathscr{V}_{\text{obj}} = \mathscr{G}_{\text{obj}}$. A morphism of polygraph couples $(u, f): (\mathscr{V},\mathscr{G}) \to (\mathscr{W},\mathscr{H})$ is a pair of morphisms of polygraphs, $u: \mathscr{V} \to \mathscr{W}$ and $f: \mathscr{G} \to \mathscr{H}$, such that they coincide on objects, $f_{\text{obj}} = u_{\text{obj}}$.

Remark 2.7. There exists an adjunction between polygraphs and strict monoidal categories. Any monoidal category $\mathbb{C}$ can be seen as a polygraph $\mathscr{U}_{\mathbb{C}}$ where the edges $\mathscr{U}_{\mathbb{C}}(A_0,\dots,A_n;B_0,\dots,B_m)$ are the morphisms $\mathbb{C}(A_0 \otimes \dots \otimes A_n, B_0 \otimes \dots \otimes B_m)$, and we forget about composition and tensoring. Given a polygraph $\mathscr{G}$, the free strict monoidal category $\mathrm{Mon}(\mathscr{G})$ is the strict monoidal category that has as morphisms the string diagrams over the generators of the polygraph.

We will construct a similar adjunction between polygraph couples and effectful categories. Let us start by formally adding the runtime to a free monoidal category.

Definition 2.8 (Runtime monoidal category)**.** Let $(\mathscr{V},\mathscr{G})$ be a polygraph couple. Its *runtime monoidal category*, $\mathrm{Mon}_{\mathrm{Run}}(\mathscr{V},\mathscr{G})$, is the monoidal category freely generated from adding an extra object – the runtime, R – to the input and output of every effectful generator in $\mathscr{G}$ (but not to those in $\mathscr{V}$), and letting that extra object be braided with respect to every other object of the category.

In other words, it is the monoidal category freely generated by the following polygraph, $\mathrm{Run}(\mathscr{V},\mathscr{G})$, (Figure 5), assuming $A_0,\dots,A_n$ and $B_0,\dots,B_m$ are distinct from R

- $\mathrm{Run}(\mathscr{V},\mathscr{G})_{\text{obj}} = \mathscr{G}_{\text{obj}} + \{R\} = \mathscr{V}_{\text{obj}} + \{R\}$,
- $\mathrm{Run}(\mathscr{V},\mathscr{G})(R,A_0,\dots,A_n;R,B_0,\dots,B_n) = \mathscr{G}(A_0,\dots,A_n;B_0,\dots,B_n)$,
- $\mathrm{Run}(\mathscr{V},\mathscr{G})(A_0,\dots,A_n;B_0,\dots,B_n) = \mathscr{V}(A_0,\dots,A_n;B_0,\dots,B_n)$,
- $\mathrm{Run}(\mathscr{V},\mathscr{G})(R,A_0;A_0,R) = \mathrm{Run}(\mathscr{V},\mathscr{G})(A_0,R;R,A_0) = \{\sigma\}$,

with $\mathrm{Run}(\mathscr{V},\mathscr{G})$ empty in any other case, and quotiented by the braiding axioms for R (Figure 6).

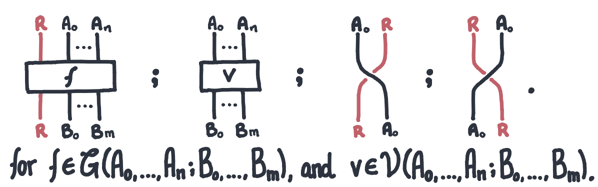


Figure 5: Generators for the runtime monoidal category.

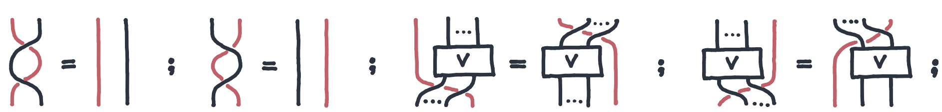


Figure 6: Axioms for the runtime monoidal category.

Somehow, we are asking the runtime R to be in the Drinfeld centre [11] of the monoidal category. The extra wire that R provides is only used to prevent interchange, and so it does not really matter where it is placed in the input and the output. We can choose to always place it on the left, for instance – and indeed we will be able to do so – but a better solution is to just consider objects "up to some runtime braidings". This is formalized by the notion of *braid clique*.

Definition 2.9 (Braid clique)**.** Given any list of objects $A_0,\dots,A_n$ in $\mathscr{V}_{\text{obj}} = \mathscr{G}_{\text{obj}}$, we construct a *clique* [41, 39] in the category $\mathrm{Mon}_{\mathrm{Run}}(\mathscr{V},\mathscr{G})$: we consider the objects, $A_0 \otimes \dots \otimes R_{(i)} \otimes \dots \otimes A_n$, created by inserting the runtime R in all of the possible $0 \leqslant i \leqslant n+1$ positions; and we consider the family of commuting isomorphisms constructed by braiding the runtime,

$$\sigma_{i,j} \colon A_0 \otimes \ldots \otimes R_{(i)} \otimes \ldots \otimes A_n \to A_0 \otimes \ldots \otimes R_{(j)} \otimes \ldots \otimes A_n.$$

We call this the *braid clique*, $\mathrm{Braid}_R(A_0,\ldots,A_n)$, on that list.

Definition 2.10. A *braid clique morphism*, $f \colon \mathrm{Braid}_R(A_0,\ldots,A_n) \to \mathrm{Braid}_R(B_0,\ldots,B_m)$ is a family of morphisms in the runtime monoidal category, $\mathsf{Mon}_{\mathsf{Run}}(\mathscr{V},\mathscr{G})$, from each of the objects of first clique to each of the objects of the second clique,

$$f_{ik} \colon A_0 \otimes \ldots \otimes R_{(i)} \otimes \ldots \otimes A_n \to B_0 \otimes \ldots \otimes R_{(k)} \otimes \ldots \otimes B_m,$$

that moreover commutes with all braiding isomorphisms, $f_{ij} ⨟ \sigma_{jk} = \sigma_{il} ⨟ f$.

A braid clique morphism $f \colon \mathrm{Braid}_R(A_0,\ldots,A_n) \to \mathrm{Braid}_R(B_0,\ldots,B_m)$ is fully determined by *any* of its components, by pre/post-composing it with braidings. In particular, a braid clique morphism is always fully determined by its leftmost component $f_{00} \colon R \otimes A_0 \otimes \ldots \otimes A_n \to R \otimes B_0 \otimes \ldots \otimes B_m$.

Lemma 2.11. *Let $(\mathscr{V},\mathscr{G})$ be a polygraph couple. There exists a premonoidal category, $\mathsf{Eff}(\mathscr{V},\mathscr{G})$, that has objects the braid cliques, $\mathrm{Braid}_R(A_0,\ldots,A_n)$, in $\mathsf{Mon}_{\mathsf{Run}}(\mathscr{V},\mathscr{G})$, and as morphisms the braid clique morphisms between them. See Appendix.*

Lemma 2.12. *Let $(\mathscr{V},\mathscr{G})$ be a polygraph couple. There exists an identity-on-objects functor $\mathsf{Mon}(\mathscr{V}) \to \mathsf{Eff}(\mathscr{V},\mathscr{G})$ that strictly preserves the premonoidal structure and whose image is central. See Appendix.*

Lemma 2.13. *Let $(\mathscr{V},\mathscr{G})$ be a polygraph couple and consider the effectful category determined by $\mathsf{Mon}(\mathscr{V}) \to \mathsf{Eff}(\mathscr{V},\mathscr{G})$. Let $\mathbb{V} \to \mathbb{C}$ be a strict effectful category endowed with a polygraph couple morphism $F \colon (\mathscr{V},\mathscr{G}) \to \mathscr{U}(\mathbb{V},\mathbb{C})$. There exists a unique strict effectful functor from $(\mathsf{Mon}(\mathscr{V}) \to \mathsf{Eff}(\mathscr{V},\mathscr{G}))$ to $(\mathbb{V} \to \mathbb{C})$ commuting with F as a polygraph couple morphism. See Appendix.*

Theorem 2.14 (Runtime as a resource)**.** *The free strict effectful category over a polygraph couple $(\mathscr{V},\mathscr{G})$ is $\mathsf{Mon}(\mathscr{V}) \to \mathsf{Eff}(\mathscr{V},\mathscr{G})$. Its morphisms $A \to B$ are in bijection with the morphisms $R \otimes A \to R \otimes B$ of the runtime monoidal category,*

$$\mathsf{Eff}(\mathscr{V},\mathscr{G})(A,B) \cong \mathsf{Mon}_{\mathsf{Run}}(\mathscr{V},\mathscr{G})(R \otimes A, R \otimes B).$$

Proof. We must first show that $\mathsf{Mon}(\mathscr{V}) \to \mathsf{Eff}(\mathscr{V},\mathscr{G})$ is an effectful category. The first step is to see that $\mathsf{Eff}(\mathscr{V},\mathscr{G})$ forms a premonoidal category (Lemma 2.11). We also know that $\mathsf{Mon}(\mathscr{V})$ is a monoidal category: in fact, a strict, freely generated one. There exists an identity on objects functor, $\mathsf{Mon}(\mathscr{V}) \to \mathsf{Eff}(\mathscr{V},\mathscr{G})$, that strictly preserves the premonoidal structure and centrality (Lemma 2.12).

Let us now show that it is the free one over the polygraph couple $(\mathscr{V},\mathscr{G})$. Let $\mathbb{V} \to \mathbb{C}$ be an effectful category, with an polygraph couple map $F \colon (\mathscr{V},\mathscr{G}) \to \mathscr{U}(\mathbb{V},\mathbb{C})$. We can construct a unique effectful functor from $(\mathsf{Mon}(\mathscr{V}) \to \mathsf{Eff}(\mathscr{V},\mathscr{G}))$ to $(\mathbb{V} \to \mathbb{C})$ giving its universal property (Lemma 2.13). □

Corollary 2.15 (String diagrams for effectful categories)**.** *We can use string diagrams for effectful categories, quotiented under the same isotopy as for monoidal categories, provided that we do represent the runtime as an extra wire that needs to be the input and output of every effectful morphism.*

3 Profunctors and Promonads

We have elaborated on string diagrams for effectful categories. Let us now show that effectful categories are fundamental objects. The profunctorial counterpart of a monad is a promonad. Promonads have been widely used for functional programming semantics, although usually with an extra assumption of strength and under the name of "arrows" [17, 19, 20]. Promonads over a category endow it with some new, "effectful", morphisms; while the base morphisms of the category are called the "pure" morphisms. This terminology will coincide when regarding effectful categories as promonads.

In this section, we introduce profunctors and promonads. In the following sections, we show that effectful categories are to promonads what monoidal categories are to categories: they are the pseudomonoids of a suitably constructed monoidal bicategory of promonads. In order to obtain this result, we introduce the pure tensor of promonads in Section 4. The pure tensor of promonads combines the effects of two promonads over different categories into a single one. In some sense, it does so in the universal way that turns "purity" into "centrality" (Theorem 4.2).

3.1 Profunctors: an algebra of processes

Profunctors $P\colon \mathbb{A}^{op} \times \mathbb{B} \to \mathsf{Set}$ [3, 7, 4] can be thought as indexing *families of processes* $P(A,B)$ by the types of an input channel A and an output channel B [10].

The category $\mathbb{A}$ has as morphisms the pure transformations $f\colon A' \to A$ that we can apply to the input of a process $p \in P(A,B)$ to obtain a new process, which we call $(f > p) \in P(A',B)$. Analogously, the category $\mathbb{B}$ has as morphisms the pure transformations $g\colon B \to B'$ that we can apply to the output of a process $p \in P(A,B)$ to obtain a new process, which we call $(p < g) \in P(A,B')$. The profunctor axioms encode the compositionality of these transformations.

Definition 3.1. A *profunctor* $(P,>,<)$ between two categories $\mathbb{A}$ and $\mathbb{B}$ is a family of sets $P(A,B)$ indexed by objects of $\mathbb{A}$ and $\mathbb{B}$, and endowed with jointly functorial left and right actions of the morphisms of $\mathbb{A}$ and $\mathbb{B}$, respectively. Explicitly, types of these actions are $(>)\colon \hom(A',A) \times P(A',B) \to P(A,B)$, and $(<)\colon \hom(B,B') \times P(A,B) \to P(A,B')$. They must satisfy

- compatibility, $(f > p) < g = f > (p < g)$,
- preserve identities, $\mathrm{id} > p = p$, and $p < \mathrm{id} = p$,
- and composition, $(p < f) < g = p < (f \,;\, g)$ and $f > (g > p) = (f \,;\, g) > p$.

More succinctly, a *profunctor* $P\colon \mathbb{A} \nrightarrow \mathbb{B}$ is a functor $P\colon \mathbb{A}^{op} \times \mathbb{B} \to \mathsf{Set}$. When presented as a family of sets with a pair of actions, profunctors are sometimes called *bimodules*.

A profunctor homomorphism $\alpha\colon P \to Q$ transforms processes of type $P(A,B)$ into processes of type $Q(A,B)$. The homomorphism affects only the effectful processes, and not the pure transformations we could apply in $\mathbb{A}$ and $\mathbb{B}$. This means that $\alpha(f > p) = f > \alpha(p)$ and that $\alpha(p < g) = \alpha(p) < g$.

Definition 3.2 (Profunctor homomorphism)**.** A *profunctor homomorphism* from the profunctor $P\colon \mathbb{A} \nrightarrow \mathbb{B}$ to the profunctor $Q\colon \mathbb{A} \nrightarrow \mathbb{B}$ is a family of functions $\alpha_{A,B}\colon P(A,B) \to Q(A,B)$ preserving the left and right actions, $\alpha(f > p < g) = f > \alpha(p) < g$. Equivalently, it is a natural transformation $\alpha\colon P \to Q$ between the two functors $\mathbb{A}^{op} \times \mathbb{B} \to \mathsf{Set}$.

How to compose two families of processes? Assume we have a process $p \in P(A,B_1)$ and a process $q \in Q(B_2,C)$. Moreover, assume we have a transformation $f\colon B_1 \to B_2$ translating from the output of the second to the input of the first. In this situation, we can plug together the processes: $p \in P(A,B_1)$ writes to an output of type B_1, which is translated by f to an input of type B_2, then used by $q \in Q(B_2,C)$.

There are two slightly different ways of describing this process, depending on whether we consider the translation to be part of the first or the second process. We could translate just after finishing the first process, $(p < f, q)$; or translate just before starting the second process, $(p, f > q)$.

These are two different pairs of processes, with different types. However, if we take the process interpretation seriously, it does not really matter when to apply the translation. These two descriptions represent the same process. They are *dinaturally equivalent* [10, 25].

Definition 3.3 (Dinatural equivalence)**.** Let $P\colon \mathbb{A} \nrightarrow \mathbb{B}$ and $Q\colon \mathbb{B} \nrightarrow \mathbb{C}$ be two profunctors. Consider the set of matching pairs of processes, with a given input A and output C,

$$R_{P,Q}(A,C) = \sum_{B\in\mathbb{B}} P(A,B) \times Q(B,C).$$

Dinatural equivalence $(\sim)$, on the set $R_{P,Q}(A,C)$ is the smallest equivalence relation satisfying $(p < g, q) \sim (p, g > q)$. The set of matching processes $R_{P,Q}(A,C)$ quotiented by dinaturality $(\sim)$ is written as $(P \diamond Q)(A,C)$. It is a particular form of colimit over the category $\mathbb{B}$, called a *coend*, usually denoted by an integral sign.

$$(P \diamond Q)(A,C) = R_{P,Q}(A,C)/(\sim) = \int^{B\in\mathbb{B}} P(A,B) \times Q(B,C).$$

Definition 3.4 (Profunctor composition)**.** The composition of two profunctors $P\colon \mathbb{A} \nrightarrow \mathbb{B}$ and $Q\colon \mathbb{B} \nrightarrow \mathbb{C}$ is the profunctor $(P \diamond Q)\colon \mathbb{A} \nrightarrow \mathbb{C}$ has as processes the matching pairs of processes in P and Q quotiented by dinaturality on $\mathbb{B}$,

$$(p, g < q) \sim (p > g, q).$$

Its actions are the left and right actions of p and q, respectively, $f > (p,q) < g = (f > p, q < g)$.

The identity profunctor $\mathbb{A}\colon \mathbb{A} \nrightarrow \mathbb{A}$ has as processes the morphisms of the category $\mathbb{A}$, it is given by the hom-sets. Its actions are pre and post-composition, $f > h < g = f \mathbin{⨟} h \mathbin{⨟} g$.

Profunctors are better understood as providing a double categorical structure to the category of categories. A double category $\mathbb{D}$ contains 0-cells (or "objects"), two different types of 1-cells (the "arrows" and the "proarrows"), and cells [37]. Arrows compose in an strictly associative and unital way, while proarrows come equipped with natural isomorphisms representing associativity and unitality. We employ the graphical calculus of double categories [29], with arrows going left to right and proarrows going top to bottom.

Definition 3.5. The double category of categories, **CAT**, has as objects the small categories $\mathbb{A}, \mathbb{B}, \dots$, as arrows the functors between them, $F\colon \mathbb{A} \to \mathbb{A}'$, as proarrows the profunctors between them, $P\colon \mathbb{A} \nrightarrow \mathbb{B}$, and as cells, the natural transformations, $\alpha_{A,B}\colon P(A,B) \to Q(FA,GB)$.

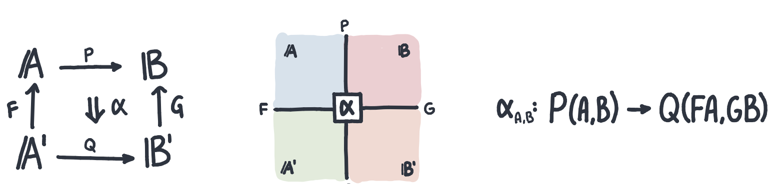


Figure 7: Cell in the double category of categories.

Every functor has a companion and a conjoint profunctors: their representable and corepresentable profunctors [14]. This structure makes **CAT** into the paradigmatic example of a proarrow equipment (or *framed bicategory* [37]).

3.2 Promonads: new morphisms for an old category

Promonads are to profunctors what monads are to functors.[2] It may be then surprising to see that so little attention has been devoted to them, relative to their functorial counterparts. The main source of examples and focus of attention has been the semantics of programming languages [19, 30, 20]. Strong monads are commonly used to give categorical semantics of effectful programs [28], and the so-called *arrows* (or *strong promonads*) strictly generalize them.

Part of the reason behind the relative unimportance given to promonads elsewhere may stem from the fact that promonads over a category can be shown in an elementary way to be equivalent to identity-on-objects functors from that category [25]. The explicit proof is, however, difficult to find in the literature, and so we include it here (Theorem 3.9).

Under this interpretation, promonads are new morphisms for an old category. We can reinterpret the old morphisms into the new ones in a functorial way. The paradigmatic example is again that of Kleisli or cokleisli categories of strong monads and comonads. This structure is richer than it may sound, and we will explore it further during the rest of this text.

Definition 3.6 (Monoids and promonoids)**.** A *monoid* in a double category is an arrow $T\colon \mathbb{A} \to \mathbb{A}$ together with cells $m \in \hom(M \otimes M; 1, 1; M)$ and $e \in \mathrm{cell}(1; 1, 1; M)$, called multiplication and unit, satisfying unitality and associativity. A *promonoid* in a double category is a proarrow $M\colon \mathbb{A} \nrightarrow \mathbb{A}$ together with cells $m \in \mathrm{cell}(1; M \otimes M, M, 1)$ and $e \in \mathrm{cell}(1; 1, M; 1)$, called promultiplication and prounit, satisfying unitality and associativity.

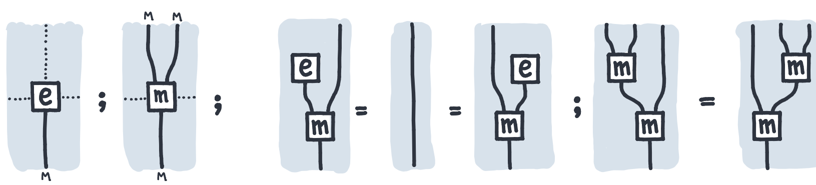


Figure 8: Data and axioms of a promonoid in a double category.

Dually, we can define *comonoids* and *procomonoids*.

A monad is a monoid in the category of categories, functors and profunctors **Cat**. In the same way, a promonad is a promonoid in **Cat**.

Definition 3.7. A *promonad* $(P, \star, ^\circ)$ over a category $\mathbb{C}$ is a profunctor $P\colon \mathbb{C} \nrightarrow \mathbb{C}$ together with natural transformations representing inclusion $(^\circ)_{X,Y}\colon \mathbb{C}(X,Y) \to P(X,Y)$ and multiplication $(\star)_{X,Y}\colon P(X,Y) \times P(Y,Z) \to P(X,Z)$, and such that

i. the right action is premultiplication, $f^\circ \star p = f > p$;
ii. the left action is posmultiplication, $p \star f^\circ = p < f$;
iii. multiplication is dinatural, $p \star (f > q) = (p < f) \star q$;
iv. and multiplication is associative, $(p_1 \star p_2) \star p_3 = p_1 \star (p_2 \star p_3)$.

Equivalently, promonads are promonoids in the double category of categories, where the dinatural multiplication represents a transformation from the composition of the profunctor P with itself.

Lemma 3.8 (Kleisli category of a promonad)**.** *Every promonad $(P, \star, ^\circ)$ induces a category with the same objects as its base category, but with hom-sets given by $P(\bullet,\bullet)$, composition given by $(\star)$ and identities given by (id°). This is called its* Kleisli category, $\mathrm{kleisli}(P)$. *Moreover, there exists an identity-on-objects functor $\mathbb{C} \to \mathrm{kleisli}(P)$, defined on morphisms by the unit of the promonad. See Appendix.*

[2]To quip, a promonad is just a monoid on the category of endoprofunctors.

The converse is also true: every category $\mathbb{C}$ with an identity-on-objects functor from some base category $\mathbb{V}$ arises as the Kleisli category of a promonad.

Theorem 3.9. *Promonads over a category $\mathbb{C}$ correspond to identity-on-objects functors from the category $\mathbb{C}$. Given any identity-on-objects functor $i\colon \mathbb{C} \to \mathbb{D}$ there exists a unique promonad over $\mathbb{C}$ having $\mathbb{D}$ as its Kleisli category: the promonad given by the profunctor $\hom_{\mathbb{D}}(i(\bullet), i(\bullet))$. See Appendix.*

3.3 Homomorphisms and transformations of promonads

We have characterized promonads as identity-on-objects functors. We now characterize the homomorphisms and transformations of promonads as suitable pairs of functors and natural transformations.

Definition 3.10 (Promonoid homomorphism)**.** Let $(\mathbb{A}, M, m, e)$ and $(\mathbb{B}, N, n, u)$ be promonoids in a double category. A promonoid homomorphism is an arrow $T\colon \mathbb{A} \to \mathbb{B}$ together with a cell $t \in \mathrm{cell}(F; M, N; F)$ that preserves the promonoid promultiplication and prounit.

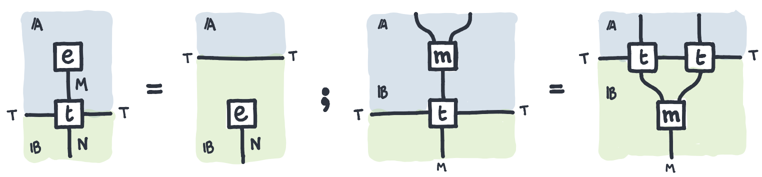


Figure 9: Axioms for a promonoid homomorphism.

Definition 3.11 (Promonad homomorphism)**.** Let $(\mathbb{A}, P, \star, ^\circ)$ and $(\mathbb{B}, Q, \star, ^\circ)$ be two promonads, possibly over two different categories. A promonad homomorphism (F_0, F) is a functor between the underlying categories $F_0\colon \mathbb{A} \to \mathbb{B}$ and a natural transformation $F_{X,Y}\colon P(X,Y) \to Q(FX, FY)$ preserving composition and inclusions. That is, $F(p_1 \star p_2) = F(p_1) \star F(p_2)$, and $F(f^\circ) = F_0(f)^\circ$.

Proposition 3.12. *A promonad homomorphism between two promonads understood as identity-on-objects functors, $\mathbb{V} \to \mathbb{C}$ and $\mathbb{W} \to \mathbb{D}$, is equivalently a pair of functors (F_0, F) that commute strictly with the two identity-on-objects functors on objects $F_0(X) = F(X)$ and morphisms $F_0(f)^\circ = F(f^\circ)$. See Appendix.*

Definition 3.13 (Promonoid modification)**.** Let $(\mathbb{A}, M, m, e)$ and $(\mathbb{B}, N, n, u)$ be promonoids in a double category, and let $t \in \mathrm{cell}(F; M, N; F)$ and $r \in \mathrm{cell}(G; M, N; G)$ be promonoid homomorphisms. A *promonoid modification* is a cell $\alpha \in \mathrm{cell}(F; 1, 1; G)$ such that its precomposition with t is its postcomposition with r.

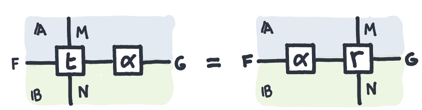


Figure 10: Axiom for a promonoid transformation.

Definition 3.14. A *promonad modification* between two promonad homomorphisms (F_0, F) and (G_0, G) between the same promonads $(\mathbb{A}, P, \star, ^\circ)$ and $(\mathbb{B}, Q, \star, ^\circ)$ is a natural transformation $\alpha_X\colon F_0(X) \to G_0(X)$ such that $\alpha_X > G(p) = F(p) < \alpha_Y$ for each $p \in P(X,Y)$.

Proposition 3.15. *A promonad modification between two promonad homomorphisms understood as commutative squares of identity-on-objects functors $F_0(f)^\circ = F(f^\circ)$ and $G_0(f)^\circ = G(f^\circ)$ is a natural transformation $\alpha\colon F_0 \Rightarrow G_0$ that can be lifted via the identity-on-objects functor to a natural transformation $\alpha^\circ\colon F \Rightarrow G$. In other words, a pure natural transformation.*

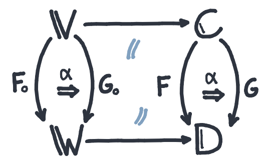


Figure 11: Promonad modifications are cylinder transformations.

Summarizing this section, we have shown a correspondence between promonads, their homomorphisms and modifications, and identity-on-objects functors, squares and cylinder transformations of squares. The double category structure allows us to talk about homomorphisms and modifications, which would be more difficult to address in a bicategory structure.

Promonad	Identity-on-objects functor	Theorem 3.9
Promonad homomorphism	Commuting square	Proposition 3.12
Promonad modification	Cylinder transformation	Proposition 3.15

4 Pure Tensor of Promonads

This section introduces the *pure tensor of promonads*. The pure tensor of promonads combines the effects of two promonads, possibly over different categories, into the effects of a single promonad over the product category. Effects do not generally interchange. However, this does not mean that no morphisms should interchange in the pure tensor of promonads: in our interpretation of a promonad $\mathbb{V} \to \mathbb{C}$, the morphisms coming from the inclusion are *pure*, they produce no effects; pure morphisms with no effects should always interchange with effectful morphisms, even if effectful morphisms do not interchange among themselves.

A practical way to encode and to remember all of the these restrictions is to use monoidal string diagrams. This is another application of the idea of runtime: we introduce an extra wire so that all the rules of interchange become ordinary interchange laws in a monoidal category. That is, we insist again that effectful morphisms are just pure morphisms using a shared resource – the runtime. When we compute the pure tensor of two promonads, the runtime needs to be shared between the impure morphisms of both promonads.

4.1 Pure tensor, via runtime

Definition 4.1 (Pure tensor)**.** Let $\mathbb{C}\colon \mathbb{V} \nrightarrow \mathbb{V}$ and $\mathbb{D}\colon \mathbb{W} \nrightarrow \mathbb{W}$ be two promonads. Their *pure tensor*, $\mathbb{C} * \mathbb{D}\colon \mathbb{V} \times \mathbb{W} \to \mathbb{V} \times \mathbb{W}$, is a promonad over $\mathbb{V} \times \mathbb{W}$ where elements of $\mathbb{C} * \mathbb{D}(X,Y;X',Y')$, the morphisms $X \otimes R \otimes Y \to X' \otimes R \otimes Y'$ in the freely presented monoidal category generated by the elements of Figure 12 and quotiented by the axioms of Figure 13.

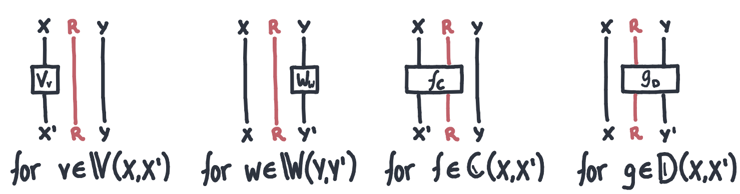


Figure 12: Generators for the elements of the pure tensor of promonads.

Multiplication is defined by composition in the monoidal category, and the unit is defined by the inclusion of pairs, as depicted in Figure 14.

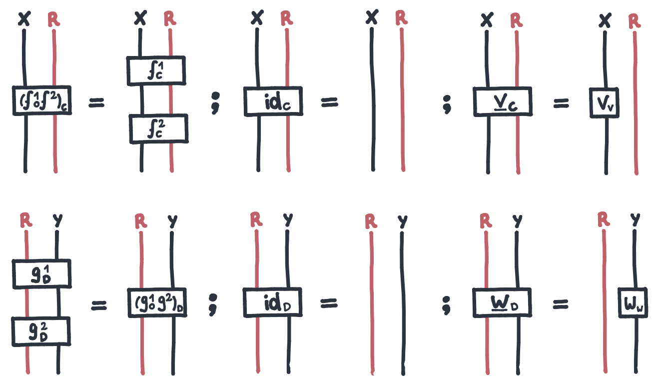


Figure 13: Axioms for the elements of the pure tensor of promonads.

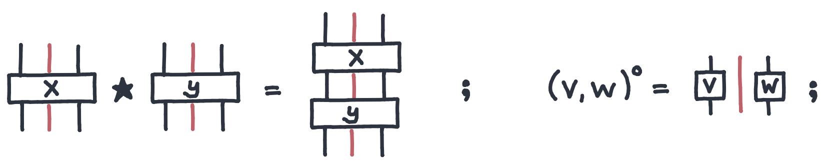


Figure 14: The pure tensor promonad.

In other words, the elements of the pure tensor are the morphisms the category presented by the graph that has as objects the pairs of objects (X,Y) with $X \in \mathbb{V}_{\text{obj}}$ and $Y \in \mathbb{W}_{\text{obj}}$, formally written as $X \otimes R \otimes Y$; and the morphisms generated by

- an edge $f_{\mathbb{C}} \colon X \otimes R \otimes Y \to X' \otimes R \otimes Y$ for each arrow $f \in \mathbb{C}(X,X')$ and each object $Y \in \mathbb{W}$;
- an edge $g_{\mathbb{D}} \colon X \otimes R \otimes Y \to X \otimes R \otimes Y'$ for each arrow $g \in \mathbb{D}(Y,Y')$ and each object $X \in \mathbb{V}$;
- an edge $v_{\mathbb{V}} \colon X \otimes R \otimes Y \to X' \otimes R \otimes Y$ for each arrow $v \in \mathbb{V}(X,X')$ and each object $Y \in \mathbb{W}$;
- and an edge $w_{\mathbb{W}} \colon X \otimes R \otimes Y \to X \otimes R \otimes Y'$ for each arrow $w \in \mathbb{W}(Y,Y')$ and each object $X \in \mathbb{V}$;

quotiented by centrality of pure morphisms: $f_{\mathbb{C}} ⨟ w_{\mathbb{W}} = w_{\mathbb{W}} ⨟ f_{\mathbb{C}}$ and $g_{\mathbb{D}} ⨟ v_{\mathbb{V}} = v_{\mathbb{V}} ⨟ g_{\mathbb{D}}$; by compositions and identities of one promonad: $f_{\mathbb{C}} ⨟ f'_{\mathbb{C}} = (f \star f')_{\mathbb{C}}$ and $\mathrm{id}_{\mathbb{C}} = \mathrm{id}$; by compositions and identities of the other promonad: $g_{\mathbb{D}} ⨟ g'_{\mathbb{D}} = (g \star g')_{\mathbb{D}}$ and $\mathrm{id}_{\mathbb{D}} = \mathrm{id}$; and by the coincidence of pure morphisms and their effectful representatives: $v_{\mathbb{V}} = v^{\circ}_{\mathbb{C}}$ and $w_{\mathbb{W}} = w^{\circ}_{\mathbb{D}}$.

Crucially in this definition, $f_{\mathbb{C}}$ and $g_{\mathbb{D}}$ do not interchange: they are sharing the runtime, and that prevents the application of the interchange law. The pure tensor of promonads, $\mathbb{C} * \mathbb{D}$, takes its name from the fact that, if we interpret the promonads $\mathbb{V} \to \mathbb{C}$ and $\mathbb{W} \to \mathbb{D}$ as declaring the morphisms in $\mathbb{V}$ and $\mathbb{W}$ as pure, then the pure morphisms of the composition interchange with all effectful morphisms. The spirit is similar to the *free product of groups with commuting subgroups* [26].

4.2 Universal property of the pure tensor

There are multiple canonical ways in which one could combine the effects of two promonads, $\mathbb{C} \colon \mathbb{V} \nrightarrow \mathbb{V}$ and $\mathbb{D} \colon \mathbb{W} \nrightarrow \mathbb{W}$, into a single promonad, such as taking the product of both, $\mathbb{C} \times \mathbb{D} \colon \mathbb{V} \times \mathbb{W} \nrightarrow \mathbb{V} \times \mathbb{W}$. Let us show that the pure tensor has a universal property: it is the universal one in which we can include impure morphisms from each promonads, interchanging with pure morphisms from the other promonad, so that purity is preserved.

Theorem 4.2. *Let $\mathbb{C}\colon \mathbb{V} \nrightarrow \mathbb{V}$ and $\mathbb{D}\colon \mathbb{W} \nrightarrow \mathbb{W}$ be two promonads and let $\mathbb{C} * \mathbb{D}\colon \mathbb{V}\times\mathbb{W} \to \mathbb{V}\times\mathbb{W}$ be their pure tensor. There exist a pair of promonad homomorphisms $L\colon \mathbb{C}\times\mathbb{W} \to \mathbb{C}*\mathbb{D}$ and $R\colon \mathbb{V}\times\mathbb{D} \to \mathbb{C}*\mathbb{D}$. These are universal in the sense that, for every pair of promonad homomorphisms, $A\colon \mathbb{C}\times\mathbb{W} \to \mathbb{E}$ and $B\colon \mathbb{V}\times\mathbb{D} \to \mathbb{E}$, there exists a unique promonad homomorphism $(A\vee B)\colon \mathbb{C}*\mathbb{D} \to \mathbb{E}$ that commutes strictly with them, $(A\vee B) \mathbin{⨾} L = A$ and $(A\vee B) \mathbin{⨾} R = B$. See Appendix.*

5 Effectful Categories are Pseudomonoids

We will now use the pure tensor of promonads to justify effectful categories as the promonadic counterpart of monoidal categories: effectful categories are pseudomonoids in the monoidal bicategory of promonads with the pure tensor. Pseudomonoids [9, 43] are the categorification of monoids. They are still formed by a 0-cell representing the carrier of the monoid and a pair of 1-cells representing multiplication and units. However, we weaken the requirement for associativity and unitality to the existence of invertible 2-cells, called the *associator* and *unitor*.

In the same way that monoids live in monoidal categories, pseudomonoids live in monoidal bicategories. A monoidal bicategory $\mathbb{A}$ is a bicategory in which we can tensor objects with a pseudofunctor $(\boxtimes)\colon \mathbb{A}\times\mathbb{A} \to \mathbb{A}$ and we have a tensor unit $I\colon 1 \to \mathbb{A}$, these are associative and unital up to equivalence, and satisfy certain coherence equations up to invertible modification [36].

5.1 Pseudomonoids

Definition 5.1. In a monoidal bicategory, a *pseudomonoid* over a 0-cell M is a pair of 1-cells, $M\boxtimes M \to M$ and $I \to M$, together with the following triple of invertible 2-cells representing associativity and unitality (Figure 15), and satisfying the pentagon and triangle equations (see Appendix).A homomorphism of pseudomonoids is given by a 1-cell between their underlying 0-cells and the following invertible 2-cells, representing preservation of the multiplication and the unit (Figure 15), and satisfying compatibility with associativity and unitality (see Appendix).

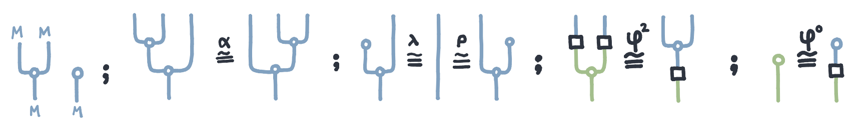


Figure 15: Data for a pseudomonoid and pseudomonoid homomorphism.

A pseudomonoid is *strict* when the associators and unitors are identity cells. Note that, in strict 2-categories (sometimes called 2-categories, in contrast to bicategories), this is the same as a monoid in the monoidal category that we obtain by ignoring the 2-cells.

Remark 5.2. A pseudomonoid in the monoidal bicategory of categories with the cartesian product of categories, $(\mathbf{Cat},\times)$ is a monoidal category. A strict pseudomonoid in the same monoidal bicategory is a strict monoidal category.

A strict pseudomonoid in the monoidal bicategory of categories with the funny tensor product of categories $(\mathbf{Cat},\square)$ is a strict premonoidal category. However, it is not immediately clear how to recover premonoidal categories as pseudomonoids. A naive attempt will fail: $(\mathbf{Cat},\square)$ is usually made into a monoidal bicategory with non-necessarily-natural transformations, but we do want our coherence morphisms to be natural, so we must ask at least naturality. This will not be enough: taking natural transformations as 2-cells will give us premonoidal categories where the associators and unitors do not need to be *central*. Centrality is what requires a more careful approach.

5.2 Effectful categories are promonad pseudomonoids

Promonads form a monoidal category with the pure tensor product and moreover a strict monoidal bicategory with promonad modifications. Effectful categories are the pseudomonoids in this category.

Theorem 5.3. *An effectful category (or monoidal Freyd category) is a pseudomonoid on the monoidal 2-category of promonads with promonad homomorphism, promonad transformations and the pure tensor of promonads. A pseudomonoid homomorphism between effectful categories is an effectful functor.*

As a consequence, preomonoidal categories with their centre *are pseudomonoids. See Appendix.*

6 Conclusions

Premonoidal categories are monoidal categories with runtime, and we can stil use monoidal string diagrams and unrestricted topological deformations to reason about them. Instead of dealing directly with premonoidal categories, we employ the better behaved notion of non-cartesian Freyd categories, effectful categories. There exists a more fine-grained notion of "Cartesian effect category" [12], which generalizes Freyd categories and justifies calling "effectful category" to the general case.

Promonads have been arguably under-appreciated, possibly because of their characterization as "just" identity-on-objects functors. However, speaking of promonads as the proarrow counterpart of monads makes many aspects of the theory of monads clearer: every monad and every comonad induce a promonad (their Kleisli category) via the proarrow equipment, monad morphisms lift to promonad morphisms, distributive laws of monads induce a way of composing morphisms from different kleisli categories [8]. Justifying effectful categories in terms of promonads highlights their importance as the monadic counterpart of monoidal categories.

Ultimately, this is a first step towards our more ambitious project of presenting the categorical structure of programming languages in a purely diagrammatic way, revisiting Alan Jeffrey's work [22, 21, 35]. The internal language of premonoidal categories and effectful categories is given by the *arrow do-notation* [30]; at the same time, we have shown that it is given by suitable string diagrams. This correspondence allows us to translate between programs and string diagrams (Figure 16).

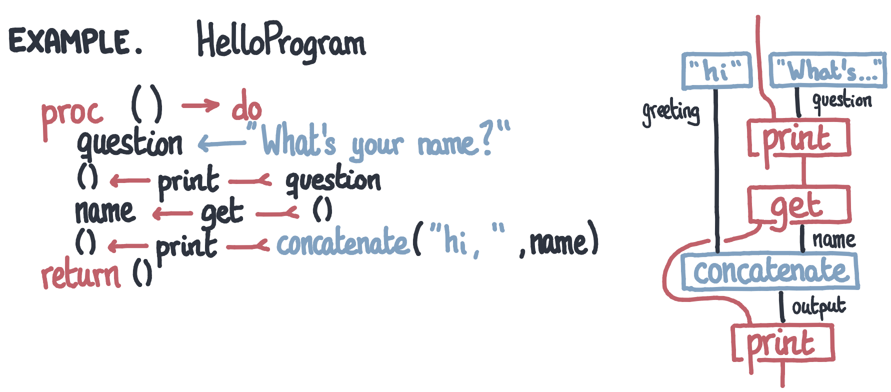


Figure 16: Premonoidal program in arrow do-notation and string diagrams.

Related work. Staton and Møgelberg [27] propose a formalization of Jeffrey's graphical calculus for effectful categories that arise as the Kleisli category of a strong monad. They prove that *'every strong monad is a linear-use state monad'*, that is, a state monad of the form $R \multimap !(\bullet) \otimes R$, where the state R, is an object that cannot be copied nor discarded.

7 Acknowledgements

The author wants to thank Tarmo Uustalu, Sam Staton, Niels Voorneveld and Paweł Sobociński for many helpful comments and discussion. The author gratefully thanks the anonymous reviewers at ACT'22 for many constructive suggestions that improved this manuscript. Mario Román was supported by the European Union through the ESF funded Estonian IT Academy research measure (2014-2020.4.05.19-0001); this work was also supported by the Estonian Research Council grant PRG1210.

References

[1] Samson Abramsky & Bob Coecke (2009): *Categorical Quantum Mechanics*. In Kurt Engesser, Dov M. Gabbay & Daniel Lehmann, editors: *Handbook of Quantum Logic and Quantum Structures*, Elsevier, Amsterdam, pp. 261–323, doi:10.1016/B978-0-444-52869-8.50010-4.

[2] Samson Abramsky, Esfandiar Haghverdi & Philip J. Scott (2002): *Geometry of Interaction and Linear Combinatory Algebras*. *Math. Struct. Comput. Sci.* 12(5), pp. 625–665, doi:10.1017/S0960129502003730.

[3] Jean Bénabou (1967): *Introduction to bicategories*. In: *Reports of the Midwest Category Seminar*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–77, doi:10.1007/BFb0074299.

[4] Jean Bénabou (2000): *Distributors at work*. *Lecture notes written by Thomas Streicher* 11.

[5] R.F. Blute, J.R.B. Cockett, R.A.G. Seely & T.H. Trimble (1996): *Natural deduction and coherence for weakly distributive categories*. *Journal of Pure and Applied Algebra* 113(3), pp. 229–296, doi:10.1016/0022-4049(95)00159-X.

[6] Filippo Bonchi, Jens Seeber & Pawel Sobocinski (2018): *Graphical Conjunctive Queries*. In Dan R. Ghica & Achim Jung, editors: *27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4-7, 2018, Birmingham, UK*, *LIPIcs* 119, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 13:1–13:23, doi:10.4230/LIPIcs.CSL.2018.13.

[7] Francis Borceux (1994): *Handbook of Categorical Algebra*. *Encyclopedia of Mathematics and its Applications* 1, Cambridge University Press, doi:10.1017/CBO9780511525858.

[8] Eugenia Cheng (2021): *Distributive Laws for Lawvere Theories (Invited Talk)*. In Fabio Gadducci & Alexandra Silva, editors: *9th Conference on Algebra and Coalgebra in Computer Science, CALCO 2021, August 31 to September 3, 2021, Salzburg, Austria*, *LIPIcs* 211, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 1:1–1:1, doi:10.4230/LIPIcs.CALCO.2021.1.

[9] Brian Day & Ross Street (1997): *Monoidal Bicategories and Hopf Algebroids*. *Advances in Mathematics* 129(1), pp. 99–157, doi:10.1006/aima.1997.1649.

[10] Elena Di Lavore, Giovanni de Felice & Mario Román (2022): *Monoidal Streams for Dataflow Programming*. In: *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '22, Association for Computing Machinery, New York, NY, USA, doi:10.1145/3531130.3533365.

[11] Vladimir Drinfeld, Shlomo Gelaki, Dmitri Nikshych & Victor Ostrik (2010): *On braided fusion categories I*. *Selecta Mathematica* 16(1), pp. 1–119, doi:10.1007/s00029-010-0017-z.

[12] Jean-Guillaume Dumas, Dominique Duval & Jean-Claude Reynaud (2011): *Cartesian effect categories are Freyd-categories*. *Journal of Symbolic Computation* 46(3), pp. 272–293, doi:10.1016/j.jsc.2010.09.008.

[13] Jean-Yves Girard (1989): *Geometry of Interaction 1: Interpretation of System F*. In R. Ferro, C. Bonotto, S. Valentini & A. Zanardo, editors: *Logic Colloquium '88*, *Studies in Logic and the Foundations of Mathematics* 127, Elsevier, pp. 221–260, doi:10.1016/S0049-237X(08)70271-4.

[14] Marco Grandis & Robert Paré (1999): *Limits in double categories*. *Cahiers de topologie et géométrie différentielle catégoriques* 40(3), pp. 162–220. Available at `http://www.numdam.org/item/CTGDC_1999__40_3_162_0/`.

[15] René Guitart (1980): *Tenseurs et machines*. *Cahiers de topologie et géométrie différentielle catégoriques* 21(1), pp. 5–62. Available at `http://www.numdam.org/item/CTGDC_1980__21_1_5_0/`.

[16] Jules Hedges (2019): *Folklore: Monoidal kleisli categories*. Available at `https://julesh.com/2019/04/18/folklore-monoidal-kleisli-categories/`.

[17] Chris Heunen & Bart Jacobs (2006): *Arrows, like Monads, are Monoids*. In Stephen D. Brookes & Michael W. Mislove, editors: *Proceedings of the 22nd Annual Conference on Mathematical Foundations of Programming Semantics, MFPS 2006, Genova, Italy, May 23-27, 2006*, *Electronic Notes in Theoretical Computer Science* 158, Elsevier, pp. 219–236, doi:10.1016/j.entcs.2006.04.012.

[18] Naohiko Hoshino, Koko Muroya & Ichiro Hasuo (2014): *Memoryful geometry of interaction: from coalgebraic components to algebraic effects*. In Thomas A. Henzinger & Dale Miller, editors: *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, ACM, pp. 52:1–52:10, doi:10.1145/2603088.2603124.

[19] John Hughes (2000): *Generalising monads to arrows*. *Science of Computer Programming* 37(1-3), pp. 67–111, doi:10.1016/S0167-6423(99)00023-4.

[20] Bart Jacobs, Chris Heunen & Ichiro Hasuo (2009): *Categorical semantics for arrows*. *J. Funct. Program.* 19(3-4), pp. 403–438, doi:10.1017/S0956796809007308.

[21] Alan Jeffrey (1997): *Premonoidal categories and a graphical view of programs*. *Preprint at ResearchGate*. Available at `https://www.researchgate.net/profile/Alan-Jeffrey/publication/228639836_Premonoidal_categories_and_a_graphical_view_of_programs/links/00b495182cd648a874000000/Premonoidal-categories-and-a-graphical-view-of-programs.pdf`.

[22] Alan Jeffrey (1997): *Premonoidal categories and flow graphs*. *Electron. Notes Theor. Comput. Sci.* 10, p. 51, doi:10.1016/S1571-0661(05)80688-7.

[23] André Joyal & Ross Street (1991): *The geometry of tensor calculus, I*. *Advances in Mathematics* 88(1), pp. 55–112, doi:10.1016/0001-8708(91)90003-P.

[24] Paul Blain Levy (2022): *Call-by-Push-Value*. *ACM SIGLOG News* 9(2), p. 7–29, doi:10.1145/3537668.3537670.

[25] Fosco Loregian (2021): *(Co)end Calculus*. London Mathematical Society Lecture Note Series, Cambridge University Press, doi:10.1017/9781108778657.

[26] Wilhelm Magnus, Abraham Karrass & Donald Solitar (2004): *Combinatorial group theory: Presentations of groups in terms of generators and relations*. Courier Corporation.

[27] Rasmus Ejlers Møgelberg & Sam Staton (2014): *Linear usage of state*. *Log. Methods Comput. Sci.* 10(1), doi:10.2168/LMCS-10(1:17)2014.

[28] Eugenio Moggi (1991): *Notions of Computation and Monads*. *Inf. Comput.* 93(1), pp. 55–92, doi:10.1016/0890-5401(91)90052-4.

[29] David Jaz Myers (2016): *String Diagrams For Double Categories and Equipments*, doi:10.48550/ARXIV.1612.02762.

[30] Ross Paterson (2001): *A New Notation for Arrows*. In Benjamin C. Pierce, editor: *Proceedings of the Sixth ACM SIGPLAN International Conference on Functional Programming (ICFP '01), Firenze (Florence), Italy, September 3-5, 2001*, ACM, pp. 229–240, doi:10.1145/507635.507664.

[31] Dusko Pavlovic (2013): *Monoidal computer I: Basic computability by string diagrams*. *Inf. Comput.* 226, pp. 94–116, doi:10.1016/j.ic.2013.03.007.

[32] John Power (2002): *Premonoidal categories as categories with algebraic structure*. *Theor. Comput. Sci.* 278(1-2), pp. 303–321, doi:10.1016/S0304-3975(00)00340-6.

[33] John Power & Edmund Robinson (1997): *Premonoidal Categories and Notions of Computation*. *Math. Struct. Comput. Sci.* 7(5), pp. 453–468, doi:10.1017/S0960129597002375.

[34] John Power & Hayo Thielecke (1999): *Closed Freyd- and kappa-categories*. In Jirí Wiedermann, Peter van Emde Boas & Mogens Nielsen, editors: *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, *Lecture Notes in Computer Science* 1644, Springer, pp. 625–634, doi:10.1007/3-540-48523-6_59.

[35] Mario Román (2022): *Notes on Jeffrey's A Graphical View of Programs*. Available at `https://www.ioc.ee/~mroman/data/talks/premonoidalgraphicalview.pdf`.

[36] Christopher J. Schommer-Pries (2011): *The Classification of Two-Dimensional Extended Topological Field Theories*. arXiv:1112.1000.

[37] Michael Shulman (2008): *Framed Bicategories and Monoidal Fibrations*. *Theory and Applications of Categories* 20(18), pp. 650–738, doi:10.48550/arxiv.0706.1286.

[38] Michael Shulman (2016): *Categorical logic from a categorical point of view*. *Personal webpage*. Available at https://mikeshulman.github.io/catlog/catlog.pdf.

[39] Michael Shulman (2018): *The 2-Chu-Dialectica construction and the polycategory of multivariable adjunctions*. *arXiv preprint arXiv:1806.06082*, doi:10.48550/arxiv.1806.06082.

[40] Sam Staton & Paul Blain Levy (2013): *Universal properties of impure programming languages*. In Roberto Giacobazzi & Radhia Cousot, editors: *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*, ACM, pp. 179–192, doi:10.1145/2429069.2429091.

[41] Todd Trimble (2010): *Coherence Theorem for Monoidal Categories (nLab entry), Section 3. Discussion*. https://ncatlab.org/nlab/show/coherence+theorem+for+monoidal+categories, Last accessed on 2022-05-10.

[42] Tarmo Uustalu & Varmo Vene (2008): *Comonadic Notions of Computation*. In Jiří Adámek & Clemens Kupke, editors: *Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science, CMCS 2008, Budapest, Hungary, April 4-6, 2008*, *Electronic Notes in Theoretical Computer Science* 203, Elsevier, pp. 263–284, doi:10.1016/j.entcs.2008.05.029.

[43] Dominic Verdon (2017): *Coherence for braided and symmetric pseudomonoids*. *CoRR* abs/1705.09354, doi:10.48550/arxiv.1705.09354. arXiv:1705.09354.

String Diagrams for Layered Explanations

Leo Lobski Fabio Zanasi
University College London

We propose a categorical framework to reason about scientific *explanations*: descriptions of a phenomenon meant to translate it into simpler terms, or into a context that has been already understood. Our motivating examples come from systems biology, electrical circuit theory, and concurrency. We demonstrate how three explanatory models in these seemingly diverse areas can be all understood uniformly via a graphical calculus of *layered props*. Layered props allow for a compact visual presentation of the same phenomenon at different levels of precision, as well as the translation between these levels. Notably, our approach allows for *partial explanations*, that is, for translating just one part of a diagram while keeping the rest of the diagram untouched. Furthermore, our approach paves the way for formal reasoning about counterfactual models in systems biology.

1 Introduction

Different fields of science and engineering come with their own notions and traditions of explaining one phenomenon in terms of another one. For example, statistical mechanics explains thermodynamics, since it relies on fewer assumptions, which are moreover perceived as more fundamental than those of thermodynamics. A similar pattern may be found in the reduction of climate science to various areas of physics and biology. The converse move, from a "lower" to a "higher" level, is also interesting: for instance, temperature and vessel shape may be used to explain crystallisation. Choosing the right level of abstraction is paramount for successful communication between different disciplines, as well as between the scientific community and the general public. In particular, the definition of what constitutes an explanation is an increasingly important topic in the areas of automated reasoning and artificial intelligence [19].

Perhaps the most drastic divide between different modes of explaining can be found in biology, where some phenomena are explained *mechanistically* (or *reductively*), that is, by reducing them to the underlying chemical or physical laws, while others are explained *functionally*, that is, by appealing to what an organism does as a part of a larger whole [10, 21]. For instance, when explaining production of ATP within a cell, the mitochondria can either be introduced as elementary blocks providing energy to the cell (functional), or as compartments containing a whole pathway to process ATP (mechanistic). This divide is not merely of conceptual interest, but has practical implications for the modelling of biological systems: the ability to replicate biological functions is taken as a measure of success of the rule based models [8, 10]. However, the existing rule based languages that model molecular interactions are typically not able to formally distinguish between mechanistic and functional rules, as these exist at different levels of abstraction [10].

The goal of this work is to identify fundamental mathematical structures underlying explanations across different fields of science. Upon these structures, we develop a formalism that is able to describe the different levels of abstractions involved in an explanation, and account for more elaborate aspects such as the divide above. Additionally, we attempt to provide a uniform framework for *counterfactual reasoning* by allowing explanations that depend on what could potentially occur. Ability to model counterfactual dependencies is of interest in rule-based models of molecular interactions [13]. We shall

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 362–382, doi:10.4204/EPTCS.380.21

illustrate our approach by showing how it models case studies in diverse scientific areas.[1]

In our framework, explanations always concern a certain *process*. The process can be thought of as an actually occurring natural phenomenon, or a computation, or a rule in some formal system. An explanation should then consist of another process whose level of abstraction is strictly lower than that of the process being explained. In addition to the lower level process, an explanation should state in what way the two processes are related, for example by giving a translation from one to the other. Moreover, we want the explanations to be *modular* or *compositional*, in the sense that the same explanation may be reused multiple times in case different systems have equivalent subsystems, and that the explanations can be composed to create larger, more complicated explanations. The reason for requiring modularity is twofold. First, it allows explanations to be reused by potentially different areas, in much the same way lemmas and theorems in mathematics are used to develop different theories. Second, this allows for a certain efficiency, as we may be interested in explaining only a part of a large system; in such a case modularity allows us to focus on this one part only, instead of explaining the whole system.

The above requirements for what an explanation should be like lead naturally to *monoidal categories*, as these allow for both sequential and parallel composition of processes (i.e. morphisms in a category). We assume that the monoidal categories are partially ordered "by abstraction", so that more abstract theories (i.e. categories) are higher in the order. We want to be able to compose not just the processes but also the explanations, so that we require the categories and functors under consideration to have a monoidal structure. We thus arrive at a 2-category which is able to simultaneously talk about processes in all the individual categories (0-cells), translations between processes (1-cells), compositions of the processes and the translations, as well as rules or equations between the processes and the translations (2-cells). The definitions of an explanation (Definitions 7 and 8) use all of this structure. This is the motivation for what we call a *layered prop* (Definition 2).

It is worth noting that, in the categorical approaches inspired by the paradigm of functorial semantics, an explanation and what is being explained live in two separate categories, with some translation between them expressed as a functor — see e.g. [1, 4, 3]. Within this perspective, some equality or relation in the domain is explained by passing to the codomain (or vice versa). Our framework allows to treat such situations in a single language, staying within one category. The main technical advantage of our approach is that partial interpretations are built into our language from the very beginning, potentially reducing the amount of computation that is needed. More conceptually, unlike in the functorial semantics approach, working in our framework allows for counterfactual reasoning: since we can mix-and-match categories and morphisms, this gives the flexibility to ask such questions as *What would happen if p did not occur?*

Our contributions are organised as follows. In Section 2 we define layered props and outline their connection to the so-called internal string diagram construction. Section 3 briefly outlines how a layered prop can be interpreted in the bicategory of pointed profunctors. We give three definitions of an explanation in Section 4: one applies to 1-cells, another one to 2-cells, and the last one formalises counterfactual explanations. The remaining sections contain case studies formalised in our framework. Section 5 shows an example involving biology and chemistry. Section 6 shows the explanation of electrical circuit behaviour in terms of signal flow graphs — as it draws from the circuit theory developed in [3, 4], this example also clarifies how our approach relates to the 'functorial semantics' approaches. Finally, Section 7 presents a case study from concurrency, involving the explanation of CCS expressions.

[1] On the other hand, we do not delve into the philosophical ramifications of our approach. Rather, the aim is to offer an abstract formalisation of of existing intuitions, thus potentially providing precise tools for debating what a scientific explanation *should* be.

2 Layered Props

We shall build our language on *string diagrammatic* syntax: the standard representation of morphisms in (strict) monoidal categories [22]. Algebraic reasoning on string diagrams is typically formulated using props (**pro**duct and **p**ermutation categories), which are just symmetric strict monoidal categories with the natural numbers as objects — see e.g. [15, 12, 24] for an overview. In fact, in order to model the different layers involved in an explanation, we will need a more sophisticated concept: *layered props*.

Since we want to talk about "string diagrams in context", the context being a theory at a particular level of abstraction, we draw the string diagrams inside a rectangle which represents its context. This allows us to reason both *internally* with the string diagrams, as well as *externally* by pasting and piling the rectangles. In order to formalise such graphical intuition as a layered prop, we need the preliminary notions of a *system of sets* and a *layered monoidal theory*.

We begin with systems of sets, which we think of as contexts and translations between them. Fix a collection of sets Ω. An Ω-*type* is a finite list of pairs $(\omega_1,\alpha_1;\dots;\omega_n,\alpha_n)$ where each ω_i is in Ω and each $\alpha_i \in \omega_i^*$ is an element in the free monoid on ω_i. Precisely, define Ω-types recursively as:

- the empty list ε is an Ω-type,
- if t is a type, $\omega \in \Omega$, and $\alpha \in \omega^*$, then $(t;\ \omega,\alpha)$ is an Ω-type.

We denote the collection of all Ω-types by $\mathtt{type}^{\Omega}$.

We call Ω a *system of sets* when it is equipped with a partial order, and, for each comparable pair $\omega \leq \omega'$, with a choice of a homomorphism $f : \omega'^* \to \omega^*$. Intuitively, as we think of the sets $\omega \in \Omega$ as contexts, the partial order is saying which contexts are more abstract, and the homomorphisms are translations from more abstract contexts to less abstract ones. We now introduce the counterpart of algebraic theories for monoidal categories (typically called monoidal theories, see e.g. [24]) based on this structure.

Definition 1 (Layered monoidal theory)**.** A *layered monoidal theory* is a tuple $(\Omega,\Sigma,\mathtt{ar},\mathtt{coar})$ consisting of a system of sets Ω, a set Σ (*signature*), and functions $\mathtt{ar},\mathtt{coar} : \Sigma \to \mathtt{type}^{\Omega}$.

It is convenient to introduce notation for the *internal signature* Σ^i, defined as

$$\Sigma^i := \{\sigma \in \Sigma : \text{ there are } \omega \in \Omega \text{ and } \alpha,\beta \in \omega^* \text{ s.t. } \mathtt{ar}(\sigma) = (\omega,\alpha) \text{ and } \mathtt{coar}(\sigma) = (\omega,\beta)\}.$$

The idea is that the generators in Σ^i are completely contained in a single context ω: there is no transition between contexts involved. We define the *terms* and the corresponding *sorts* (arity-coarity pairs of types $(t \mid s)$) of a layered monoidal theory by the recursive procedure in Figure 1. For the $\otimes_\omega$-rule, there is a side condition that only the rules for Σ^i, identity, composition and $\otimes_\omega$ are used in constructing the terms x and y. This ensures that x and y only contain generators from the internal signature, so that it makes sense to graphically represent the term $x \otimes_\omega y$ as juxtaposition of x and y inside the rectangle representing ω. We call the terms that are generated using only these four rules *internal*. If a layered monoidal theory is generated by monoidal categories (see Section 2.1 below), the internal terms will correspond precisely to morphisms inside the categories.

We think of the *pants* and the *copants* (line 3 of Figure 1) as composition and decomposition within a level of abstraction. The black and white triangles (line 4 of Figure 1) are translations between the levels: ◀ translates an abstract layer to a more concrete one (*refinement*), while ▷ maps towards a higher abstraction (*coarsening*). In the pointed profunctor semantics (Section 3), pants will be interpreted as the monoidal product (seen as a profunctor), and copants as its adjoint profunctor (cf. axioms in Figure 3). Likewise, refinement will be interpreted as a monoidal functor (seen as a profunctor), and coarsening as its adjoint (cf. axioms in Figure 4).

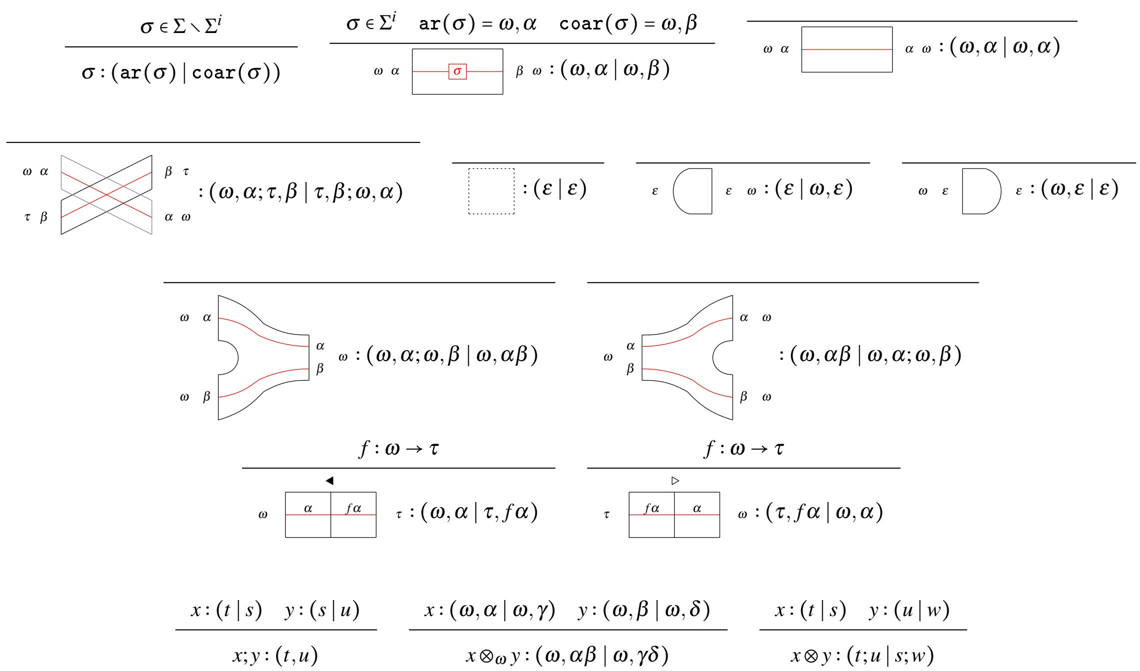
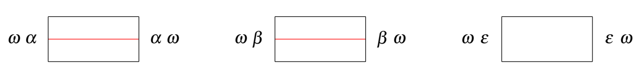


Figure 1: Recursive construction of the terms of a layered monoidal theory. Each term of the sort $(\omega,\alpha \mid \tau,\beta)$ is drawn as an area connecting the type ω,α on the left to the type τ,β on the right. The area inside a term, demarcated by black lines, is to be thought as representing the set ω, and an internal red wire as α (the element of ω^*). The change of type $\alpha \to \beta$ inside ω is drawn as a red box. The change of type at the level of sets $\omega \to \tau$ is drawn as a vertical black line.

In order to define a layered prop, we need to consider the terms modulo certain equations. Given $\omega \in \Omega$ and $\alpha,\beta \in \omega^*$, consider the internal terms with the sort $(\omega,\alpha \mid \omega,\beta)$. We may quotient this subset by the usual rules of monoidal categories: the identities and the monoidal unit are given by the third rule on the first line,



while the monoidal product $\otimes_\omega$ is represented by vertical juxtaposition inside the ω-rectangle. Further, we may quotient all the terms with the sort $(t \mid s)$ by the usual rules of symmetric monoidal categories: the identities are given by appropriate vertical juxtapositions of terms generated by the third rule on the first line, the monoidal unit is given by the second rule on the second line, and the monoidal product $\otimes$ is once again represented by vertical juxtaposition, this time of whole rectangles.

Definition 2 (Layered prop)**.** A *layered prop* generated by a layered monoidal theory $(\Omega,\Sigma,\mathtt{ar},\mathtt{coar})$ is a 2-category whose 0-cells are the types $\mathtt{type}^\Omega$ and whose 1-cells $t \to s$ are terms with sort $(t \mid s)$ quotiented by the laws of symmetric monoidal categories both internally and externally, as discussed above. The 2-cells are generated by the rules in Figures 2, 3 and 4. Where arrows are going in both directions, we require the 2-cells to be inverses. Further, we require the usual triangle identities to hold for each unit-counit pair in Figure 3, and the usual laws of monoidal categories to hold for the isomorphisms in Figure 4.

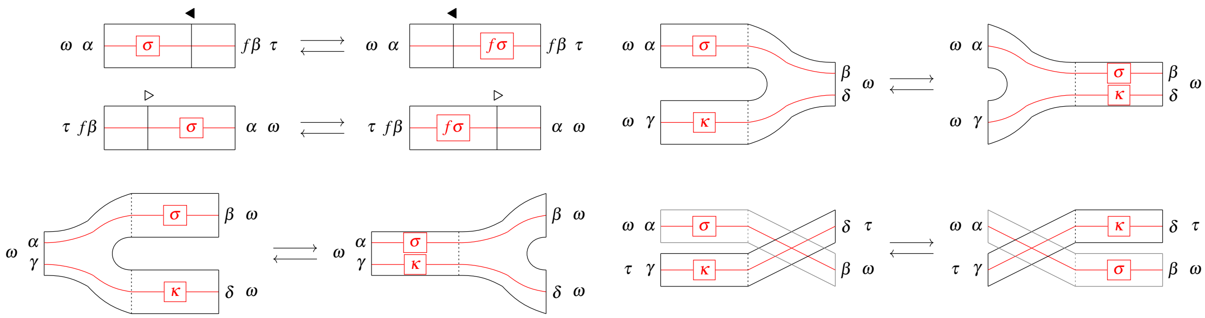


Figure 2: 2-cells of a layered prop expressing functoriality of refinement, coarsening, pants and copants.

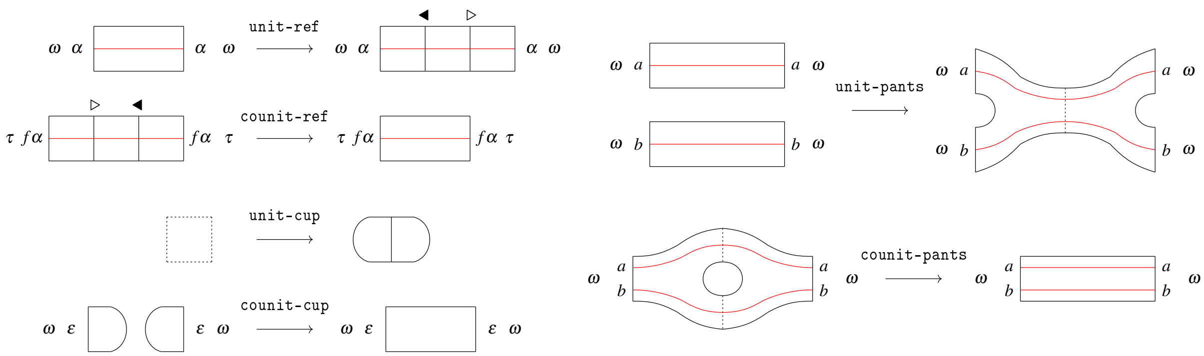


Figure 3: 2-cells of a layered prop that exhibit pants-copants and refinement-coarsening as two adjoint pairs.

Note that the 1-categorical structure of a layered prop can be seen as a generalisation of a coloured prop: any coloured prop gives rise to a layered prop with just one layer (i.e. with just one set in Ω). Furthermore, layered props are known in the literature as the *internal string diagram construction*. This was first introduced in the work of Bartlett, Douglas, Schommer-Pries and Vicary on topological quantum field theories [2]. The connection to profunctors is discussed by Hu [9].

2.1 Layered Props from Monoidal Categories

It is natural to build layered props from existing monoidal categories. In fact all the examples of layered props we consider arise in this way — see Sections 5-7 below. We assume that instead of a system of sets, we have a system of *monoidal categories* Ω with monoidal functors instead of homomorphisms. The construction of the layered prop $\mathcal{L}(\Omega)$ then proceeds as before, taking the internal signature to contain all morphisms in each category in Ω. We now proceed to define this formally.

A *system of monoidal categories* Ω is a subcategory of $\mathbf{Cat}$ such that

- every category $\omega \in \Omega$ is strict monoidal,
- every functor in Ω is strict monoidal,
- there is at most one functor between any pair of categories, that is, Ω is posetal.

The last condition is assumed merely for simplicity, we could construct a layered prop from Ω with multiple monoidal functors between a pair of monoidal categories. The formalism could be modified to incorporate non-strict monoidal categories, we leave this for future work.

We view a system of monoidal categories as a system of sets in a straightforward manner: the col-

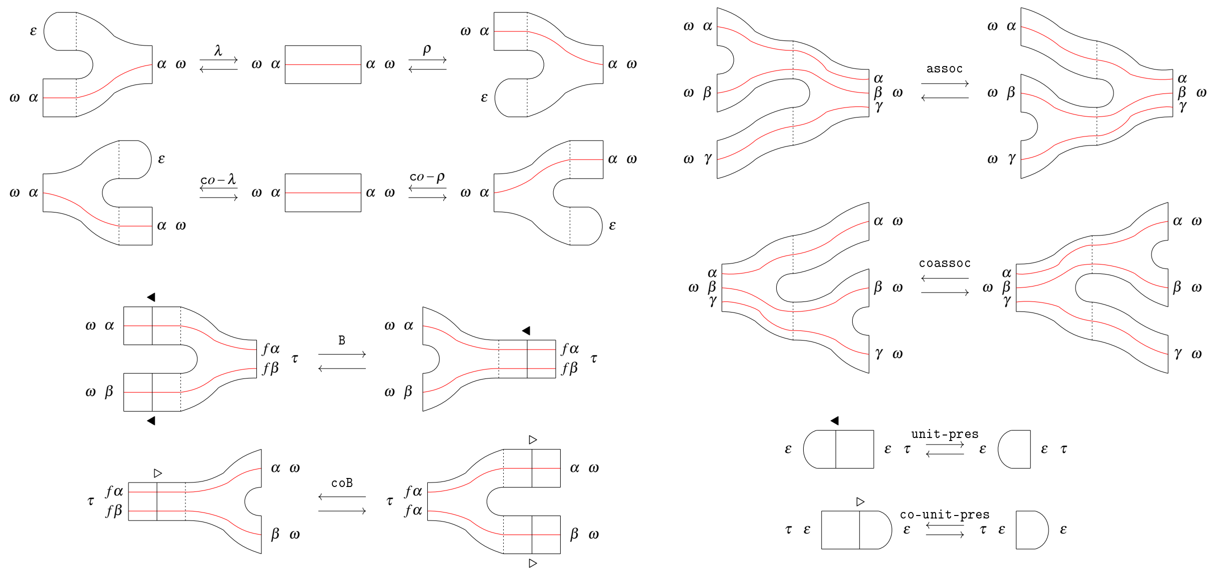


Figure 4: 2-cells of a layered prop that are motivated by monoidal categories and functors.

lection of sets is given by $\{\mathcal{O}b(\omega) : \omega \in \mathcal{O}b(\Omega)\}$, we identify $\alpha\beta \coloneqq \alpha \otimes \beta$ for all $\alpha, \beta \in \omega$ and $\omega \in \Omega$, we have $\omega \leq \omega'$ whenever there is a functor $f : \omega' \to \omega$, and the monoid homomorphisms are given by the restriction of each functor to objetcs.

Assuming that all the categories $\omega \in \Omega$ as well as Ω itself are small, we define the signature $\Sigma(\Omega)$ as follows:

$$\Sigma(\Omega) \coloneqq \left\{\sigma_{\omega}^{\alpha,\beta}\right\}_{\omega\in\mathcal{O}b(\Omega),\alpha,\beta\in\mathcal{O}b(\omega),\sigma:\alpha\to\beta}.$$

The arities and coarities are defined as:

$$\mathtt{ar}\left(\sigma_{\omega}^{\alpha,\beta}\right) = \omega, \alpha \qquad\qquad \mathtt{coar}\left(\sigma_{\omega}^{\alpha,\beta}\right) = \omega, \beta.$$

In other words, $\Sigma(\Omega)$ contains every morphism in every category $\omega \in \Omega$.

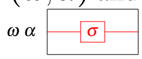
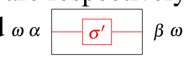
Definition 3 (Layered prop generated by a system of monoidal categories)**.** A *layered prop generated by a system of monoidal categories* Ω is the layered prop generated by the layered monoidal theory $(\Omega, \Sigma(\Omega), \mathtt{ar}, \mathtt{coar})$. Additional generators for 2-cells are given by the equalities of morphisms in each category $\omega \in \Omega$.

We denote the layered prop generated by a system of monoidal categories Ω by $\mathcal{L}(\Omega)$.

3 Pointed Profunctor Semantics

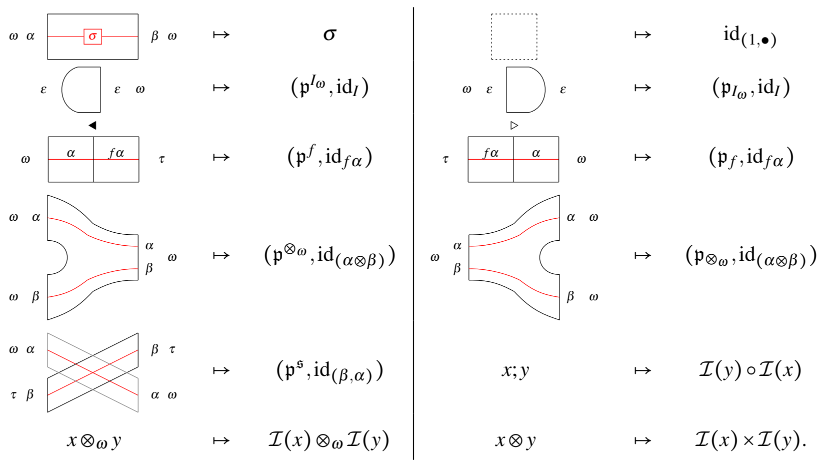
While our framework is purely syntactic (indeed, the whole point of constructing a layered prop is that we are able to treat all the layers in the same language), we are able to provide a semantic justification for the layered prop formalism: as we show in this section, they can be naturally interpreted in the category of pointed profunctors $\mathbf{Prof}_*$. We include the Appendix A on profunctors and pointed profunctors as a quick reference and to disambiguate any notation. For a proper introduction, see Borceux [6] and Loregian [14], and references therein.

Definition 4. Let $\mathcal{L}$ be a layered prop. A *profunctor model* of $\mathcal{L}$ is a 2-functor $\mathcal{L} \to \mathbf{Prof}_*$ which is consistent in the sense that

- if the 0-cells (ω, α) and (ω, β) are respectively mapped to $(\mathcal{C}, c)$ and $(\mathcal{D}, d)$, then $\mathcal{C} = \mathcal{D}$,
- if the 1-cells  and  are respectively mapped to (P, f) and (Q, g), then $P = Q$.

For the rest of this section, we assume that Ω is a system of monoidal categories. We will show that there is a natural profunctor model of the layered prop generated by Ω. To this end, we wish to define a 2-functor $\mathcal{I} : \mathcal{L}(\Omega) \to \mathbf{Prof}_*$.

Let us define $\mathcal{I}$ on objects (i.e. Ω-types) recursively as follows: $\mathcal{I}(\varepsilon) := (1, \bullet)$, and $\mathcal{I}(t; \omega, \alpha) := \mathcal{I}(t) \times (\omega, \alpha)$. In order to define $\mathcal{I}$ on morphisms, for each $\omega \in \Omega$, let us write $I_\omega : 1 \to \omega$ for the functor sending the unique object of 1 to the monoidal unit of ω. Likewise, let us write $\mathfrak{s} : \mathcal{C} \times \mathcal{D} \to \mathcal{D} \times \mathcal{C}$ for the symmetry map in $\mathbf{Cat}$. Note that since $\mathfrak{s}$ is an isomorphism, we have $\mathfrak{p}_{\mathfrak{s}} \simeq \mathfrak{p}^{\mathfrak{s}}$. We then define $\mathcal{I}$ by the following action on the generators:



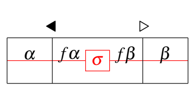
where $\mathfrak{p}^-$ and $\mathfrak{p}_-$ are the covariant and contravariant embeddings of $\mathbf{Cat}$ in the category of profunctors, and σ stands for the pointed profunctor $(\hom_\omega, \sigma)$. We prove the following proposition in Appendix B.

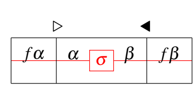
Proposition 5. *The assignment $\mathcal{I}$ is a profunctor model of $\mathcal{L}(\Omega)$. Namely, it preserves the equalities of morphisms in each category $\omega \in \Omega$ as well as the rules in Figures 2, 3 and 4.*

4 Explanations

Using the formalism introduced in the previous sections, we are now able to formulate precisely the notion of an explanation. First, we give names to two special shapes of 1-cells in a layered prop and outline their connection to explanations. We assume that we are working with a layered prop generated by a system of monoidal categories Ω.

Definition 6 (Window, cowindow)**.** A *window* is a morphism in a layered prop of the form on the left below. Dually, a *cowindow* is a morphism in a layered prop of the form on the right below.





Windows correspond to *reductive* explanations: a process at the higher level gets translated to the lower level, where we can apply laws or rules that are (presumably) more flexible, after which we translate back to the higher level, hence completing the explanation. This remark should be compared to the shape of the explanation of glucose phosphorylation in Section 5 below.

Cowindows, in turn, correspond to *functional* explanations: a process at the lower level is justified by passing through a higher level in such a way that the higher level process translates back to what is being explained. It can thus be thought that the lower level process takes place in order to yield the appropriate form at the higher level. Axioms `unit-ref` and `counit-ref` state that there is an asymmetry between reductive and functional explanations: using `unit-ref`, it is always possible to create a window (and hence give a reductive explanation), while `counit-ref` only allows reducing a trivial cowindow to the identity. Note that what we call a cowindow is usually called a *functorial box* in the literature — see e.g. [16].

We may now define what an explanation is: we do this separately for 1-cells and for 2-cells. Both correspond to a *reduction*: an explanation of a 1-cell reduces a process to another one at a lower level of abstraction, while an explanation of a 2-cell reduces a rule between two processes to a rule between reductions of these processes. For examples of explanations (now in a formal sense), see Figure 6 and the discussion in Section 7 (1-cell), and Figure 7 (2-cell).

Definition 7 (Explanation of a 1-cell)**.** Let $\mathfrak{e}$ and σ be parallel 1-cells in a layered prop (that is, having the same domain and codomain). We say that $\mathfrak{e}$ is an *explanation* of σ if

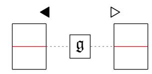
1. σ is an internal morphism contained in some category $\omega \in \Omega$,
2. every internal non-identity morphism of $\mathfrak{e}$ is contained in some category ω' such that $\omega' < \omega$ in the partial order of Ω,
3. there is either a 2-cell $\mathfrak{e} \to \sigma$ or a 2-cell $\sigma \to \mathfrak{e}$.

Definition 8 (Explanation of a 2-cell)**.** Let η and μ be parallel 2-cells in a layered prop. We say that η is an *explanation* of μ if

1. μ is generated by an equality of morphisms in some category $\omega \in \Omega$,
2. η can be constructed using the generating 2-cells of a layered prop and the 2-cells that come from an equality of morphisms in those categories ω' for which $\omega' < \omega$ in the partial order of Ω.

The above definitions correspond to the intuitive understanding of a (reductive) explanation we outlined in Section 1. The first condition in both definitions ensures that what is being explained is internal to a particular category, that is, to a description at a particular level of abstraction. The second condition says that the explanation is indeed reductive: it may only use lower levels of description than what is being explained (in addition to the metalanguage of the layered prop). This implies that an explanation must contain at least one window. The third condition in the first definition ensures that the explanation is *relevant* in the sense that it is either a sufficient or a necessary cause for what is being explained. There is no such condition for the explanations of 2-cells since in our setup there are no 3-cells. We thus simply assume that an explanation is relevant. This assumption need not be made if we are working with higher categories. These definitions can be dualised, this gives definitions of functional explanations, or "coexplanations". We will not need these in this work, and therefore omit the explicit statements.

Interestingly, if we require that the third condition of Definition 7 does not hold (i.e. there is no 2-cell between the explanation and what is being explained), we obtain the definition of a *counterfactual explanation*. Ability to model counterfactual reasoning is important for the causal analysis in the rule-based models of molecular interactions, such as the Kappa language [13]. While a particular simulation of a rule-based model may tell us that a rule $\mathfrak{e}$ was invoked in the computation of the effect σ, so that $\mathfrak{e}$ explains σ in the sense of Definition 7, this tells nothing about necessity (or sufficiency) of $\mathfrak{e}$ for σ. Thus a rule-based model is not (without modifications) able to deal with such questions as *Would* σ *occur had* $\mathfrak{e}$ *not occurred?* In a layered prop, the positive answer to such question (establishing non-necessity) can be provided by finding a counterfactual explanation of σ that has the same sort as $\mathfrak{e}$. Intuitively, a (possibly) counterfactual explanation can be thought of as a 1-cell that "fills in the gap" left by $\mathfrak{e}$:



We give an example of a counterfactual explanation in our discussion of concurrency in Section 7.

Models based on variable substitution [18] and trajectory sampling [13] have been proposed to model counterfactual statements. Since our setup remains agnostic about what the internal morphisms in a layered prop actually are, we expect that both of these situations can be modelled within a layered prop. We leave this investigation for future work.

5 Example: Glucose Phosphorylation

In this section, we construct a minimal example — inspired by Krivine [11] — that illustrates our notion of an explanation (specifically, Definition 7) for an important biochemical process known as *phosphorylation of glucose*. This is motivated by the problem of systematising a vast amount of experimental data in systems biology in a way that is easy for humans to both understand and use. Our strategy is to define three monoidal categories that are capable of talking about chemical reactions at three different abstraction levels:

$\mathcal{L}^+$	English names of the relevant molecules
$\mathcal{M}ol^+$	Molecules
$\mathcal{P}art\mathcal{M}ol^+$	Partitions of molecules into smaller units

First, let us define $\mathcal{L}^+$ as the free monoidal category with generating objects

$$\{glucose, ATP, \textit{glucose-6-phosphate}, ADP, hydrogen\ ion\},$$

whose monoidal product is denoted by +, and with just one generating morphism

$$glucose + ATP \longrightarrow \textit{glucose-6-phosphate} + ADP + hydrogen\ ion. \tag{9}$$

The generating morphism simply represents the high-level chemical rule describing phosphorylation of glucose. Here ATP and ADP stand for *adenosine triphosphate* and *adenosine diphosphate*.

5.1 Molecules and Molecule Partitions

We define a *molecule partition* as a certain connected multigraph (Definition 10). We then identify as *molecules* those molecule partitions that do not have free variables. Fix a countable set of *free variables* FW. We denote the elements of FW by lowercase Greek letters $\alpha, \beta, \gamma, \ldots$. Let us define the set of *atoms* as containing the symbol for each main-group element of the periodic table together with the symbols − and +: $\mathrm{At} \coloneqq \{-, +, H, C, O, P, \ldots\}$. Define the function $\mathbf{v} : \mathrm{At} \sqcup \mathrm{FW} \to \mathbb{N}$ as taking each element symbol to the valence of that element[2], define $\mathbf{v}(-) = \mathbf{v}(+) = 1$ and finally for all $\alpha \in \mathrm{FW}$ let $\mathbf{v}(\alpha) = 1$.

[2] This is a bit of a naive model, as valence is in general context-sensitive and not determined by a single atom. Yet this is good enough for the purposes of this example.

Definition 10 (Molecule partition)**.** A *molecule partition* is a triple (V,τ,m), where V is a finite set of *vertices*, $\tau : V \to \mathrm{At} \sqcup \mathrm{FW}$ is a function taking each vertex to its *type* and $m : V \times V \to \mathbb{N}$ is a function satisfying the following conditions:

- for all $v \in V$, we have $m(v,v) = 0$,
- for all $v,w \in V$, we have $m(v,w) = m(w,v)$,
- for all $v,u \in V$ with $v \neq u$, there are $w_0,\ldots,w_n \in V$ such that $w_0 = v$ and $w_n = u$ and $m(w_{i-1},w_i) \neq 0$ for each $i = 1,\ldots,n$,
- for all $v \in V$, we have $\sum_{u \in V} m(u,v) = \mathbf{v}\tau(v)$.

In other words, the integers $m(i,j)$ form an adjacency matrix of an irreflexive, symmetric and connected multigraph, and the sum of each row or column gives the valence of the (type of) corresponding vertex.

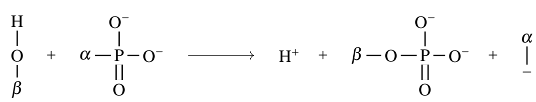
Definition 11 (Molecule)**.** We say that a molecule partition (V,τ,m) is a *molecule* if the image of the function τ is contained in At.

We denote the set of all molecules by $\mathcal{M}ol$ and the set of all molecule partitions by $\mathcal{P}art\mathcal{M}ol$. Define the *partitioning relation* $R \subseteq \mathcal{P}art\mathcal{M}ol \times (\mathcal{P}art\mathcal{M}ol \times \mathcal{P}art\mathcal{M}ol)$ as follows. Let $M = (V,\tau,m)$ be a molecule partition, let $u,v \in V$ and let $\alpha \in \mathrm{FW}$. Denote by $m' : V \times V \to \mathbb{N}$ the function such that $m'(u,v) = m'(v,u) = 0$ and $m' = m$ otherwise. Suppose that the following conditions are satisfied:

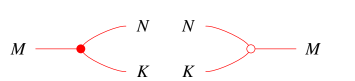
1. $m(u,v) = 1$,
2. the graph (V,m') is not connected,
3. α does not appear as a free variable in M (that is, $\tau(w) \neq \alpha$ for all $w \in V$).

In such case we denote by $V(u)$ and $V(v)$ the connected components of u and v, respectively, in (V,m'). Let $M_u^\alpha = (V(u) \sqcup \{\alpha\},\tau_\alpha,m_u)$ be the molecule partition where $\tau_\alpha(\alpha) = \alpha$ and $\tau_\alpha = \tau$ otherwise, and $m_u(u,\alpha) = m_u(\alpha,u) = 1$ and $m_u = m$ otherwise. The molecule partition $M_v^\alpha = (V(v) \sqcup \{\alpha\},\tau_\alpha,m_v)$ is defined similarly. Now we finally define R by stipulating that $MR(M_u^\alpha,M_v^\alpha)$ for all M, v, u and α that satisfy the above conditions.

Let us define $\mathcal{M}ol^+$ as the free monoidal category with generating objects $\mathcal{M}ol$ and just one generating morphism, which has the same shape as the generating morphism of $\mathcal{L}^+$ (9), except that all the English names of the molecules are translated to the corresponding graphs (see Figure 5). Similarly, define $\mathcal{P}art\mathcal{M}ol^+$ as the free monoidal category with generating objects $\mathcal{P}art\mathcal{M}ol$. For all variables α and β we add the rule



as a generating morphism to $\mathcal{P}art\mathcal{M}ol^+$. We draw this as a box: . Further, for all molecule partitions M,N and K such that $MR(N,K)$ we introduce the following generators



We now wish to define monoidal functors $\mathcal{L}^+ \xrightarrow{T} \mathcal{M}ol^+ \xhookrightarrow{i} \mathcal{P}art\mathcal{M}ol^+$ so as to make this into a system of monoidal categories. First, define a monoidal functor $T : \mathcal{L}^+ \to \mathcal{M}ol^+$ by the action on the generating objects in Figure 5, where we use the convention from chemistry that an unlabelled vertex represents a carbon atom with an appropriate number of hydrogen atoms attached to it to make its valence equal to 4. The only generating morphism of $\mathcal{L}^+$ is mapped to the only generating morphism of $\mathcal{M}ol^+$. The monoidal functor $\mathcal{M}ol^+ \xhookrightarrow{i} \mathcal{P}art\mathcal{M}ol^+$ is identity on objects and maps the only generating morphism of $\mathcal{M}ol^+$ to the composite morphism in the middle rectangle of Figure 6.

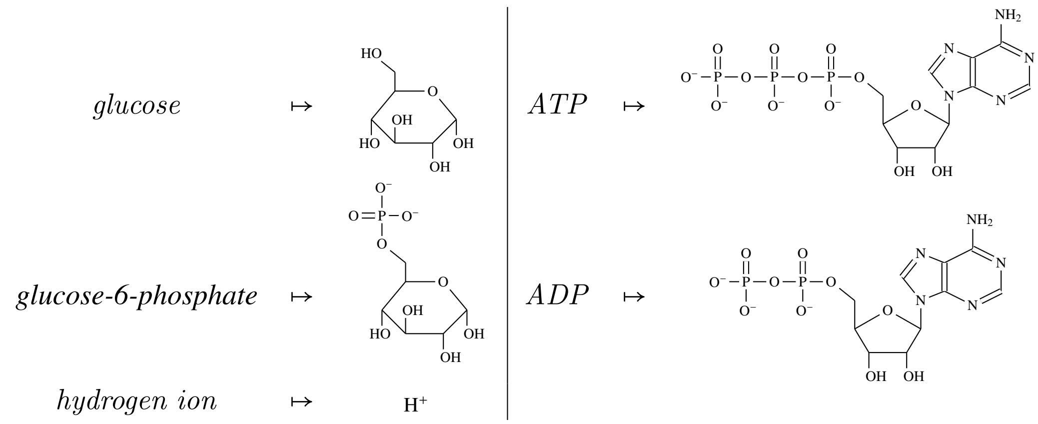


Figure 5: Translation of English names to chemical graphs.

5.2 Explaining Phosphorylation

We can now use the lowest level language $\mathcal{P}art\mathcal{M}ol^{+}$ to explain the high-level rule (9) as is shown in Figure 6. Note that this is indeed an explanation according to Definition 7, since the rule that is being explained is internal to $\mathcal{L}^{+}$, the explanation does not use any non-identity morphisms from $\mathcal{L}^{+}$, and the explanation can be derived starting from the rule (9) using the 2-cells of the layered prop, whose composite gives a 2-cell from the rule to the explanation. While the diagram in Figure 6 fulfills the

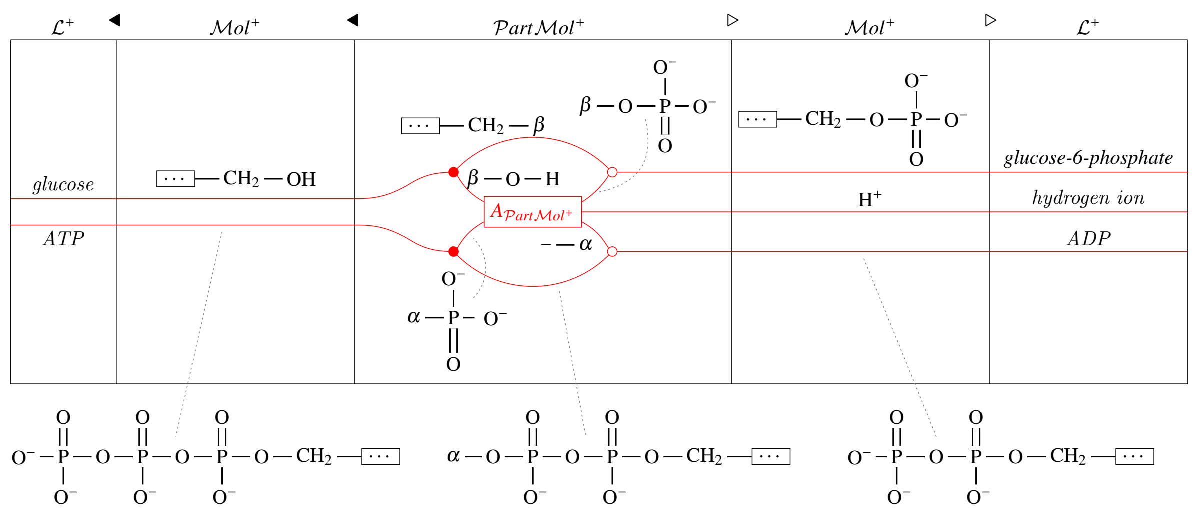


Figure 6: Explaining glucose phosphorylation: each area between the vertical black bars represents a layer, so in this case $\mathcal{L}^{+}$, $\mathcal{M}ol^{+}$ or $\mathcal{P}art\mathcal{M}ol^{+}$.

definition of an explanation, it is not very “explanatory" in an intuitive sense. This is because we chose to stop at a fairly high level of abstraction. It is important to note that the morphism $A_{\mathcal{P}art\mathcal{M}ol^{+}}$ is just a black box, which could itself be explained at the level of atoms exchanging electrons. Modularity of layered props would then allow us to add this further level to the diagram. The resulting explanation would bring us closer to satisfactorily answering the question *Why does this reaction occur?*.

We conclude this example by remarking that we didn't have to assume that we already know the higher level chemical rule (9). Instead we could have chosen to *generate* the higher level rules by declaring as morphisms every 1-cell from $(\mathcal{L}^+, c)$ to $(\mathcal{L}^+, d)$ for some objects $c, d \in \mathcal{L}^+$. Instead of an explanation, this would correspond to deriving higher level rules from a single lower level rule.

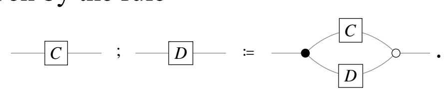
6 Example: Electrical Circuits

While in the previous section we constructed a minimal example from scratch, in this section we take an existing example from the literature where explanations are already used implicitly. Namely, we focus on the research program that has formalised electrical circuits in terms of string diagrams and given them an interpretation in graphical affine algebra [1, 4, 5, 3].

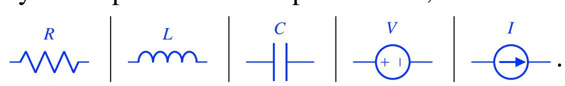
The string diagrammatic electrical circuit theory is a paradigmatic example of explanations taking a functorial form: the relations between electrical components are proved by interpreting them as morphisms in the graphical affine algebra. Thus this example also shows how functorial explanations can be incorporated into our framework. Note, however, that Boisseau and Sobociński [3] already use something like layered explanations to only partially translate their diagrams. They call the notational device used for this an *impedance box*. In our language, impedance boxes arise in a principled way as instances of a general definition: they are just windows (Definition 6) of a particular shape.

We define the props *graphical affine algebra* $\mathbf{GAA}$ and *electrical circuits* $\mathbf{ECirc}$ as well as the translation functor $\mathcal{I} : \mathbf{ECirc} \to \mathbf{GAA}$ as in [3], except that we quotient the morphisms in $\mathbf{ECirc}$ by equality under $\mathcal{I}$. This makes $\mathcal{I}$ faithful, which we reflect in our syntax by adding a left inverse to the 2-cell `unit-ref` in Figure 3[3]. Additionally, we define the *impedance category* $\mathbf{Imp}$ and the category of bipoles $\mathbf{Bip}$ in order to express the impedance calculus of [3] formally within our setup.

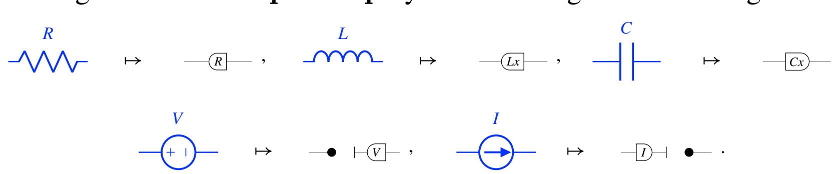
Definition 12 (Impedance category)**.** The *impedance category* $\mathbf{Imp}$ is a prop whose generating morphisms are all the morphisms of $\mathbf{GAA}$ with exactly one input and exactly one output. The identity is —• ◦—, and composition is given by the rule



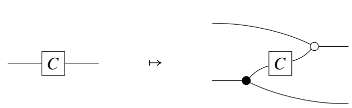
Definition 13 (Bipole category)**.** The *bipole category* $\mathbf{Bip}$ is the subcategory of $\mathbf{ECirc}$ given by those generators which have exactly one input and one output. That is, it is the free prop generated by

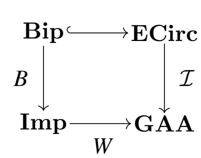


Define the “boxing" functor $B : \mathbf{Bip} \to \mathbf{Imp}$ by the following action on the generators:



Further, define a "wrapping" functor $W : \mathbf{Imp} \to \mathbf{GAA}$ which acts as $n \mapsto 2n$ on objects and on morphisms as shown below left. The boxing and the wrapping functors are so defined that we have a commutative square below right:

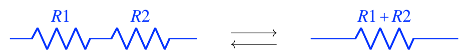




[3]This causes some problems for the semantic interpretation of Section 3, whose resolution we leave for a more technical paper.

where the top horizontal morphism is the inclusion functor, and $\mathcal{I}$ is the translation of electrical circuits to graphical affine algebra. Treating the above diagram of monoidal functors as a system of monoidal categories, we obtain a layered prop. Within this layered prop, we are able to replicate what is called the *impedance calculus* in [3]. To illustrate this, we give an explanation of the rule governing the sequential composition of resistors. This rule is a 2-cell in the layered prop, and the explanation is therefore that of a 2-cell (Definition 8).

Figure 7 shows how the rule for composing two resistors



can be explained (this is essentially part (i) of Proposition 3 of [3]). This is indeed an explanation of a 2-cell (Definition 8), since we are explaining an equality in $\mathbf{ECirc}$ using only the 2-cells of a layered prop and a 2-cell from $\mathbf{Imp}$ (the third 2-cell of the derivation).

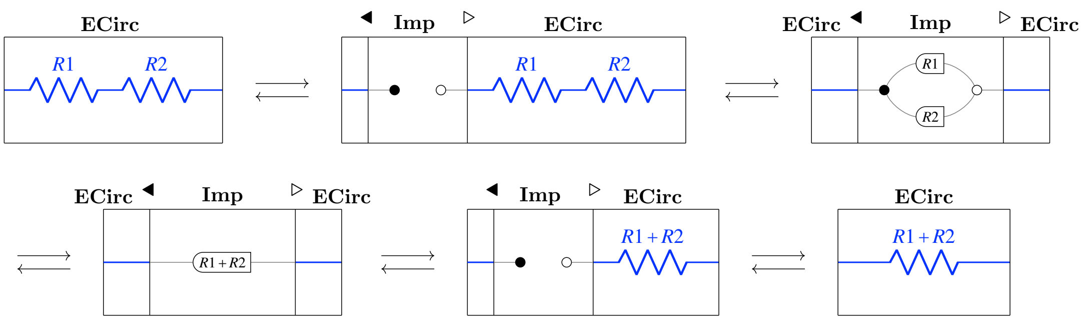


Figure 7: Explaining sequential composition of resistors. Note that the explanation relies on the composition in $\mathbf{Imp}$. This could, in turn, be itself explained by translating to $\mathbf{GAA}$.

As for the example with glucose phosphorylation, we could choose to generate the equalities in $\mathbf{ECirc}$ rather than assume them a priori. In this case, there would be no need to quotient morphisms in $\mathbf{ECirc}$ by equality under the translation functor, yet the equality of 1-cells should be taken up to a trivial window.

7 Example: Calculus of Communicating Systems

The calculus of communicating systems (CCS) [17] is widely used to reason about programs, formal languages and concurrency. Here we consider a restricted version of CCS and two ways to give semantics to the CCS expressions: *reduction semantics* is very heavily syntactic, in addition to the structural congruences, it only allows for only one rewrite rule (the *reduction*), while the *labelled transition system* (LTS) semantics [17] is more flexible and comes with more rewrite rules. Our goal is to show how the LTS semantics can be used to give an explanation (this time in the sense of Definition 7) of the rewrite rule of the reduction semantics. Intuitively, the LTS semantics may be seen as a lower level implementation of the concurrent processes described abstractly by CCS. Furthermore, we demonstrate that LTS semantics has a larger scope of allowed derivations than the reduction semantics by giving a counterfactual explanation of a rewrite rule in the reduction semantics.

Let us fix a set of *action names* A. Define $\bar{A} \coloneqq \{\bar{a} : a \in A\}$ and $Act \coloneqq A \cup \bar{A} \cup \{\tau\}$. The set of *processes* is defined recursively as follows, where x ranges over Act:

$$P ::= \quad 0 \;\big|\; x.P \;\big|\; P \parallel P.$$

Definition 14 (Congruence)**.** Define the *congruence* as the smallest equivalence relation $\sim$ on the set of processes that satisfies:

$$P \parallel Q \sim Q \parallel P, \qquad (P \parallel Q) \parallel R \sim P \parallel (Q \parallel R),$$

$$0 \parallel P \sim P, \qquad \text{if } P \sim P' \text{ and } Q \sim Q', \text{ then } P \parallel Q \sim P' \parallel Q'.$$

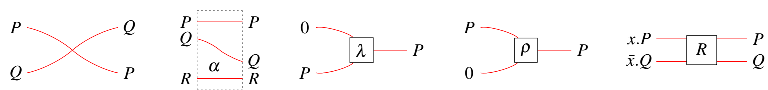
Definition 15 (Reduction semantics)**.** A *rewrite rule* in *reduction semantics* is an ordered pair of processes, which we write as $P \to Q$, generated by the following three *deduction rules*:

$$\frac{}{x.P \parallel \bar{x}.Q \to P \parallel Q} \qquad \frac{P \to Q}{P \parallel R \to Q \parallel R} \qquad \frac{P \to Q \quad P \sim P' \quad Q \sim Q'}{P' \to Q'}$$

In other words, rewrite rules are parallel compositions of the *reduction* (first rule in the above definition) up to the congruence. For instance, we can derive the following rewrite rule:

$$x.0 \parallel (y.0 \parallel \bar{x}.0) \to 0 \parallel (y.0 \parallel 0). \tag{16}$$

In order to talk about layered props, we wish to express reduction semantics as a monoidal category. Let $\mathbf{Red}$ be the monoidal category whose objects are the processes, monoidal product on objects is the parallel composition $\parallel$, and whose morphisms are generated by:



together with inverses for the first four generators. Here P, Q and R range over processes, and x ranges over A. The first four morphisms correspond to the congruence, and R corresponds to the first deduction rule for transitions. The parallel composition is taken care of by the monoidal structure. Note that the monoidal product is not strictly associative, so we need to keep track of the bracketing of the wires.

Next, we introduce a LTS as an alternative semantics for the above fragment of CCS.

Definition 17 (Labelled transition)**.** A *labelled transition* is a triple (P, x, Q), where P and Q are processes and $x \in Act$, generated by the deduction rules below. We write $P \xrightarrow{x} Q$ for such triple. Note that we write the silent action τ as an unlabelled arrow.

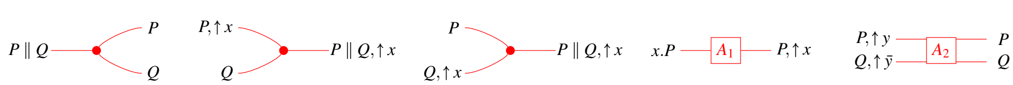
$$\frac{P' \xrightarrow{x} P}{P' \parallel Q \xrightarrow{x} P \parallel Q} \qquad \frac{P' \xrightarrow{x} P}{Q \parallel P' \xrightarrow{x} Q \parallel P} \qquad \frac{}{x.P \xrightarrow{x} P} \qquad \frac{P' \xrightarrow{x} P \quad Q' \xrightarrow{\bar{x}} Q}{P' \parallel Q' \to P \parallel Q}$$

Definition 18 (Bisimulation)**.** A *bisimulation* on the set of processes is a binary relation b such that for all processes P and Q and all $x \in Act$, we have that PbQ implies

- if $P \xrightarrow{x} P'$, then there is a process Q' such that $Q \xrightarrow{x} Q'$ and $P'bQ'$,
- if $Q \xrightarrow{x} Q'$, then there is a process P' such that $P \xrightarrow{x} P'$ and $P'bQ'$.

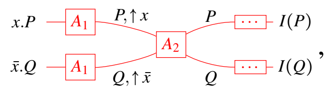
The *largest bisimulation* is the union of all bisimulations.

Labelled transitions define the *LTS semantics*, which, similarly to the reduction semantics, can be modelled as a monoidal category. Thus let $\mathbf{LTS}$ be the free monoidal category whose generating objects are pairs $P, \uparrow x$, where P is a process and $x \in Act$. We think of $\uparrow x$ as the "pending action", and omit the silent pending action: $P \coloneqq P, \uparrow \tau$. The morphisms of $\mathbf{LTS}$ are generated by



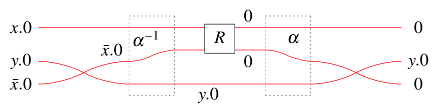
where P and Q range over processes, $x \in Act$ and $y \in A \cup \bar{A}$ and we identify $\bar{\bar{y}} := y$. The structural isomorphisms of the monoidal category have the same form as the structural isomorphisms of $\mathbf{Red}$, and correspond to the largest bisimulation. The other morphisms in $\mathbf{LTS}$ model those rewrite rules of the usual LTS semantics that are derivable via our restricted set of deduction rules.

There is a monoidal functor $I : \mathbf{Red} \to \mathbf{LTS}$, whose action on objects is defined as $0 \mapsto 0, \uparrow \tau$, $x.P \mapsto P, \uparrow \tau$ and $P \parallel Q \mapsto (I(P), I(Q))$. For morphisms, I takes each structural isomorphism in $\mathbf{Red}$ to the corresponding isomorphism in $\mathbf{LTS}$, and the morphism R to

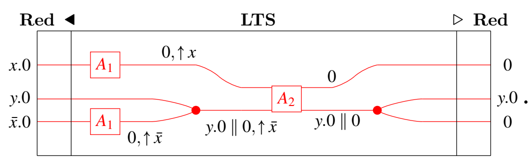


where the dots refer to an appropriate decomposition of P and Q into $I(P)$ and $I(Q)$.

We use the functor I to view the LTS semantics as the lower level language that explains the reduction semantics. For instance, we can explain the rewrite rule (16) by just moving its derivation in $\mathbf{Red}$



through the window, that is, essentially by applying I. In this case, we are also able to give a counterfactual explanation:



The above diagram is indeed a counterfactual explanation (see the discussion in Section 4) of the rewrite rule (16): (1) the rewrite rule is an internal morphism in $\mathbf{Red}$, (2) every non-identity internal morphism in the diagram is contained in $\mathbf{LTS}$, which is strictly below $\mathbf{Red}$ in the partial order of the layered prop, (3) there are no 2-cells between the rewrite rule and the diagram. To see that (3) is indeed the case, note that there are in fact no 2-cells having the above diagram as either domain or codomain (one can see this by going through the generators of 2-cells of a layered prop one by one).

The fact that there is a counterfactual explanation of the rewrite rule (16) shows that it is not *necessary* to invoke the (analogue of) rule R in its derivation at the level of LTS. This observation allows us to show neatly that LTS semantics is more flexible than the reduction semantics, in the sense that there are more derivations of the same transitions. Note that the counterfactual explanation does not need to be more complex than an ordinary explanation: in this case it is in fact more direct, in the sense that it shows that there is an actual labelled transition, while the explanation obtained by translating the diagram in $\mathbf{Red}$ merely shows that there is a labelled transition up to the largest bisimulation.

8 Conclusions and Future Work

We have taken the first steps towards developing a mathematical framework for formalising explanations. Explanations in a category theoretic context usually take the form of a functor, whose domain is thought of as syntax and codomain as semantics. Our approach differs from this: in a layered prop, there are several possible translations to different levels, which are nonetheless syntactically represented in the same language (that is, within one layered prop). A layered prop allows one to easily work with different theories describing the same phenomenon, and, importantly, allows for partial translations instead of having to translate the full diagram, as we have illustrated with the examples. We have also observed how counterfactual processes arise naturally within layered props: these are those processes that “look like” a translation without being one. Furthermore, the examples show that the same abstract principles hold in areas as distant as biology, electrical circuit theory and concurrency theory. Layered props can thus indeed be conceived as the initial stage of a general mathematical theory of explanations.

On the mathematical level, the next phase of developing the theory is to explore the precise connection of layered props to pointed profunctors. Currently, there is a canonical 2-functor which translates a layered prop generated by a system of monoidal categories to the category of pointed profunctors which preserves the axioms of a layered prop. One way to proceed would be to characterise the image of this functor, thus identifying a subcategory of $\mathbf{Prof}_*$ to which a given layered prop is equivalent. Another mathematical aspect that is important for practical applications is to modify the definition of a layered prop to allow for non-strictly associative monoidal categories, as for instance described diagrammatically in [23]. As briefly remarked in Section 6, the current semantics cannot adequately handle the important special case when the translation functor is faithful. This suggests that the current interpretation of the 2-cells as natural transformations is too restrictive, and some other notion of 2-cells for pointed profunctors should be used. In order to connect layered props to known structures, it would also be useful to express them as a Grothendieck construction.

Even though it was beyond the scope of this paper, we believe it is important to connect our work with the philosophy of science literature on explanations. Since the initial motivation for our work comes from biology, it is particularly interesting to see how ideas on explanations and causality in biology fit our framework. For instance, one of the main motivations of Robert Rosen for introducing the theoretical framework of *relational biology* was to put the *function* of an organism on equal grounding with the *mechanism* that underlies it [21]. This can be modelled within a layered prop: reductive and functional explanations are *a priori* completely symmetric, and in any case equally well-defined.

Several systems with multiple layers are known in the applied category theory literature. In addition to the already discussed [23] and [3] (Section 6), we mention the formalism of *hierarchical petri nets* [7], and Román’s notion of an *open diagram* [20]. All of these rely on an intuitive notion composing processes at different levels, and hence we plan to explore them using layered props.

Acknowledgements The credit for the original idea for the “calculus of refinement and coarsening”, as well as for the examples featuring chemical reactions and CCS goes to Jean Krivine. We also thank him for feedback and inspiring discussions during various stages of this work. We thank Samson Abramsky for discussing and giving feedback on ideas that lead to this work. We thank Cole Comfort for conversations and for pointing us towards the literature on internal string diagrams. LL thanks Jamie Vicary, Nick Hu, Alex Rice, Calin Tataru and Ioannis Markakis for an opportunity to present and discuss an early version of this work.

References

[1] John C. Baez & Brendan Fong (2015): *A Compositional Framework for Passive Linear Networks*, doi:10.48550/ARXIV.1504.05625. Available at `https://arxiv.org/abs/1504.05625`.

[2] Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries & Jamie Vicary (2015): *Modular categories as representations of the 3-dimensional bordism 2-category*, doi:10.48550/ARXIV.1509.06811. Available at `https://arxiv.org/abs/1509.06811`.

[3] Guillaume Boisseau & Paweł Sobociński (2021): *String Diagrammatic Electrical Circuit Theory*, doi:10.48550/ARXIV.2106.07763. Available at `https://arxiv.org/abs/2106.07763`.

[4] Filippo Bonchi, Robin Piedeleu, Pawel Sobocinski & Fabio Zanasi (2019): *Graphical Affine Algebra*. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, IEEE, pp. 1–12, doi:10.1109/LICS.2019.8785877.

[5] Filippo Bonchi, Paweł Sobociński & Fabio Zanasi (2021): *A Survey of Compositional Signal Flow Theory*, pp. 29–56. Springer International Publishing, Cham, doi:10.1007/978-3-030-81701-5_2.

[6] Francis Borceux (1994): *Bicategories and distributors*, p. 281–324. *Encyclopedia of Mathematics and its Applications* 1, Cambridge University Press, doi:10.1017/CBO9780511525858.009.

[7] Fabrizio Romano Genovese, Jelle Herold, Fosco Loregian & Daniele Palombi (2021): *A Categorical Semantics for Hierarchical Petri Nets*. *Electronic Proceedings in Theoretical Computer Science* 350, pp. 51–68, doi:10.4204/eptcs.350.4.

[8] William S. Hlavacek, James R. Faeder, Michael L. Blinov, Richard G. Posner, Michael Hucka & Walter Fontana (2006): *Rules for modeling signal-transduction systems. Science's STKE : signal transduction knowledge environment* 2006(344), p. re6, doi:10.1126/stke.3442006re6.

[9] Nick Hu (2019): *External traced monoidal categories*. Master's thesis, Department of Computer Science. Available at `https://www.cs.ox.ac.uk/publications/publication13596-abstract.html`.

[10] Jean Krivine (2017): *Systems Biology*. *ACM SIGLOG News* 4(3), p. 43–61, doi:10.1145/3129173.3129182.

[11] Jean Krivine (2019): *Physical systems, composite explanations and diagrams*. `https://youtu.be/w8vTTvA0HTE` and `https://www.cl.cam.ac.uk/events/syco/strings3-syco5/`.

[12] Stephen Lack (2004): *Composing PROPs*. *Theory and Applications of Categories* 13(9), pp. 147–163.

[13] Jonathan Laurent, Jean Yang & Walter Fontana (2018): *Counterfactual Resimulation for Causal Analysis of Rule-Based Models*. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, International Joint Conferences on Artificial Intelligence Organization, pp. 1882–1890, doi:10.24963/ijcai.2018/260.

[14] Fosco Loregian (2015): *Coend calculus*, doi:10.48550/ARXIV.1501.02503. Available at `https://arxiv.org/abs/1501.02503`.

[15] Saunders Maclane (1965): *Categorical Algebra*. *Bulletin (new series) of the American Mathematical Society* 71(1), pp. 40–106, doi:10.1090/S0002-9904-1965-11234-4.

[16] Paul-André Melliès (2006): *Functorial Boxes in String Diagrams*. In Zoltán Ésik, editor: *Computer Science Logic*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–30, doi:10.1007/11874683_1.

[17] Robin Milner (1980): *A Calculus of Communicating Systems*. *Lecture Notes in Computer Science* 92, Springer Netherlands, Netherlands, doi:10.1007/3-540-10235-3.

[18] Judea Pearl (2009): *The Logic of Structure-Based Counterfactuals*, 2 edition, p. 201–258. Cambridge University Press, doi:10.1017/CBO9780511803161.009.

[19] (2019): *Explainable AI: the Basics - Policy Briefing*. Available at `royalsociety.org/ai-interpretability`.

[20] Mario Romá n (2021): *Open Diagrams via Coend Calculus*. *Electronic Proceedings in Theoretical Computer Science* 333, pp. 65–78, doi:10.4204/eptcs.333.5.

[21] Robert Rosen (1991): *Life itself : a comprehensive inquiry into the nature, origin, and fabrication of life / Robert Rosen*. Complexity in ecological systems series, Columbia University Press, New York.

[22] P. Selinger (2010): *A Survey of Graphical Languages for Monoidal Categories*. In: *New Structures for Physics*, Springer Berlin Heidelberg, pp. 289–355, doi:10.1007/978-3-642-12821-9_4.

[23] Paul Wilson, Dan Ghica & Fabio Zanasi (2022): *String diagrams for non-strict monoidal categories*, doi:10.48550/ARXIV.2201.11738. Available at `https://arxiv.org/abs/2201.11738`.

[24] Fabio Zanasi (2015): *Interacting Hopf Algebras: the theory of linear systems*. Ph.D. thesis, Laboratoire de l'Informatique du Parallélisme, doi:10.48550/ARXIV.1805.03032. Available at `https://arxiv.org/abs/1805.03032`.

A Profunctors and Pointed Profunctors

In order to fix notational conventions, we recall the standard definition of profunctors. We also define the not-so-standard category of pointed profunctors. We state the results about (pointed) profunctors needed in the main body of the paper, mostly without proof.

A.1 Profunctors

We follow Loregian [14] in our discussion of profunctors and coends.

Definition 19 (Bicategory of profunctors)**.** Define the bicategory of *profunctors* $\mathbf{Prof}$ as follows.

- the 0-cells are (small) categories,
- the 1-cells, denoted by $\mathcal{C} \relbar\!\mapsto \mathcal{D}$, are functors

$$\mathcal{C}^{op} \times \mathcal{D} \to \mathbf{Set},$$

- the $2-cells$ are natural transformations $\alpha : F \Rightarrow G$,
- the composition

$$c_{\mathcal{A},\mathcal{B},\mathcal{C}} : \mathbf{Prof}(\mathcal{A},\mathcal{B}) \times \mathbf{Prof}(\mathcal{B},\mathcal{C}) \to \mathbf{Prof}(\mathcal{A},\mathcal{C})$$

takes profunctors $F : \mathcal{A} \relbar\!\mapsto \mathcal{B}$ and $G : \mathcal{B} \relbar\!\mapsto \mathcal{C}$ to the coend $G \circ F = \int^B F(-,B) \times G(B,=)$. Explicitly, we define

$$(G \circ F)(A,C) \coloneqq \int^{B \in \mathcal{B}} F(A,B) \times G(B,C).$$

There is a bifunctor

$$\times : \mathbf{Prof} \times \mathbf{Prof} \to \mathbf{Prof}$$

defined as the product functor of n-cells for each $n = 0,1,2$ which equips $\mathbf{Prof}$ with a symmetric monoidal structure.

Given a 2-category $\mathcal{K}$, let us write $\mathcal{K}^{op}$ for the 2-category whose 0-cells and 2-cells are those of $\mathcal{K}$ and whose 1-cells are the reversed 1-cells of $\mathcal{K}$, that is, for all 0-cells A and B we have $\mathcal{K}^{op}(A,B) = \mathcal{K}(B,A)^{op}$. Similarly, we write $\mathcal{K}^{co}$ for the 2-category whose 0-cells and 1-cells are those of $\mathcal{K}$ and whose 2-cells are the reversed 2-cells of $\mathcal{K}$, that is, for all 0-cells A and B we have $\mathcal{K}^{co}(A,B) = \mathcal{K}(A,B)^{op}$.

There are two ways to embed the 2-category $\mathbf{Cat}$ into $\mathbf{Prof}$: one is contravariant on the 1-cells, the other on the 2-cells. Both embeddings are identity on objects. The embedding

$$\mathfrak{p}^{-} : \mathbf{Cat}^{co} \to \mathbf{Prof}$$

takes a functor $F : \mathcal{C} \to \mathcal{D}$ to the profunctor $\mathfrak{p}^F : \mathcal{C} \relbar\!\mapsto \mathcal{D}$ defined on objects by $\mathfrak{p}^F(C,D) \coloneqq \mathcal{D}(FC,D)$, and a natural transformation $\eta : F \to G$ to the natural transformation $\mathfrak{p}^G \to \mathfrak{p}^F$ whose (C,D)-component is given by $- \circ \eta_C$.

Dually, the embedding

$$\mathfrak{p}_{-} : \mathbf{Cat}^{op} \to \mathbf{Prof}$$

takes a functor $F : \mathcal{C} \to \mathcal{D}$ to the profunctor $\mathfrak{p}_F : \mathcal{D} \relbar\!\mapsto \mathcal{C}$ defined on objects by $\mathfrak{p}_F(C,D) \coloneqq \mathcal{D}(D,FC)$, and a natural transformation $\eta : F \to G$ to the natural transformation $\mathfrak{p}^F \to \mathfrak{p}^G$ whose (C,D)-component is given by $\eta_C \circ -$.

Both $\mathfrak{p}^{-}$ and $\mathfrak{p}_{-}$ are 2-functors, locally fully faithful and for every functor F the 1-cell $\mathfrak{p}^F$ is the left adjoint to $\mathfrak{p}_F$ in the bicategory $\mathbf{Prof}$ (see section 5.1 of Loregian [14] for the details).

Proposition 20. *Both* $\mathfrak{p}^- : \mathbf{Cat}^{co} \to \mathbf{Prof}$ *and* $\mathfrak{p}_- : \mathbf{Cat}^{op} \to \mathbf{Prof}$ *are monoidal 2-functors.*

Proof. We prove the result for $\mathfrak{p}^-$: the argument for $\mathfrak{p}_-$ is dual.

Since the embedding is identity on objects, the monoidal product of 0-cells (which is just the cartesian product of categories) is preserved.

For 1-cells, let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{C}' \to \mathcal{D}'$ be functors. We wish to show that $\mathfrak{p}^{F\times G} \simeq \mathfrak{p}^F \times \mathfrak{p}^G$. We compute as follows:

$$\begin{aligned}
\mathfrak{p}^{F\times G}(C,C';D,D') &= \mathcal{D}\times\mathcal{D}'(F\times G(C,C'),(D,D'))\\
&= \mathcal{D}\times\mathcal{D}'((FC,GC'),(D,D'))\\
&= \mathcal{D}(FC,D)\times\mathcal{D}'(GC',D')\\
&= \mathfrak{p}^F(C,D)\times\mathfrak{p}^G(C',D')\\
&= \left(\mathfrak{p}^F\times\mathfrak{p}^G\right)(C,C';D,D'),
\end{aligned}$$

whence it follows that $\mathfrak{p}^{F\times G}$ and $\mathfrak{p}^F \times \mathfrak{p}^G$ agree on objects. The fact that they agree on morphisms is a similar computation.

For 2-cells, let $F,F' : \mathcal{C} \to \mathcal{D}$ and $G,G' : \mathcal{C}' \to \mathcal{D}'$ all be functors. Given natural transformations $\eta : F \to F'$ and $\mu : G \to G'$, we wish to show that $\mathfrak{p}^{\eta\times\mu} = \mathfrak{p}^\eta \times \mathfrak{p}^\mu$. This follows by observing that their components coincide:

$$\mathfrak{p}^{\eta\times\mu}_{(C,C';D,D')} = -\circ(\eta\times\mu)_{(C,C')} = (-\circ\eta_C)\times(-\circ\mu_{C'}) = \mathfrak{p}^\eta_{C,D}\times\mathfrak{p}^\mu_{C',D'} = (\mathfrak{p}^\eta\times\mathfrak{p}^\mu)_{C,C';D,D'}.$$

□

A.2 Pointed Profunctors

Definition 21 (Pointed profunctors)**.** Define the bicategory of *pointed profunctors* $\mathbf{Prof}_*$ as follows:

- the 0-cells are pairs $(\mathcal{C},c)$ of a (small) category $\mathcal{C}$ and an object $c \in \mathcal{O}b(\mathcal{C})$,
- the 1-cells $(P,f) : (\mathcal{C},c) \to (\mathcal{D},d)$ consist of a profunctor $P : \mathcal{C} \to\!\!\!\!\!\mapsto \mathcal{D}$, that is, a functor

  $$P : \mathcal{C}^{op}\times\mathcal{D} \to \mathbf{Set},$$

  together with an element $f \in P(c,d)$,
- the 2-cells $\alpha : (P,f) \to (Q,g)$ are natural transformations $\alpha : P \Rightarrow Q$ such that $\alpha_{c,d}(f) = g$,
- the composition of $(P,f) : (\mathcal{C},c) \to (\mathcal{D},d)$ and $(Q,g) : (\mathcal{D},d) \to (\mathcal{E},e)$ is given by $(Q\circ P,[f,g])$, where $\circ$ is the composition of profunctors and $[f,g]$ the equivalence class of the pair (f,g) in $(Q\circ P)(c,e)$.

Note that a pointed hom-functor $(\mathcal{C}(-,-),f) : (\mathcal{C},c) \to (\mathcal{C},c')$ is precisely a morphism $f : c \to c'$. Thus we will simply write the hom-functor $(\mathcal{C}(-,-),f)$ as f. For a category $\mathcal{C}$, we define an assignment

$$\begin{aligned}
z_{\mathcal{C}} : \mathcal{C} &\to \mathbf{Prof}_*\\
c &\mapsto (\mathcal{C},c)\\
(f : c \to c') &\mapsto (f : (\mathcal{C},c) \to (\mathcal{C},c'))
\end{aligned}$$

Proposition 22. *The assignment* $z_{\mathcal{C}}$ *is a pseudofunctor (when* $\mathcal{C}$ *is taken to have the trivial bicategory structure).*

Proof. We first show that $z_{\mathcal{C}}$ preserves composition. Thus let $f : c \to d$ and $g : d \to e$ be morphisms in $\mathcal{C}$. First, $\mathcal{C}(-,-) \circ \mathcal{C}(-,-) \simeq \mathcal{C}(-,-)$ since the hom-profunctor is the identity profunctor. Observe that such an isomorphism is given by $\int^a \mathcal{C}(c,a) \times \mathcal{C}(a,e) \xrightarrow{\sim} \mathcal{C}(c,e)$ given by $[n,m] \mapsto m \circ n$ (this is well-defined). Thus in particular $[f,g] \mapsto gf$, whence

$$z_{\mathcal{C}}(g) \circ z_{\mathcal{C}}(f) = (\mathcal{C}(-,-),g) \circ (\mathcal{C}(-,-),f) \simeq (\mathcal{C}(-,-),gf) = z_{\mathcal{C}}(gf).$$

From the above it follows that $\mathrm{id}_c : (\mathcal{C},c) \to (\mathcal{C},c)$ is the identity on $(\mathcal{C},c)$, so that $z_{\mathcal{C}}$ preserves the identities. □

There is a pseudofunctor

$$\times : \mathbf{Prof}_* \times \mathbf{Prof}_* \to \mathbf{Prof}_*$$

defined as

- $(C,c) \times (D,d) \coloneqq (C \times D, (c,d))$ on the 0-cells,
- $(P,f) \times (Q,g) \coloneqq (P \times Q, (f,g))$ on the 1-cells,
- the product of natural transformations on the 2-cells.

Writing 1 for the terminal category and • for its unique object, we have the following:

Proposition 23. $(\mathbf{Prof}_*, \times, (1, \bullet))$ *is a symmetric monoidal bicategory.*

B Semantic Properties of Layered Props

We discuss the properties that the interpretation functor $\mathcal{I} : \mathcal{L}(\Omega) \to \mathbf{Prof}_*$ has. Throughout the section, we assume that Ω is a system of monoidal categories.

We begin by proving that $\mathcal{I}$ is indeed a pointed profunctor model (Proposition 5).

Proof of Proposition 5. The equalities of morphisms for each category $\omega \in \Omega$ are preserved by Proposition 22. The unit and counit maps in Figure 3 are preserved and the triangle equalities for them hold since we have defined each pair of profunctors as an adjoint pair. Since all the internal morphisms are identities, there is nothing to show for the composition of the internal morphisms.

All the rules in Figure 4 follow from the fact that each category and functor in Ω is monoidal and that both $\mathfrak{p}^-$ and $\mathfrak{p}_-$ are monoidal 2-functors (Proposition 20). For example, by (strict) associativity we have that $\otimes(\mathrm{id} \times \otimes) = \otimes(\otimes \times \mathrm{id})$ in $\mathbf{Cat}$. We get the desired equations by applying the embeddings:

$$\mathfrak{p}^{\otimes} \circ (\mathfrak{p}^{\otimes} \times \mathrm{id}) = \mathfrak{p}^{\otimes} \circ (\mathrm{id} \times \mathfrak{p}^{\otimes}) \quad \texttt{assoc},$$
$$(\mathrm{id} \times \mathfrak{p}_{\otimes}) \circ \mathfrak{p}_{\otimes} = (\mathfrak{p}_{\otimes} \times \mathrm{id}) \circ \mathfrak{p}_{\otimes} \quad \texttt{coassoc}.$$

It remains to show that the rules in Figure 2 are preserved. These are the only rules with a non-trivial internal structure. Observe that all these rules are either of the form

$$(\mathfrak{p}^f, \mathrm{id}_{f\beta}) \circ \sigma \simeq f\sigma \circ (\mathfrak{p}^f, \mathrm{id}_{f\alpha}) \qquad \text{or} \qquad \sigma \circ (\mathfrak{p}_f, \mathrm{id}_{f\beta}) \simeq (\mathfrak{p}_f, \mathrm{id}_{f\alpha}) \circ f\sigma$$

for some functor $f : \omega \to \tau$ and some morphism $\sigma : \alpha \to \beta$. We show the isomorphism on the left. First, at the level of profunctors the isomorphism holds since hom-functors are the identities. It remains to show that $[\sigma, \mathrm{id}_{f\beta}] \sim [\mathrm{id}_{f\alpha}, f\sigma]$ under this isomorphism. To this end, note that the left-hand side evaluates via the isomorphism

$$\int^{\gamma \in \omega} \omega(\alpha,\gamma) \times \omega(f\gamma, f\beta) \simeq \omega(f\alpha, f\beta) \qquad [h,k] \mapsto k \circ fh$$

to $f\sigma$. Similarly, the right-hand side evaluates via the isomorphism

$$\int^{\gamma\in\tau} \omega(f\alpha,\gamma)\times\omega(\gamma,f\beta)\simeq\omega(f\alpha,f\beta) \qquad [h,k]\mapsto k\circ h$$

also to $f\sigma$, whence the desired identification follows. The argument for the rules of the second form is dual. □

The following proposition shows that we can detect properties of monoidal categories in a layered prop. This observation is not relevant for the examples that we discuss in this work, yet it is important for the development of the general theory of layered props.

Proposition 24. *If both $\tau,\omega\in\Omega$ are monoidal closed (*resp. coclosed*) and $f:\omega\to\tau$ in Ω is also monoidal closed (*resp. coclosed*), then the interpretation $\mathcal{I}$ preserves the 2-cell* `C` *(*resp. `coC`*) in Figure 8.*

Proof. For `C`, we have to show that

$$\mathfrak{p}^{\otimes}\circ(\mathfrak{p}_f\times\mathrm{id})\simeq\mathfrak{p}_f\circ\mathfrak{p}^{\otimes}\circ(\mathrm{id}\times\mathfrak{p}^f).$$

Both profunctors are of the type $\tau\times\omega \relbar\joinrel\mapstochar\joinrel\rightarrow \omega$. Let us compute both sides on the triple of objects (D,C,C'). The right-hand side computes to

$$\int^{B,E\in\tau}\tau(fC,B)\times\tau(D\otimes B,E)\times\tau(E,FC')\simeq\tau(D\otimes FC,FC'),$$

while in order to reduce the left-hand side we use the monoidal closed structure:

$$\begin{aligned}\int^{A\in\omega}\tau(D,FA)\times\omega(A\otimes C,C')&\simeq\int^{A\in\omega}\tau(D,FA)\times\omega(A,[C,C'])\\&\simeq\tau(D,F[C,C'])\\&\simeq\tau(D,[FC,FC'])\\&\simeq\tau(D\otimes FC,FC').\end{aligned}$$

Since these agree and all isomorphisms are natural, we have the desired isomorphism. The argument for `coC` is dual, using that the categories and functors are monoidal coclosed. □

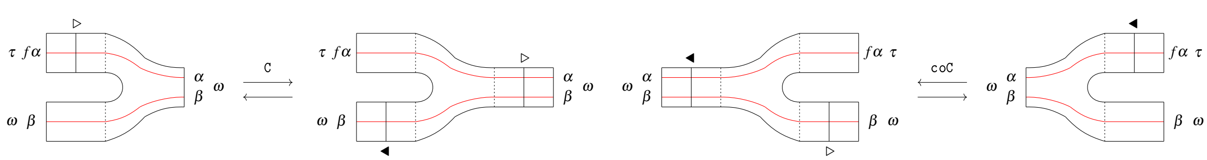


Figure 8: Monoidal (co)closure equations.

Unification of Modal Logic via Topological Categories

Lingyuan Ye
Tsinghua University
Beijing, China
ye.lingyuan.ac@gmail.com

In this paper we provide a unifying description of different types of semantics of modal logic found in the literature via the framework of topological categories. In the style of categorical logic, we establish an exact correspondence between various syntactic extensions of modal logic on one hand, including modal dependence, group agent structures, and logical dynamic, and semantic structures in topological categories on the other hand. This framework provides us a uniform treatment of interpreting these syntactic extensions in all different types of semantics of modal logic, and it deepens our conceptual understanding of the abstract structure of modal logic.

1 Introduction

Throughout the history of modal logic, many different types of semantics have been developed to interpret the modal language, with various applications in mind. Starting from the seminal work by von Wright [24] and the later extension by Hintikka in [14], the Kripkean style semantics of modal logic has been widely applied in the philosophical study of epistemology. Tarski and McKinsey in [19] have also discovered that the interior operator induced by a topological space could be used to interpret modal formulas as well, which naturally finds its connection with propositional intuitionistic logic. Other variations include neighbourhood semantics for modal logic, first suggested by Scott in [22] in order to study certain non-normal fragments of modal logic. Finally, we also have semantics of a more algebraic flavour, extending the usual algebraisation of propositional logic using Boolean algebras.

These various forms then naturally bear the following question: Is it possible to provide a *unifying* description of all types of semantic models of modal logic? To provide a positive answer, this paper starts with the following observation: In all of the above mentioned examples, in fact in many more cases, the categories of semantics of modal logic all organise themselves into *topological categories* (over **Set**).

The notion of a topological category is introduced in [1], with the aim of axiomatising the structure of those categories containing objects X equipped with certain *geometric data*, with X living in an ambient category $\mathbf{X}$. This results in the notion of topological categories over an arbitrary base $\mathbf{X}$. For our purpose though, we will exclusively work over **Set**, and this is our default for topological categories henceforth. The prototypical example is **Top**, the category of topological spaces, whose objects are sets equipped with a topology. We will give an overview of topological categories in Section 2, and provide another equivalent way of describing topological categories more suitable for modal logic (cf. Theorem2.5). According to this theorem, it can then be immediately recognised that all the mentioned examples of semantics conform to such a description: Kripke models are sets equipped with a binary relation, which are often depicted diagrammatically. We've already mentioned topological spaces, and neighbourhood models are no exceptions. Perhaps surprisingly, a particular style of algebraic semantics, using *complete atomic Boolean algebra with operators* (CABAO), can also be recognised as topological or geometrical over **Set**, once we take its dual category. This is arguably an incarnation of the duality principle between algebra and geometry within the context of modal logic. We will prove in Proposition 2.6 that all these

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 383–399, doi:10.4204/EPTCS.380.22

types of semantics, and in fact much more, are instances of topological categories, hence building the foundations of unification.

But such fact alone is far from convincing that this is a good framework for unifying modal logic. The more important topic is how the semantic structures of topological categories would explain the various logical features that are present in a modal context. In this paper, we will follow the philosophy of categorical logic, establishing *exact correspondences* between different syntactic patterns of modal logic with semantic structures of topological categories. Such correspondences are witnessed by considering transformation of models, viz. functors between topological categories.

The first thing to explain is the interpretation of modalities. As we will see in more detail in Section 3, it is precisely the geometric data of a topological category that is responsible for its interpretation. Furthermore, the structure of topological categories also connects tightly with many other *extensions* of basic modal logic studied in the literature, including the multi-agency, group agency, modal dependence, logical dynamics, etc.. For each of these reasoning patterns we have established theorems (see Theorem 3.4, 4.4 and 5.3), showing that functors preserve certain structures of topological categories if, and only if, the linguistic interpretation of the corresponding fragment of modal logic remains unchanged under the transformation. These results significantly improve our conceptual understanding of modal logic, and will be the main topics of Section 4 and 5.

To the best knowledge of the author, in the current literature there has been no theoretic framework to enable all these different fragments of modal logic to be described in a uniform way for all types of semantics. Our systematic approach allows seamless generalisation of all these constructions in modal logic to any other semantics. For instance, it has been actively discussed what is the corresponding notion of common knowledge in topological semantics [4], how to extend different forms of logical dynamics to wider contexts [5], or how to develop modal dependence described in [3] and [2] for other semantic types. Our work provides a novel answer to all these different questions by accommodating them to the framework of topological categories, and it has ample potential applications.

2 Preliminaries

In many existing texts, e.g. in [1, 15], topological categories are usually introduced as *fibrations* over **Set** satisfying certain lifting properties. It is well-known from the Grothendieck construction that fibrations can be equivalently described by *indexing categories*, or functors mapping out of **Set**. For our purpose, it is this equivalent *indexing* point of view of topological categories that is more suitable for making connections with modal logic. We will discuss this in more detail below.

Recall that a *concrete category*, or a *construct*, is simply a faithful functor $U : \mathscr{A} \to \mathbf{Set}$. When it is clear from the context what the functor U is, we will simply refer to $\mathscr{A}$ as a concrete category.

Example 2.1. We take this opportunity to introduce the main examples of category of semantics:

- **Kr** denotes the category of Kripke frames, whose objects are sets equipped with a binary relation on them, with morphisms being monotone maps. It has certain useful full subcategories including **Pre** and **Eqv**, whose objects only contains preorders or equivalence relations.
- We've mentioned that **Top** will denote the category of topological spaces.
- **Nb** is the category of neighbourhood frames, whose objects are sets X equipped with a neighbourhood relation $E \subseteq X \times \wp(X)$, and whose morphisms $f : (X,E) \to (Y,F)$ are functions from X to Y satisfying a continuity condition: For any $x \in X$ and $V \subseteq Y$, $fxFV \Rightarrow xEf^{-1}V$.

- We let objects of **CABAO** be pairs (X,m) with m being an arbitrary endo-function on $\wp(X)$, and morphisms $f:(X,m)\to(Y,n)$ are functions from X to Y satisfying $f^{-1}\circ n\subseteq m\circ f^{-1}$, where we extend the order $\subseteq$ on $\wp(X)$ point-wise to the function space $\wp(X)^{\wp(Y)}$.[1]
- Besides models of the above form, to interpret modal formulas we also need evaluation functions to interpret propositional letters. For a fixed set P of propositional variables, we introduce the category **Evl** of evaluations, whose objects are pairs (X,V) with $V:\mathsf{P}\to\wp(X)$, and morphisms $f:(X,V)\to(Y,W)$ are functions from X to Y satisfying $V\subseteq f^{-1}\circ W$, where similarly the order is the point-wise extension of the subset relation on the function space $\wp(X)^{\mathsf{P}}$.

In each case, there is an evident forgetful functor to **Set** that identifies them as concrete categories. $\Diamond$

Let us say a few more words on the category **CABAO**. From a well-known theorem of Tarski, we know every CABA is isomorphic to a power set algebra $\wp(X)$ (and every power set algebra is a CABA), and every morphism between them is of the form $f^{-1}:\wp(Y)\to\wp(X)$ for some function $f:X\to Y$. Hence, our definition of a CABAO as a pair (X,m) does not lose anything, and it builds in the duality, since it uses f, rather than f^{-1}, as morphisms. Notice that the morphisms we choose between CABAOs are not the *algebraic* ones, which should commute with the operators on both sides, but *lax* ones that only require an inequality. A possible intuition for this choice is to read the operators m,n as interior operators induced by a topology, and the above continuity condition is exactly saying that f is a continuous map for the two topological spaces. We will see later that such a choice makes **CABAO** topological over **Set**.

There is also an accompanying notion of *concrete functors* between concrete categories: A functor F between two concrete categories $(\mathscr{A},|-|_{\mathscr{A}})$ and $(\mathscr{B},|-|_{\mathscr{B}})$ is a concrete functor iff it commutes with the forgetful functors, i.e. iff it preserves the underlying sets. Obviously, each forgetful functor of $|-|$ of a construct $\mathscr{A}$ constitute a concrete functor from $(\mathscr{A},|-|)$ to $(\mathbf{Set},1_{\mathbf{Set}})$, which establish **Set** as the terminal object in the (large) category of concrete categories and concrete functors.

The faithfulness of the forgetful functor of a concrete category has many consequences. For any construct $(\mathscr{A},|-|)$, we will identify the Hom-sets $\mathscr{A}(A,B)$ simply as subsets of $\mathbf{Set}(|A|,|B|)$, and say a function $f:|A|\to|B|$ is an *$\mathscr{A}$-morphism* if it belongs to $\mathscr{A}(A,B)$. For instance, f is a **Top**-morphism if it is continuous. Faithfulness of $|-|$ also implies that each fibre $\mathscr{A}_X$ over a set X is a (possibly large) preorder — recall that a morphism in $\mathscr{A}_X$ is a morphism in $\mathscr{A}$ above id_X. If each fibre is indeed small, then we say the construct $\mathscr{A}$ is *fibre-small*. It is easy to verify that all the introduced categories in Example 2.1 have small fibres. All the constructs considered in the future will be fibre-small.

As mentioned, topological categories are constructs that satisfy certain lifting properties. For any construct $(\mathscr{A},|-|)$, a *structured source* is defined to be a set of functions of the form $\{f_i:X\to|A_i|\}_{i\in I}$,[2] where each $A_i\in\mathscr{A}$. An *initial lift* of such a structured source is an object A in the fibre $\mathscr{A}_X$, satisfying the following universal properties: For any function $g:|B|\to|A|$, g is an $\mathscr{A}$-morphism iff $f_i\circ g:|B|\to|A_i|$ is an $\mathscr{A}$-morphism for any $i\in I$. Evidently, initial lifts are identified up to isomorphisms in the fibre $\mathscr{A}_X$.

Definition 2.2 (Topological Categories)**.** A construct $(\mathscr{A},|-|)$ is a *topological category* if every structured source has a *unique* initial lift.

We can break the definition of a topological category into two parts: It first requires the *existence* of initial lifts of structured sources, and it also requires the *uniqueness* of such lifts. The notion of initial lift of structured source is a generalisation of cartesian lifts for Grothendieck fibrations. In fact, cartesian lift is exactly initial lift for a *singleton structured source*, viz. a structured source consisting of only one

[1] This definition of the category **CABAO** contains certain subtle points, which we will explain in a minute.

[2] If we don't restrict to fibre-small constructs, then we need to consider structured sources whose size are *proper classes*. However, this is not a problem for us to worry about. We refer the readers to [1] for more details.

function. This in particular suggests that topological categories are special types of *fibrations* where we can perform lifts against an arbitrary set of morphisms with a common codomain. Together with the uniqueness part of the definition, a topological category satisfies many desirable properties:[3]

Lemma 2.3. *If $\mathscr{A}$ is a topological category, then each fibre $\mathscr{A}_X$ is a complete lattice for any set X.*

Proof. For any family $\{A_i\}_{i\in I}$ in the fibre $\mathscr{A}_X$, consider the structured source, $\{1_X : X \to |A_i|\}_{i\in I}$. It is routine to verify that its unique initial lift is precisely the meet of this family in $\mathscr{A}_X$. □

The existence of initial lifts guarantees each fibre to be complete preorders, and the uniqueness then implies that they are indeed posets. As a fibration, given any function $f : X \to Y$, the initial lifts along f will induce functions of the form $f^* : \mathscr{A}_Y \to \mathscr{A}_X$. Again, f^* being a well-defined function is guaranteed by the uniqueness of initial lifts, and we will also denote maps of the form f^* as *pullback maps*. Furthermore, uniqueness also suggests that the fibration *splits*, in the sense that $1_X^* = 1_{\mathscr{A}_X}$ and $g^*f^* = (gf)^*$. The more important observation is that each pullback map preserves meets in the fibre:

Lemma 2.4. *Let $(\mathscr{A}, |-|)$ be a topological category, then for any function $f : X \to Y$, the pullback map $f^* : \mathscr{A}_Y \to \mathscr{A}_X$ preserves arbitrary meets.*

Proof. For any family $\{B_i\}_{i\in I}$ in $\mathscr{A}_Y$, we only need to prove $\bigwedge_{i\in I} f^*B_i \le f^* \bigwedge_{i\in I} B_i$. By definition, this holds iff the identity function, viewed as a map $1_X : |\bigwedge_{i\in I} f^*B_i| \to |f^* \bigwedge_{i\in I} B_i|$, is an $\mathscr{A}$-morphism. By the universal property of initial lift, it is so iff $f \circ 1_X = f : |\bigwedge_{i\in I} f^*B_i| \to |\bigwedge_{i\in I} B_i|$ is an $\mathscr{A}$-morphism, and again, this is furthermore equivalent to all the maps in the structured source $\{f : |\bigwedge_{i\in I} f^*B_i| \to |B_i|\}$ being $\mathscr{A}$-morphisms. However, we know that $\bigwedge_{i\in I} f^*B_i \le f^*B_i$ for any $i \in I$, which means both $1_X : |\bigwedge_{i\in I} f^*B_i| \to |f^*B_i|$ and $f : |f^*B_i| \to |B_i|$ are $\mathscr{A}$-morphisms, hence so is the composite. □

It follows that each pullback map f^* has a unique left adjoint, which we denote as $f_!$ and call it the *pushforward map*. By the adjunction $f_! \dashv f^*$ and the universal property of initial lift, it is easy to see that $f_!$ are exactly describing the cocartesian lifts, which makes a topological category an *opfibration* as well, hence a *bifibration*. As Theorem 2.5 will show, the data of fibres and pullback or pushforward maps uniquely determines a topological category:

Theorem 2.5. *Let* **InfL** *(resp.* **SupL***) be the category of inflattices (suplattices).*[4] *Recall that they are canonically dual to each other. The data of a topological category $(\mathscr{A}, |-|)$ is the same as the data of a functor $\mathscr{A}_{(-)} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{InfL}$, or equivalently $\mathscr{A}_- : \mathbf{Set} \to \mathbf{SupL}$.*

Proof. We've already shown that a topological category induces a functor from $\mathbf{Set}^{\mathrm{op}}$ to **InfL**. On the other hand, since **InfL** is a subcategory of **Cat**, any functor $F : \mathbf{Set}^{\mathrm{op}} \to \mathbf{InfL}$ admits a Grothendieck construction, resulting in a fibration $p : \mathscr{F} \to \mathbf{Set}$. The objects of $\mathscr{F}$ are pairs (X,A) with A being an element in $F(X)$; a morphism $f : (X,A) \to (Y,B)$ is a function $f : X \to Y$, such that $A \le Ff(B)$. The forgetful functor p is evident. To this end, we only need to verify that for arbitrary structured source $\{f_i : X \to p(X_i,A_i)\}_{i\in I}$, it has a unique initial lift, which we claim is given by $\bigwedge_{i\in I}(Ff_i)(A_i)$ over X. For any function $g : p(Y,B) \to p(X, \bigwedge_{i\in I}(Ff_i)(A_i))$, by definition it is an $\mathscr{F}$-morphism iff

$$B \le Fg \bigwedge_{i\in I}(Ff_i)(A_i) = \bigwedge_{i\in I} F(f_i \circ g)(A_i) \Leftrightarrow \forall i \in I[F(f_i \circ g)B \le A],$$

which exactly means that all $f_i \circ g : p(Y,B) \to p(X_i,A)$ are $\mathscr{F}$-morphisms. Hence, $(\mathscr{F}, p)$ is a topological category, and we leave the readers to verify that the above two processes are mutually inverse. □

[3] The following two lemmas are both contained in [1]. We include the proof here for the convenience of the readers.

[4] **InfL** (resp. **SupL**) is the category of complete lattices with meet (resp. join) preserving maps. For more detailed description of various categorical structures on **InfL** or **SupL**, we refer the readers to [16, Chapter I].

Proposition 2.6. *All the categories of semantics mentioned in Example 2.1 are topological categories.*

Proof Sketch. It is evident that all the fibres of those mentioned examples are complete lattices. We only describe in each case how the pullback or pushforward maps are constructed, and trust the readers to verify the universal properties and functoriality. Given a function $f : X \to Y$:

- In **Kr**, f^* lifts a relation R on Y to the largest relation in X such that f is monotone, i.e. for any $x, x' \in X$, $(x, x') \in f^*R$ iff $(fx, f'x) \in R$. The pullback maps in **Pre**, **Eqv** are inherited from **Kr**.
- In **Top**, the pullback f^* maps a topology γ on Y to the so-called weak topology on X, i.e. $U \in f^*\gamma$ iff there exists $V \in \gamma$ that $U = f^{-1}(V)$.
- In **Nb**, the description of f^* is similar to that in **Top**. For a neighbourhood relation F on Y, the lift f^*F satisfies that $(x, U) \in f^*F$ iff there exists $V \subseteq Y$ that $U = f^{-1}(V)$ and $(fx, V) \in F$.
- In **CABAO**, it is easier to describe the pushforward maps. Given any endo-function m on $\wp(X)$, its pushforward is the operator $\forall_f \circ m \circ f^{-1}$ on Y, where $\forall_f$ is the right adjoint of f^{-1}.
- In **Evl**, evidently the pullback f^* is obtained by post-composing with f^{-1}. □

At this point, we have accomplished our first goal to recognise all the instances of semantics in Example 2.1 as topological categories. We end this section by describing the *product* construction:

Definition 2.7 (Product of Topological Categories)**.** For any family $\{\mathscr{A}_i\}_{i\in I}$ of topological categories viewed as functors $\{\mathscr{A}_i : \mathbf{Set}^{\mathrm{op}} \to \mathbf{InfL}\}_{i\in I}$, their product $\prod_{i\in I} \mathscr{A}_i$ is given as the following composition,

$$\mathbf{Set}^{\mathrm{op}} \xrightarrow{\prod_{i\in I}\mathscr{A}_i} \prod_{i\in I}\mathbf{InfL} \xrightarrow{\bigoplus_{i\in I}} \mathbf{InfL}.$$

The functor $\bigoplus_{i\in I}$ is the biproduct functor on **InfL**, which takes a family of inflattices to its set-theoretic product with entry-wise order. In other words, the fibre $(\prod_{i\in I} \mathscr{A}_i)_X$ of a product is simply the product of the fibres $\prod_{i\in I} (\mathscr{A}_i)_X$. It is easy to verify that $\prod_{i\in I} \mathscr{A}_i$ is indeed their categorical product in the category of concrete categories and concrete functors. The product construction for instance allows us to combine a Kripke model with an evaluation function by looking at $\mathbf{Kr} \times \mathbf{Evl}$, or to consider a family of models by introducing $\mathscr{A}^\Sigma$ for any set Σ, which is the Σ-indexed product of $\mathscr{A}$ with itself.

3 Interpreting Modalities via Geometric Data

In this section, we will see how the categorical structure we have described in Section 2 would unify the *interpretation of modalities* in each different types of semantics. We start by briefly recalling the very basics of the modal language and its interpretation; standard references include [11, 7]. Let a non-empty set Σ serve as the signature, and let P be a non-empty set of propositional variables. The modal language $\mathscr{L}_\Sigma$ over the signature Σ and the variable set P is the smallest set of formulas containing P and closed under forming conjunctions, negations, and adding modalities $\Box_a$ for all $a \in \Sigma$. When Σ is a singleton, we will omit the subscript, and $\mathscr{L}$ denotes the usual modal language with a single modality. We will refer to it as the *basic modal language*. Other logical connectives are viewed as defined notions.

In any set-based semantics of modal logic, the classical propositional connectives are always interpreted by the Boolean operations on the power set algebra. From an algebraic point of view, the interpretation of the additional modality, in its most general form, should be given by an arbitrary *endo-function on the power set*, which is exactly the structure of a CABAO. Hence, we define the structure of a *semantic functor* to provide the interpretation of basic modal language:

Definition 3.1 (Semantic Functor and Modal Category)**.** Let $(\mathscr{A}, |-|)$ be a topological category. A *semantic functor* on $\mathscr{A}$ is a concrete functor $(-)^+ : \mathscr{A} \to \mathbf{CABAO}$. A *modal category* is then a topological category together with a semantic functor.

For any modal category $\mathscr{A}$ with semantic functor $(-)^+$, we recursively define the interpretation of modal formulas as follows: For any set X and any pair (A,V) in $(\mathscr{A} \times \mathbf{Evl})_X$,

$$[\![p]\!]_A^V = V(p), \quad [\![\varphi \wedge \psi]\!]_A^V = [\![\varphi]\!]_A^V \cap [\![\psi]\!]_A^V, \quad [\![\neg\varphi]\!]_A^V = X \backslash [\![\varphi]\!]_A^V, \quad [\![\Box\varphi]\!]_A^V = A^+([\![\varphi]\!]_A^V).$$

We may also define the more familiar *local* version of semantics, and write $A,V,x \models \varphi$ whenever $x \in [\![\varphi]\!]_A^V$. Evidently, the identity functor on **CABAO** establishes itself as a modal category. We see below that all other categories of semantics mentioned previously have modal category structures:

Proposition 3.2. *There exist* fully faithful *modal functors on* **Kr**, **Pre**, **Eqv**, **Top** *and* **Nb** *that embeds them into* **CABAO***, inducing the usual semantics of modal logic.*

Proof Sketch. Again, we only describe the construction of semantic functors in each case, and trust the readers to verify their fully faithfulness:

- Recall for any relation $R \subseteq X \times Y$, there exists an induced operator $\forall_R : \wp(X) \to \wp(Y)$, such that for any $S \subseteq X$, $\forall_R(S) = \{\, y \in Y \mid \forall x[xRy \Rightarrow x \in S] \,\}$. We then construct the embedding $\mathbf{Kr} \hookrightarrow \mathbf{CABAO}$ by sending each relation R in fibre $\mathbf{Kr}_X$ to the operator $\forall_{R^\dagger}$, where $R^\dagger$ is the dual relation of R. The semantic functors on **Pre** and **Eqv** are inherited from the one on **Kr**.
- For **Top**, it sends each topology τ on a set X to the interior operator j_τ it induces.
- For **Nb**, it assigns E in $\mathbf{Nb}_X$ to n_E, such that $n_E(S) = \{\, x \mid (x,S) \in E \,\}$ for any $S \subseteq X$. □

Proposition 3.2 then completes our categorical unification of all the mentioned types of semantics on how they interpret the basic modal language. Clearly, our approach of given in Definition 3.1 closely relates to the spirit of *algebraic semantics* of modal logic. But one additional insight our categorical framework suggests is an even closer connection between these different types of semantics with modal algebras via Proposition 3.2, in that the single notion of continuous morphisms between CABAOs as defined in Example 2.1 explains all the different types of morphisms in these topological categories, by identifying them as *full subcategories* of **CABAO**.

Intuitively, it is precisely the semantic functor that provides the interpretation of modalities in all cases, but we can establish the correspondence in a more formal way, by considering *transformation of models* as mentioned in Section 1. We define when a concrete functor between two modal categories interacts well with a specific fragment of modal logic:

Definition 3.3 (Preservation of Language)**.** Let $\mathscr{A}, \mathscr{B}$ be two modal categories, which both support the interpretation of certain fragment of modal language $\mathscr{L}_0$ which extends $\mathscr{L}$. We say a concrete functor $F : \mathscr{A} \to \mathscr{B}$ *preserves the interpretation of the language* $\mathscr{L}_0$, if the following happens: For (A,V) in $(\mathscr{A} \times \mathbf{Evl})_X$ over some set X and for any formula $\varphi \in \mathscr{L}_0$, we have $[\![\varphi]\!]_A^V = [\![\varphi]\!]_{FA}^V$.

In other words, a concrete functor F preserves the interpretation of a language $\mathscr{L}_0$ iff the evaluation of each formula in $\mathscr{L}_0$ remains unchanged when we apply the transformation F. As a first example of establishing an exact correspondence between a semantic structure and a particular syntactic pattern, we prove the following theorem:

Theorem 3.4. *For any concrete functor $F : \mathscr{A} \to \mathscr{B}$ between two modal categories $(\mathscr{A}, (-)^+_{\mathscr{A}})$ and $(\mathscr{B}, (-)^+_{\mathscr{B}})$, it commutes with the two semantic functors iff it preserves the interpretation of $\mathscr{L}$.*

Proof. Suppose F does not commute with the two semantic functors, then for some object A in $\mathscr{A}$ over some set X, $(A)^+_{\mathscr{A}}$ and $(FA)^+_{\mathscr{B}}$ would not agree. This means that the two operators on $\wp(X)$ do not coincide, which implies they must not coincide on some subset $S \subseteq X$. Consider the simple formula $\Box p$, and an evaluation function V that assigns p to S. By definition, $[\![\Box p]\!]^V_A$ and $[\![\Box p]\!]^V_{FA}$ will *not* be the same.

The proof of the only if direction is obviously by induction on the structure of formulas, and the only interesting case is the one involving modalities. Since F is assumed to be a modal functor, we must have $(A)^+_{\mathscr{A}} = (FA)^+_{\mathscr{B}}$ for any A in $\mathscr{A}$, which means that the interpretation of the modalities by A through $(-)^+_{\mathscr{A}}$ and by FA through $(-)^+_{\mathscr{B}}$ are identical, which suffices for the inductive proof. □

Theorem 3.4 provides the precise formal content of what we mean informally by the correspondence between the syntactic structure of modalities and the semantic structure of semantic functors of a modal category. And henceforth, we will refer to those concrete functors between two modal categories which commutes with the semantic functors on both sides as *modal functors*. There are already many interesting examples of modal functors we can explore, and below we only list a few:

Example 3.5. Here we list some interesting examples of model transformations between the modal categories we have introduced so far:

- By definition, any modal category has a unique modal functor mapping into **CABAO**, which makes it the terminal object in the category of modal categories and modal functors.
- Since the semantic functors in **Pre** and **Eqv** are induced by the one in **Kr**, the embeddings **Eqv** $\hookrightarrow$ **Pre** and **Pre** $\hookrightarrow$ **Kr** are both modal functors.
- There is a modal embedding **Pre** $\hookrightarrow$ **Top**, assigning a preorder its Alexandroff topology.
- In fact, we can show that **Nb** is isomorphic to **CABAO**, which means that all the above examples has a modal embedding into **Nb** as well.

It is also instructive to look at counter-examples of modal functors. It turns out, the above modal embeddings all have either a left or a right adjoint, and these adjoints are usually *not* modal embeddings with respect to the semantic functors we have constructed in Proposition 3.2:

- We have both a left and a right adjoint **Pre** $\rightrightarrows$ **Eqv** for the modal embedding **Eqv** $\hookrightarrow$ **Pre**, sending a preorder to the smallest equivalence relation containing it and the least one it contains. These adjoints do not commute with the semantic functors since they change the relation. Similarly, there is a left adjoint **Kr** $\to$ **Pre** sending a relation to its preorder closure, which isn't modal either.
- The embedding **Pre** $\hookrightarrow$ **Top** has a right adjoint **Top** $\to$ **Pre**, sending a topological space to its specialisation order, but this construction does not preserve the information of all open neighbourhoods of a point, hence it is also not modal. ♢

However, the mere syntactic structure of a modality, arguably, has not too much to do with the rich structure of topological categories we have seen in Section 2. In fact, the notion of semantic functors and modal categories in Definition 3.1 can indeed be stated more generally for concrete categories, not only for topological ones. The true usage of the full structure of topological categories emerges when we consider further syntactic extensions of modal logic, which are the topics of the next two sections.

4 Modal Strength, Group Knowledge and Fibre Structure

In this section, we will proceed to study the extension of multi-agent fragment of modal logic, with explicit syntactic comparison of *modal strength*, or *dependence relation*, between different modalities, and forming *group agents*. Recent works [3, 2] put dependence purely in modal terms, but they have only considered the relational and topological contexts. How to form group agents is also an active topic for current research on modal logic and collective agency [13, 23], but almost all approaches focus on a single type of models. In both cases, our categorical approach allows a unifying description for all types of semantics, which is one of the main benefit. Our ultimate goal is again to identify an exact correspondence between these syntactic patterns with certain semantic structures of topological categories, with formal content similar to that of Theorem 3.4.

Let's first look at the simple extension of a multi-modal language, i.e. when the indexed set Σ is not a singleton. There will be different modalities $\Box_a, \Box_b, \cdots$ with $a, b \in \Sigma$ in the language $\mathscr{L}_\Sigma$. It should be straight forward to recognise that the multi-agent fragment $\mathscr{L}_\Sigma$ are related to taking the *products* of topological categories. Given any modal category $\mathscr{A}$, recall that we use $\mathscr{A}^\Sigma$ to denote the Σ-indexed self-product of $\mathscr{A}$. Any semantic functor $(-)^+$ on $\mathscr{A}$ naturally extends to one from $\mathscr{A}^\Sigma$ to $\mathbf{CABAO}^\Sigma$, which by an abuse of notation we also denote as $(-)^+$: Given any object $(A_a)_{a\in\Sigma}$ in the fibre $\mathscr{A}^\Sigma_X$, which by our construction in Definition 2.7 is simply a Σ-indexed tuple of objects in the fibre $\mathscr{A}_X$, we have $(A_a)^+_{a\in\Sigma} = (A_a^+)_{a\in\Sigma}$. The Σ-indexed tuple $(A_a^+)_{a\in\Sigma}$ is then expected to provide the interpretation of each modality $\Box_a$ in the language $\mathscr{L}_\Sigma$ for any $a \in \Sigma$, using the corresponding object A_a^+. Intuitively, different modalities correspond to different objects in the same fibre of a topological category. Hence, given any $((A_a)_{a\in\Sigma}, V)$ in the fibre $(\mathscr{A}^\Sigma \times \mathbf{Evl})_X$, we may change the clause of modalities in the recursive definition of evaluation of formulas to $[\![\Box_a\varphi]\!]^V_{(A_a)_{a\in\Sigma}} = (A_a)^+([\![\varphi]\!]^V_{(A_a)_{a\in\Sigma}})$, to interpret $\mathscr{L}_\Sigma$.

However, in the language $\mathscr{L}_\Sigma$, we treat different modalities as different individuals, and do not consider the possible relations between different modalities. But we do have a meaningful way comparing them, since semantically they denote different objects within the same fibre of a topological category $\mathscr{A}$, and there is a canonical order in each fibre $\mathscr{A}_X$. It turns out, this partial order within each fibre signifies the *modal strength* of different modalities. Explicitly, suppose we have two objects A, B in the fibre $\mathscr{A}_X$ that $A \leq B$. The semantic functor then gives us two operators $m_A \leq m_B$ in $\mathbf{CABAO}_X$, which, according to our definition of morphisms in **CABAO**, actually means $m_B \subseteq m_A$.

In different contexts, the modal strength relation has various incarnations. For instance, in epistemic or doxastic logic, we read the modal formula $\Box_a\varphi$ as agent-a knows or believes φ (cf. [10]). Now if we have $A_a \leq A_b$ in the fibre $\mathscr{A}_X$, the above induced two modalities satisfying $m_b \subseteq m_a$ would actually suggest that there is an *epistemic dependence* between the two agents' knowledge or belief: Whenever b knows some proposition at state $x \in X$, viz. $x \in m_b([\![\varphi]\!])$, a also knows it at that state, because $x \in m_b([\![\varphi]\!]) \subseteq m_a([\![\varphi]\!])$. In other applications, such modal strength comparison would mean something else.

This observation motivates us to add such comparison of modalities explicitly into our syntax, in the form of *dependence atoms*. For any $a, b \in \Sigma$, we could add an atomic proposition K_ab into our language, with the intuitive reading of K_ab as stating the modality denoted by a lies below the one denoted by b. We refer to this extended language as $\mathscr{L}^D_\Sigma$. But to interpret such dependence atoms as predicates, we need the following *local* version of strength orders between two operators on the same power set algebra:

Definition 4.1. For any two operators m, n in $\mathbf{CABAO}_X$ and any $U \subseteq X$, we say *m locally depends on n in U*, denoted as $m \subseteq_U n$, if for any $S \subseteq X$ and any $x \in U$, $x \in m(S) \Rightarrow x \in n(S)$.

In this way, the global relation $m \subseteq n$ is the same as $m \subseteq_X n$. When U is a singleton $\{x\}$, we simply write $m \subseteq_x n$. The following observation is crucial for us to define the interpretation of the dependence atoms:

Lemma 4.2. *For any m, n in $\mathbf{CABAO}_X$, there is a maximal subset U that $m \subseteq_U n$.*

Proof. By definition, for the empty set $\emptyset$ we always have $m \subseteq_\emptyset n$, since the universal quantification $\forall x \in \emptyset$ is vacuous. Furthermore, local dependence is closed under taking unions, since it is trivial to note that $m \subseteq_{\bigcup_{i\in I} U_i}$ iff for any $i \in I$, $m \subseteq_{U_i} n$. Thus, the maximal subset U is given by $\{\, x \mid m \subseteq_x n \,\}$. $\Box$

Given an object $(A_a)_{a\in\Sigma}$ in the fibre $\mathscr{A}_X^\Sigma$, the interpretation $[\![K_a b]\!]$ of the newly added dependence atoms should now be defined as the maximal subset U of X, such that $A_b^+ \subseteq_U A_a^+$ holds. This is exactly how the dependence atoms are interpreted in any topological categories. We might also give the local version of the truth condition, $(A_a)_{a\in\Sigma}, x \models K_a b$ iff $A_b^+ \subseteq_x A_a^+$. Notice that the interpretation of $K_a b$ is *independent* from the choice of the evaluation function V on X. We may look at the concrete meaning of such dependences in all the remaining examples we have considered so far:

Example 4.3. We list here how local dependence looks like in each exemplar modal category:

- In **Kr**, **Pre** and **Eqv**, given relations $(R_a)_{a\in\Sigma}$ on X, we have $(R_a)_{a\in\Sigma}, x \models K_a b$ iff $R_a[x] \subseteq R_b[x]$. In the epistemological interpretation, this means agent-a's uncertainty locally at x is less than b's.
- In **Top**, given topologies $(\tau_a)_{a\in\Sigma}$ on X, $(\tau_a)_{a\in\Sigma}, x \models K_a b$ iff $1_X : (X, \tau_a) \to (X, \tau_b)$ is *locally continuous* at x. This relates to the continuity view of epistemic dependence discussed in [2].
- In **Nb**, given neighbourhoods $(E_a)_{a\in\Sigma}$ on X, $(E_a)_{a\in\Sigma}, x \models K_a b$ iff $E_b[x] \subseteq E_a[x]$. In evidence based logic, this interprets as the evidence set of b's is contained in that of a's locally at x (cf. [9]). ♢

In this case, preserving the interpretation of the multi-agent modalities and the dependence atoms does not require anything else than being a modal functor:

Theorem 4.4. *For any concrete functor $F : \mathscr{A} \to \mathscr{B}$ between two modal categories, it preserves the interpretation of $\mathscr{L}_\Sigma^D$ iff it preserves the interpretation of $\mathscr{L}$.*

Proof. The only if part is trivial, since $\mathscr{L}_\Sigma^D$ is an extension of $\mathscr{L}$. For the if part, by Theorem 3.4 we know F must be a modal functor. This implies that for any $a \in \Sigma$ and any tuple $(A_a)_{a\in\Sigma}$ in $\mathscr{A}^\Sigma$, we must have $(A_a)^+_{\mathscr{A}} = (FA_a)^+_{\mathscr{B}}$, which means that A_a induces the same operator as FA_a. This suffices for the preservation of the fragment $\mathscr{L}_\Sigma$ by F. F preserving dependence atoms is also immediate, since their interpretation only relies on the operators on the underlying set. □

However, this changes once we start to combine sets of agents into a single agent and consider such group structures explicitly in our syntax. From a philosophical perspective, when modelling the inference and reasoning patterns of agents under certain information structure using modal logic, we not only care about individual agents themselves, but we would also like to study how *a group of agents* as a whole reasons and interacts with each other. As mentioned, this is an active topic on how to represent group agency in different contexts. Most of the traditional developments of group agency in modal logic are based on Kripkean semantics [7, 8], but there has been recent efforts exploring how to define common knowledge of a group in topological semantics [4]. Again, our categorical approach would uniformly describe the group structure in any topological category associated with every type of semantics.

To combine a group of agents to a single one, it requires us to transform an object in $\mathscr{A}^G$ for any subset $G \subseteq \Sigma$, which is a tuple representing each individual agent in the group G, to a single object in $\mathscr{A}$, which corresponds to the collective group agent. Naturally, there are two canonical ways to do this in general for any set G, using the fact that each fibre in a topological category is not only a poset, but indeed a *complete lattice*. In particular, we can form two (families of) concrete functors $\bigwedge, \bigvee : \mathscr{A}^G \to \mathscr{A}$. As the symbols suggest, for any tuple $(A_a)_{a\in G}$ in $\mathscr{A}^G$, they act on it as follows: $\bigwedge(A_a)_{a\in G} = \bigwedge_{a\in G} A_a$, and $\bigvee(A_a)_{a\in G} = \bigvee_{a\in G} A_a$. Functoriality of $\bigwedge, \bigvee$ should be immediate.

These functors then allow us to combine a group of agents of arbitrary size into a single one. We will denote them as the $\bigwedge$- and $\bigvee$-combination of group agents, and they correspond to two different readings of what a group of agents means. Intuitively, the $\bigwedge$-combination means the group *shares the information* of each individual, as if they are *physically together*. Because once we form a group $\bigwedge_{a\in G} A_a$, for any individual a in the group G we would have $\bigwedge_{a\in G} A_a \leq A_a$ in the fibre, which implies $A_a^+ \subseteq (\bigwedge_{a\in G} A_a)^+$. Just as we have discussed before, if we adopt an epistemic interpretation of modalities, this means that whatever agent a knows, so does the group, and this holds for any agent in this group. Furthermore, the meet taken in the fibre $\mathscr{A}_X$ actually shows that the group modelled by $\bigwedge_{a\in\Sigma} A_a$ is the universal one that has this property. This informally suggests that the group acts like an agent who has access to all the information owned by each individual agent in this group, exactly like the case when everyone in the group has come to a single location, and put all of their information on the table where anyone can see. In **Pre** or **Eqv**, the $\bigwedge$-combination simply take the conjunction of all the relations, and this is exact the well-known *distributive knowledge* of a group (cf. [8]). Hence, the $\bigwedge$-combination generalises distributive knowledge to all types of semantics.

On the other hand, the $\bigvee$-combination means the group *shares the uncertainties* of each individual, as if they are only *abstractly* considered as a single agent. Dual to the case before, we must have $(\bigvee_{a\in G} A_a)^+ \subseteq A_a$ for any $a \in G$. This implies that for the combined group, if it knows something then necessarily each individual in the group also knows this, and the group agent is the universal one that has this property. To better compare with the existing literature, we observe the following simple result:

Lemma 4.5. *If the semantic functor $(-)^+$ on a topological category $\mathscr{A}$ always induces monotone and idempotent operators, then $(\bigvee_{a\in G} A_a)^+ \subseteq A_{a_1}^+ \circ \cdots \circ A_{a_n}^+$ for any $a_1, \cdots, a_n \in G$.*

Proof. If follows by $(\bigvee_{a\in G} A_a)^+ \subseteq A_{a_i}^+$ for any i, and monotonicity, idempotence of these operators. □

Translating back to natural language, in the condition of Lemma 4.5, what the $\bigvee$-combined group knows is much more restrictive, in that if the group knows something, then any agent in the group also knows it, and furthermore a_i knows that a_j knows that $\cdots$ that a_k knows it. This shows that the $\bigvee$-combination is a generalisation of the *common knowledge* of a group (again, cf. [8]).

We may now formally define the syntactic extension where we also allow group formation in our logic. For any indexed set Σ, we let Σ_l, Σ_r be *synonyms* for the power set $\wp(\Sigma)$. The language $\mathscr{L}_{\Sigma_l}^D$ and $\mathscr{L}_{\Sigma_r}^D$ is nothing more but the modal languages with agent symbols in Σ_l, Σ_r, respectively, together with all the dependence atoms between these group agents. However, we write in this way because to interpret the language $\mathscr{L}_{\Sigma_l}^D$ or $\mathscr{L}_{\Sigma_r}^D$, we still only need to work within $\mathscr{A}^\Sigma$, not $\mathscr{A}^{\Sigma_l}$ or $\mathscr{A}^{\Sigma_r}$.

Given an object $(A_a)_{a\in\Sigma}$ in $\mathscr{A}^\Sigma$ over the set X, we can interpret the modal operators for a group of agents in the two fragments as either the $\bigwedge$- or $\bigvee$-combination. For any subset $G \subseteq \Sigma$, we define the interpretation of $\Box_G$ in $\mathscr{L}_{\Sigma_l}^D$ as the operator $(\bigwedge_{a\in G} A_a)^+$; and similarly for the language $\mathscr{L}_{\Sigma_r}^D$, $\Box_G$ is interpreted as the operator $(\bigvee_{a\in G} A_a)^+$. Building on what we have developed before, this suffices to interpret the two languages $\mathscr{L}_{\Sigma_l}^D$ and $\mathscr{L}_{\Sigma_r}^D$. Of course, for a singleton group $\{a\}$, its interpretation under the two fragments coincide, which still corresponds to the usual interpretation of the operator A_a^+. The upshot is that we can identify the following valid logical rules in the two fragments $\mathscr{L}_{\Sigma_l}^D$ and $\mathscr{L}_{\Sigma_r}^D$:

Proposition 4.6. *For any modal category $\mathscr{A}$, the following axioms are valid in $\mathscr{L}_{\Sigma_l}^D$ (resp. $\mathscr{L}_{\Sigma_r}^D$):*[5]

- **Inclusion**: *$K_G H$ (resp. $K_H G$), provided $H \subseteq G$;*
- **Additivity**: *$K_G H \wedge K_G P \to K_G(H \cup P)$ (resp. $K_H G \wedge K_P G \to K_{H\cup P} G$);*
- **Transitivity**: *$K_G H \wedge K_H P \to K_G P$ (resp. $K_G H \wedge K_H P \to K_G P$);*
- **Transfer**: *$K_G H \wedge \Box_H \varphi \to \Box_G \varphi$ (resp. $K_G H \wedge \Box_H \varphi \to \Box_G \varphi$).*

[5] Half of these axioms corresponding to the fragment $\mathscr{L}_{\Sigma_l}^D$ has already been identified in [3, 2] in the special case of **Top**.

Proof. We only prove the case for $\mathscr{L}^D_{\Sigma_l}$; the other case is completely dual. Let $(A_a)_{a\in\Sigma}$ be any object in $\mathscr{A}^\Sigma$ over X. Whenever we have $H \subseteq G \subseteq \Sigma$, we have $A_G = \bigwedge_{a\in G} A_a \subseteq \bigwedge_{a\in H} A_a = A_H$, which implies $A^+_H \subseteq A^+_G$. Hence, according to our definition of the interpretation of the dependence atoms, we have $[\![K_G H]\!] = X$, and this validates **Inclusion**. For any two groups H,P, by definition $A_{H\cup P} = \bigwedge_{a\in H\cup P} A_a = A_H \wedge A_P$, which implies $m_H \cup m_P \subseteq m_{H\cup P}$. Now locally, suppose for some $x \in X$ we have $x \in [\![K_G H]\!]$ and $x \in [\![K_G P]\!]$. Then for any $S \subseteq X$, $x \in m_{H\cup P}(S) \Rightarrow x \in m_H(S) \cup m_P(S)$. Either $x \in m_H(S)$ or $x \in m_P(S)$, we would have $x \in m_G(S)$, according to our assumption that $K_G H$ and $K_G P$ locally holds at x. Hence, the **Additivity** law also holds. The validity of **Transitivity** and **Transfer** axioms are evident. □

Up to this point, we have completed our generalisation of group structure to all the exemplar modal categories in a uniform way, and identified a set of valid inference rules. The remaining task is then to identify which part of the semantic structure in topological categories does the syntactic group-forming operation corresponds to. Considering our usage of the complete lattice structure of fibres, the following result should be of no surprise:

Theorem 4.7. *Let $F : \mathscr{A} \to \mathscr{B}$ be a modal functor between two modal categories, and suppose the semantic functor $(-)^+_{\mathscr{B}}$ is injective on objects. F preserves arbitrary meets (resp. joins) fibre-wise, i.e. the induced functions $F_X : \mathscr{A}_X \to \mathscr{B}_X$ on fibres is a morphism in* **InfL** *(resp.* **SupL***) for any set X, iff it preserves the interpretation of the language $\mathscr{L}^D_{\Sigma_l}$ (resp. $\mathscr{L}^D_{\Sigma_r}$) for any indexed set Σ.*

Proof. Again, we only prove the case for F preserving meets fibre-wise and the preservation of the interpretation of $\mathscr{L}^D_{\Sigma_l}$. We already know from Theorem 4.4 that F is a modal functor iff it preserves the interpretation of $\mathscr{L}^D_{\Sigma}$, thus it suffices to show it further preserves the interpretation of $\bigwedge$-group-formation iff it preserves meets fibre-wise. From how the $\bigwedge$-group modality is defined, it is immediate to note that F preserves the interpretation of $\mathscr{L}^D_{\Sigma_l}$ iff $(\bigwedge_{a\in\Sigma} A_a)^+_{\mathscr{A}}$, which by the fact of F being a modal functor is equal to $(F \bigwedge_{a\in\Sigma} A_a)^+_{\mathscr{B}}$, coincides with $(\bigwedge_{a\in\Sigma} FA_a)^+_{\mathscr{B}}$, for any $(A_a)_{a\in\Sigma}$. By assumption on $(-)^+_{\mathscr{B}}$, this holds iff $F \bigwedge_{a\in\Sigma} A_a = \bigwedge_{a\in\Sigma} FA_a$, which exactly means F preserves meets fibre-wise. □

Consider the various model transformations we have described in Example 3.5, Theorem 4.4 immediately tells us how these functors behave with respect to group knowledge. For instance, since the modal embedding **Eqv** $\hookrightarrow$ **Pre** has both a concrete left and right adjoint, it must preserve both meets and joins fibre-wise, which suggests that the two fragments $\mathscr{L}^D_{\Sigma_l}$ and $\mathscr{L}^D_{\Sigma_r}$ behave coherently between **Eqv** and **Pre**. However, as for the embedding of **Eqv** and **Pre** into **Kr**, it only has a concrete left adjoint but lacks a right one, which means only the $\bigwedge$-group formation, viz. the distributive knowledge, coincide in **Eqv**, **Pre** and **Kr**, but not the common knowledge. We can see this more explicitly, since the join in fibres of **Kr** are simply unions of relations, while in **Pre** and **Eqv** we must further take the transitive closure of unions of relations. Other modal embeddings can be analysed in a similar fashion.

5 Logical Dynamics and Fibre Connections

The "dynamic turn" of modal logic in the recent two decades makes *logical dynamics* another very important topic in the current literature. In this section, we will see how certain general types of logical dynamics could be subsumed into our categorical framework in a similar fashion as before.

Logical dynamics concerns with the reasoning patterns of agents when *new information* comes in, which generally *changes the underlying set* of a model. This is where the *fibre connection* plays a crucial role, because it allows us to transfer the geometric data over the original model to the updated model in a uniform way. For simplicity, below we describe all the dynamic extensions based on the simplest fragment $\mathscr{L}$, but it should be clear that our method can be equally applied to other fragments as well.

To warm up, we start by generalising the simplest form of dynamic logic, known as PAL, public announcement logic (cf. [20, 21]). It concerns with information events of publicly announcing that φ holds, which we denote as $!\varphi$. A typical formula in PAL is of the form $[!\varphi]\psi$, intuitively read as ψ is true after announcing φ. For a modal category $\mathscr{A}$, given any object (A,V) in the fibre $(\mathscr{A}\times\mathbf{Evl})_X$, the information event $!\varphi$ naturally restricts the domain X to the subset $S=[\![\varphi]\!]_A^V$. If we denote the inclusion function $S\hookrightarrow X$ as i, then the natural way to transfer the geometric data on X to S is by pulling back along i. This way, we obtain a new semantic model (i^*A, i^*V) over S, and the formula following the dynamic operator $[!\varphi]$ could be interpreted in this new model. We also need to transfer subsets of S back to subsets of X, to maintain the recursive structure of adding dynamic operators within the syntax. The natural candidates are $\exists_i$ and $\forall_i$, which we will see correspond to the pair of dual operators $\langle !\varphi\rangle$ and $[!\varphi]$.

More formally, we define the extension $\mathscr{L}^{\mathbf{PAL}}$ of $\mathscr{L}$ by the smallest set of formulas containing $\mathscr{L}$ and is closed under forming dynamic formulas of the form $[!\varphi]\psi$, with φ,ψ in $\mathscr{L}^{\mathbf{PAL}}$. Following the above informal idea, we define the interpretation of formulas in $\mathscr{L}^{\mathbf{PAL}}$ by adding the following recursive clause: For (A,V) in $(\mathscr{A}\times\mathbf{Evl})_X$, we define $[\![[!\varphi]\psi]\!]_A^V=\forall_i[\![\psi]\!]_{i^*A}^{i^*V}$, and $[\![\langle !\varphi\rangle\psi]\!]_A^V=\exists_i[\![\psi]\!]_{i^*A}^{i^*V}$, where i is the inclusiong map $[\![\varphi]\!]_A^V\hookrightarrow X$. Perhaps the more familiar form of truth conditions of these dynamic operators are the following equivalent local formulation: For any $x\in X$,

$$A,V,x\models[!\varphi]\psi\Leftrightarrow A,V,x\models\varphi \text{ implies } i^*A,i^*V,x\models\psi,$$
$$A,V,x\models\langle !\varphi\rangle\psi\Leftrightarrow A,V,x\models\varphi \text{ and } i^*A,i^*V,x\models\psi.$$

Again, if we combine this general form of semantics of PAL in any modal category with the special description of pullback maps in **Kr** given in the proof of Proposition 2.6, we recover exactly the usual PAL dynamics developed for Kripke models, but we also get the PAL dynamics in other types of semantics at the same time. This again exhibits the usefulness of a unifying description of semantics of modal logic.

Expectedly, the syntactic PAL dynamic operators in the $\mathscr{L}^{\mathbf{PAL}}$ fragment should correspond to the semantic structure of initial lifts along inclusions in a topological category:

Theorem 5.1. *Let $F:\mathscr{A}\to\mathscr{B}$ be a modal functor between two modal categories, and suppose the semantic functor $(-)^+_{\mathscr{B}}$ is injective on objects. F further preserves the interpretation of $\mathscr{L}^{\mathbf{PAL}}$ iff it preserves the initial lifts of any injections, i.e. for any inclusion map $i:S\hookrightarrow X$ and for any object A in the fibre $\mathscr{A}_X$, $Fi^*A=i^*FA$ holds.*

Proof. Again for the if direction we prove by induction, and the only case we need to think about is for the PAL dynamic operator. Given φ,ψ and any (A,V) in $(\mathscr{A}\times\mathbf{Evl})_X$, by induction hypothesis we have $[\![\varphi]\!]_A^V=[\![\varphi]\!]_{FA}^V$, and we denote the inclusion map of this subset into X by i. Now by definition of the interpretation of $[!\varphi]\psi$, we have $[\![[!\varphi]\psi]\!]_A^V=\forall_i[\![\psi]\!]_{i^*A}^{i^*V}=\forall_i[\![\psi]\!]_{Fi^*A}^{i^*V}=\forall_i[\![\psi]\!]_{i^*FA}^{i^*V}=[\![[!\varphi]\psi]\!]_{FA}^V$. Thus, F preserves the interpretation of $\mathscr{L}^{\mathbf{PAL}}$.

On the other hand, suppose for some object A in the fibre $\mathscr{A}_X$ and for some injection $i:S\hookrightarrow X$, we have i^*FA is *not* equal to Fi^*A. This in particular suggests that the associated operators $(i^*FA)^+_{\mathscr{B}}$ and $(Fi^*A)^+_{\mathscr{B}}$ on S are not identical, and they must disagree at some subset T of S. Then let V be an interpretation on X such that $V(p)=S$ and $V(q)=T$. Consider the interpretation of the formula $\langle !p\rangle\Box q$. On one hand, we have $[\![\langle !p\rangle\Box q]\!]_A^V=\exists_i[\![\Box q]\!]_{i^*A}^{i^*V}=\exists_i[\![\Box q]\!]_{Fi^*A}^{i^*V}=(Fi^*A)^+_{\mathscr{B}}(T)$. On the other hand, we have $[\![\langle !p\rangle\Box q]\!]_{FA}^V=\exists_i[\![\Box q]\!]_{i^*FA}^{i^*V}=(i^*FA)^+_{\mathscr{B}}(T)$. By assumption, $(Fi^*A)^+_{\mathscr{B}}(T)$ does not coincide with $(i^*FA)^+_{\mathscr{B}}(T)$, and thus F does not preserve the interpretation of $\mathscr{L}^{\mathbf{PAL}}$ by definition. □

For those model transformations that has a concrete left adjoint, they automatically commutes with all pullback maps, hence preserves the interpretation of $\mathscr{L}^{\mathbf{PAL}}$. Perhaps surprisingly, all of the modal embeddings described in Example 3.5 actually do commutes with pullbacks of *injections*, though not all of them have a concrete left adjoint, and this statement for arbitrary functions is false. As a result, $\mathscr{L}^{\mathbf{PAL}}$ is a particularly nice fragment of dynamic logic to work with.

However, PAL as dynamic logic is still too restrictive. A much more powerful dynamic mechanism is *product update* in DEL, dynamic epistemic logic [6, 18]. In product update, information events themselves form a model E, which carries additional geometric data signifying agent's uncertainly about which event actually happens, and the update is parametrised by E. Each event $e \in E$ is also equipped with a formula φ_e that specifies the precondition of that event happens. For any model over a set X, the updated model is a subset of the product space $E \times X$, consisting of those pairs (e, x) where x satisfies the precondition of e. The geometric data over the updated set takes into account the ones on both X and E.

There are already several categorical reformulation and generalisation of DEL in the literature, e.g. see [17, 12], but most of them are based on relational semantics, while our approach applies to arbitrary topological categories. We first define the notion of a *product type*, which generalises event models:

Definition 5.2 (Product Type)**.** A *product type* E for the modal category $\mathscr{A}$ is a tuple $\langle E, B, W, \{\psi_e\}_{e\in E}\rangle$, where E is a set, and B, W are objects in the fibre $\mathscr{A}_E, \mathbf{Evl}_E$. The family $\{\psi_e\}_{e\in E}$ is an E-indexed family of formulas within the language $\mathscr{L}$.

The notion of *product type update* we are going to describe, which generalises DEL, is parametrised by such a product type E. For any semantic model (A, V) in the fibre $(\mathscr{A} \times \mathbf{Evl})_X$, we write $E \otimes_V X$ as the underlying set of the updated model, which is given by the dependent sum $\sum_{e\in E} [\![\psi_e]\!]^V_A$. Intuitively, the updated model is indexed by events in E, whose fibre over e is the set of all possible words satisfying the precondition ψ_e. There are then two natural projection maps $\pi_X : E \otimes_V X \to X$ and $\pi_E : E \otimes_V X \to E$, and we define the geometric data $(\mathsf{E} \otimes_V A, W \otimes V)$ in the fibre $(\mathscr{A} \times \mathbf{Evl})_{E\otimes_V X}$ to be $\pi_X^* A \wedge \pi_E^* B$ and $\pi_X^* V \wedge \pi_E^* W$, respectively. A typical dynamic formula in product type update is of the form $[\mathsf{E}, S]\varphi$ or $\langle \mathsf{E}, S\rangle \varphi$, where E is a product type and S is a subset of E. We define their interpretation as follows,

$$[\![[\mathsf{E}, S]\Phi]\!]^V_A := \forall_{\pi_X} \Big((S \otimes_V X) \to [\![\Phi]\!]^{W\otimes V}_{E\otimes_V A} \Big), \quad [\![\langle \mathsf{E}, S\rangle \Phi]\!]^V_A := \exists_{\pi_X} \Big((S \otimes_V X) \cap [\![\Phi]\!]^{W\otimes V}_{E\otimes_V A} \Big),$$

where the set $S \otimes_V X = \sum_{e\in S} [\![\psi_e]\!]^V_A$ is a subset of $E \otimes_V X$, and $\to$, $\cap$ are calculated in the power set $\wp(E \otimes_V X)$. Again, interpreted our general construction back in the relational context $\mathbf{Kr}$ of Kripke models, one immediately recovers the usual product update in DEL.[6]

In a word, the way we associate the geometric data on the updated model $E \otimes_V X$ is by pulling back the ones over X and E along the two projection maps, and then take their intersection in the fibre. However, a categorically minded reader would perhaps wonder what happens to the degenerate case where we have an *empty intersection*. Though being kind of trivial, this is in fact important for correspondence results of product type update, which will be stated later. Hence, we also introduce *empty product update*, whose syntactic structure is extremely simple: It is of the form $\mathsf{U}\varphi$, and for any (A, V) in $(\mathscr{A} \times \mathbf{Evl})_X$ we define $[\![\mathsf{U}\varphi]\!]^V_A$ to be $[\![\varphi]\!]^V_{\top_X}$, where $\top_X$ is the maximal element in $\mathscr{A}_X$. This is indeed a form of dynamics, since the operators U results in the change of the geometric data, though the update is constant in all cases. We then define $\mathscr{L}^{\mathbf{PRO}}$ to be the least fragment containing $\mathscr{L}$, which is also closed under taking dynamic formulas of empty product update and product type update.

[6] In the literature, only the case where S is a singleton set $\{e\}$ is usually considered, but this is a minor generalisation. It is possible to define product update more generally along any function mapping into E, but we leave that for future work.

Now that product type update is properly generalised to arbitrary topological categories, we can realise PAL dynamics as special case of product type update. In fact, for any formula φ, we can associate it with a product type, which we also denote as $!\varphi$. Explicitly, $!\varphi$ is the tuple $\langle 1, \top, \top, \{\varphi\}\rangle$, where 1 is the singleton set, and φ is the corresponding precondition of the single element in 1. It is evident that the updated model by this product type $!\varphi$ is exactly the one obtained by publicly announcing φ in PAL dynamics. In fact, many other types of dynamics turn out to be special cases (cf. [6]).

Now, it should certainly be expected that the dynamic extension $\mathscr{L}^{\mathbf{PRO}}$ corresponds exactly to pullback maps between fibres and finite meets within fibres. However, for product type update, we need to slightly modify our definition of preservation of languages, since now in the syntax of $\mathscr{L}^{\mathbf{PRO}}$, we have explicitly included certain semantic data, viz. the product types E. We now say a concrete functor $F : \mathscr{A} \to \mathscr{B}$ preserves the interpretation of $\mathscr{L}^{\mathbf{PRO}}$ if, after uniformly changing every product type $\mathsf{E} = \langle E, B, W, \{\psi_e\}_{e\in E}\rangle$ appearing in the syntax to $F\mathsf{E} = \langle E, FB, W, \{\psi_e\}_{e\in E}\rangle$, the resulting interpretation remains unchanged under transformation of models induced by F.[7] We then have the following result:

Theorem 5.3. *Let $F : \mathscr{A} \to \mathscr{B}$ be a modal functor between two modal categories, and suppose the semantic functor $(-)^+_{\mathscr{B}}$ is injective on objects. F further preserves the interpretation of $\mathscr{L}^{\mathbf{PRO}}$ iff it preserves pullback maps and fibre-wise finite meets.*

Proof. The if part can be proven by a straight forward induction on the complexity of formulas in $\mathscr{L}^{\mathbf{PRO}}$. The only if part is technically trickier, though the general idea is no different from previous proofs of such correspondence results. We include a detailed proof in Appendix A for the convenience of referees. □

6 Conclusion

In this paper, we have used the language of topological categories to provide a unifying description of different types of semantics of modal logic, and have showed how various semantic structures within topological categories enable us to interpret different extensions of modal logic, including modal strength, group structure, and logical dynamics. We believe our approach is instructive for the current active research in the modal logic world on related topics.

For each fragment we have also proven a correspondence result, showing the equivalence for a concrete functor to preserve the interpretation of that fragment and for it to preserve certain categorical structures. Such results have established a close connection between the syntax and semantics of modal logic, and have deepened our understanding of its abstract mathematical structures. They can be seen as justification that topological category is a particularly nice framework to explore its further connections with modal logic.

Acknowledgement

In the process of preparing this paper, we are in great debt to many extremely useful suggestion and constructive comments provided by Johan van Benthem and Levin Hornischer. We would also like to thank the annonymous referees for the helpful advice on the content and the presentation of this paper.

[7] A far more general approach is to look at the relationship between a model transformation induced by a concrete functor F, and a particular syntactic translation T. Our notion of preservation of languages is then a special case when T is the identity translation, or in the case of product type updates, translating E to $F\mathsf{E}$. We leave this for future investigation.

References

[1] Jiří Adámek, Horst Herrlich & George E. Strecker (1990): *Abstract and concrete categories : the joy of cats*. Wiley, New York.

[2] Alexandru Baltag & Johan van Benthem (2021): *Knowability and Continuity: a topological account of epistemic dependence*. Unpublished draft.

[3] Alexandru Baltag & Johan van Benthem (2021): *A Simple Logic of Functional Dependence*. Journal of Philosophical Logic 50(5), pp. 939–1005, doi:10.1007/s10992-020-09588-z.

[4] Alexandru Baltag, Nick Bezhanishvili, Aybüke Özgün & Sonja Smets (2019): *A Topological Approach to Full Belief*. Journal of Philosophical Logic 48(2), pp. 205–244, doi:10.1007/s10992-018-9463-4.

[5] J. van Benthem & E. Pacuit (2011): *Dynamic Logics of Evidence-Based Beliefs*. Studia Logica 99(1), p. 61, doi:10.1007/s11225-011-9347-x.

[6] Johan van Benthem (2007): *Dynamic logic for belief revision*. Journal of Applied Non-Classical Logics 17(2), pp. 129–155, doi:10.3166/jancl.17.129-155.

[7] Johan van Benthem (2010): *Modal Logic for Open Minds*. Stanford: Center for the Study of Language and Information.

[8] Johan van Benthem (2011): *Logical dynamics of information and interaction*. Cambridge University Press, doi:10.1017/CBO9780511974533.

[9] Johan van Benthem, David Fernández-Duque & Eric Pacuit (2012): *Ecidence Logic: A New Look at Neighborhood Structures*. In Marcus Kracht, Maarten de Rijke, Heinrich Wansing & Michael Zakharyaschev, editors: Advances in Modal Logic, CSLI Publications, pp. 97–118.

[10] Patrick Blackburn, Johan FAK van Benthem & Frank Wolter (2006): *Handbook of modal logic*. Elsevier.

[11] Patrick Blackburn, Maarten de Rijke & Yde Venema (2001): *Modal Logic*. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, doi:10.1017/CBO9781107050884.

[12] G. Cinà (2017): *Categories for the working modal logician*. ILLC Dissertation.

[13] Lou Goble & John-Jules Ch Meyer (2006): *Deontic Logic and Artificial Normative Systems: 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006, Proceedings*. 4048, Springer Berlin, Heidelberg, doi:10.1007/11786849.

[14] Jaakko Hintikka (1962): *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Ithaca: Cornell University Press.

[15] Dirk Hofmann, Gavin J Seal & Walter Tholen (2014): *Monoidal Topology: A Categorical Approach to Order, Metric, and Topology*. 153, Cambridge University Press, doi:10.1017/CBO9781107517288.

[16] André Joyal & Myles Tierney (1984): *An extension of the Galois theory of Grothendieck*. 309, American Mathematical Soc.

[17] Kohei Kishida (2017): *Categories for Dynamic Epistemic Logic*. Electronic Proceedings in Theoretical Computer Science 251, pp. 353–372, doi:10.4204/EPTCS.251.26.

[18] Fenrong Liu (2011): *Reasoning about preference dynamics*. 354, Springer Science & Business Media, doi:10.1007/978-94-007-1344-4.

[19] John Charles Chenoweth McKinsey & Alfred Tarski (1944): *The algebra of topology*. Annals of mathematics, pp. 141–191, doi:10.2307/1969080.

[20] Jan Plaza (1989): *Logics of public announcements*. In: Proceedings 4th International Symposium on Methodologies for Intelligent Systems, pp. 201–216.

[21] Jan Plaza (2007): *Logics of public communications*. Synthese 158(2), pp. 165–179, doi:10.1007/s11229-007-9168-7.

[22] Dana Scott (1970): *Advice on modal logic*. In: Philosophical problems in logic, Springer, pp. 143–173, doi:10.1007/978-94-010-3272-8_7.

[23] Allard Tamminga, Hein Duijf & Frederik Van De Putte (2021): *Expressivity results for deontic logics of collective agency*. Synthese 198(9), pp. 8733–8753, doi:10.1007/s11229-020-02597-0.

[24] Georg H. von Wright (1954): *An essay in modal logic*. British Journal for the Philosophy of Science 5(18).

A Proof of Theorem 5.3

To complete the other half of the proof, we roughly need to show that any initial lift and any finite meets in the fibre could be represented by some product type update with a specific chosen product type. First of all, since empty product type update is included in our dynamic extension $\mathscr{L}^{\mathbf{PRO}}$, to preserve it we may assume F already preserves the top element within each fibre. We first show that F commutes with pullback maps. Suppose for some function $\pi : E \to X$, F does not commute with the initial lifts on π in $\mathscr{A}$ and $\mathscr{B}$. This means that we have some object A in the fibre $\mathscr{A}_X$, such that $F\pi^*A$ and π^*FA are two distinct objects in $\mathscr{B}_X$. Now since the semantic functor on $\mathscr{B}$ is injective on objects, the induced operators $(\pi^*FA)^+_{\mathscr{B}}$, which we denote as m, and $(F\pi^*A)^+_{\mathscr{B}}$, which we denote as m', will be distinct, which means they disagree on some subset T of E.

Now consider the product type $\mathsf{E} = \langle E, \top^{\mathscr{A}}_E, W, \{p_e\}_{e\in E}\rangle$, where the family of formulas is an E-indexed family of distinct propositional letters. For the evaluation function W on E, we require that for some propositional letter q distinct from p_e for any $e \in E$, we have $W(q) = T$. Now consider an evaluation function V on X, such that for any $e \in E$ we have $V(p_e) = \{\pi(e)\}$, which means that $[\![p_e]\!]^V_A$ is a singleton for any $e \in E$. We also requires that $V(q) = X$. Then by definition, we have

$$E \otimes_V X = \sum_{e\in E} [\![p_e]\!]^V_A = E,$$

and it is not hard to see that the projection map π_E is the identity on E, and π_X is simply given by π. Notice that, the above statement of the underlying set of the updated model remains true even if we have calculated it in $\mathscr{B}$.

Now by definition, the geometric data on the updated model is calculated as follows,

$$\mathsf{E} \otimes_V A = 1^*_E \top^{\mathscr{A}}_E \wedge \pi^*A = \pi^*A,$$

and for the induced product update in $\mathscr{B}$,

$$F\mathsf{E} \otimes_V FA = 1^*_E F\top^{\mathscr{A}}_E \wedge \pi^*FA = \top^{\mathscr{B}}_E \wedge \pi^*FA = \pi^*FA.$$

The above uses the fact that initial lifts preserves top elements since it is a right adjoint, and the assumption that F preserves top elements in the fibres. As for the evaluation function $W \otimes V$, it is easy to calculate that

$$(W \otimes V)(q) = W(q) \wedge \pi^{-1}V(q) = W(q) = T.$$

Finally, consider the interpretation of the formula $\langle \mathsf{E}, \{e\}\rangle \Box q$, where e is some element in E such that $e \in m(T)$ but $e \notin m'(T)$ (or the other way around). Then by definition, we have the following calculation,

$$[\![\langle \mathsf{E}, \{e\}\rangle \Box q]\!]^V_A = \exists_\pi(\{(e,\pi(e))\} \cap [\![\Box q]\!]^{W\otimes V}_{\pi^*A}) = \exists_\pi(\{(e,\pi(e))\} \cap [\![\Box q]\!]^{W\otimes V}_{F\pi^*A}) = \emptyset.$$

The first equality is due to the fact that $E \otimes_V A = \pi^*A$ as we have shown above; the second equality is by the fact that F preserves the interpretation of $\mathscr{L}$; and the final equality holds because we have assumed $e \notin m'(T)$. On the other hand, we have the other calculation as follows,

$$[\![\langle F\mathsf{E}, e\rangle \Box q]\!]^V_{FA} = \exists_\pi(\{(e,\pi(e))\} \cap [\![\Box q]\!]^{W\otimes V}_{F\mathsf{E}\otimes_V FA}) = \exists_\pi(\{(e,\pi(e))\} \cap [\![\Box q]\!]^{W\otimes V}_{\pi^*FA}) = \{\pi(e)\}.$$

These calculation are basically the same as before, only that in the final step, the result is a singleton $\{\pi(e)\}$ because $e \in m(T)$. This constructions shows that F would then not preserve the interpretation of the formula $\langle \mathsf{E}, \{e\}\rangle \Box q$ on this particular model. Hence, F must preserves the initial lift of any single structured sources.

Furthermore, we need to show that F preserves the binary meets fibre-wise as well. The basic idea is the same. Suppose F does not preserve binary meets in the fibre, then for some set X and some A, B in the fibre $\mathscr{A}_X$, we would have $F(A \wedge B)$ distinct from $FA \wedge FB$ in $\mathscr{B}_X$. Again, the operators m, m' associated to $F(A \wedge B)$ and $FA \wedge FB$ would differ on some subset T of X; we let $y \in X$ be the element in $m(T)$ but not $m'(T)$ (or the other way around). We can then construct the product type X as follows,

$$\mathsf{X} = \langle X, B, \top_X^{\mathbf{Evl}}, \{p_x\}_{x \in X} \rangle.$$

We also consider the model A on X, with a chosen evaluation function V satisfying the following condition: For any $x \in X$, we have $V(p_x) = \{x\}$, and for another distinct variable q we have $V(q) = T$. The product type update would result in the following model,

$$X \otimes_V X = \sum_{x \in X} [\![p_x]\!]_A^V = X,$$

and the two projection maps are both the identity function 1_X on X. Again, this is independent of the modal categories $\mathscr{A}$ or $\mathscr{B}$. The topology categorical structure on the updated model, calculated in $\mathscr{A}$, is simply given as follows,

$$\mathsf{X} \otimes_V A = 1_X^* B \wedge 1_X^* A = A \wedge B.$$

In the modal category $\mathscr{B}$ however, we have

$$F\mathsf{X} \otimes_V FA = 1^* XFB \wedge 1^* XFA = FA \wedge FB.$$

In both cases, it is easy to see that the updated evaluation function $\top_X^{\mathbf{Evl}} \otimes V$ remains to be V itself.

By definition, consider the evaluation of the formula $\langle \mathsf{X}, \{y\}\rangle \Box q$. On one hand,

$$[\![\langle \mathsf{X}, \{y\}\rangle \Box q]\!]_A^V = \{y\} \cap [\![\Box q]\!]_{A \wedge B}^V = \{y\} \cap [\![\Box q]\!]_{F(A \wedge B)}^V = \{y\} \cap m(T) = \{y\}.$$

On the other hand,

$$[\![\langle F\mathsf{X}, \{y\}\rangle \Box q]\!]_{FA}^V = \{y\} \cap [\![\Box q]\!]_{FA \wedge FB}^V = \{y\} \cap m'(T) = \emptyset.$$

Hence, this explicitly constructs a formula where F does not preserve its interpretation, and this completes the proof.

Universal Properties of Lens Proxy Pullbacks

Matthew Di Meglio
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
Edinburgh, Scotland
m.dimeglio@ed.ac.uk

A comprehensive account of the categorical properties of the category of small categories and asymmetric delta lenses is given in the recent works of Chollet et al. and Di Meglio. An important construction for proving many of these properties is Johnson and Rosebrugh's *"pullback"* of lenses, which we call the *proxy pullback* of lenses. We give a new treatment of the proxy pullback in terms of *compatibility*—a stronger notion of commutativity for squares of lenses. The proxy pullback is sometimes, but not always, a real pullback. Using new notions of *sync-minimal* and *independent* lens spans, we characterise when a lens span that forms a commuting square with a lens cospan has a comparison lens to a proxy pullback of the cospan.

1 Introduction

A *bidirectional transformation* is a specification of when the joint state of two systems should be regarded as consistent, together with a protocol for updating each system to restore consistency in response to a change in the other [13]. An *asymmetric* bidirectional transformation is one where the state of one of the systems, called the *view*, is completely determined by that of the other, called the *source*.

A *symmetric delta lens* is a mathematical model of a bidirectional transformation in which both of the systems involved are modelled as categories of states and transitions (deltas) rather than merely as sets of states, and the consistency restoration operations are aware of specifically which transition occurred rather than merely the state resulting from it [12]. An *asymmetric delta lens* is, in a similar way, a mathematical model of an asymmetric bidirectional transformation [11]. Johnson and Rosebrugh established an equivalence between the symmetric delta lenses between two categories and the spans of asymmetric delta lenses between the categories modulo a certain equivalence relation [14]. This correspondence is important because, although we usually need the level of generality afforded by symmetric delta lenses to describe real-world bidirectional transformations, we would rather work with asymmetric delta lenses as they are easier to reason about.

Bidirectional transformations can be chained together; this is modelled mathematically by composition of symmetric delta lenses. Under the equivalence described above, composition of spans of asymmetric delta lenses is achieved not by pullbacks, which do not even always exist [10], but by a seemingly ad hoc pullback-like construction that Johnson and Rosebrugh called the *"pullback"* (with quotation marks) [14]. As in prior work [9], we adopt the name *proxy pullback* from Bumpus and Kocsis [4].

Our goal in this work is to understand in what sense the notion of lens proxy pullback is actually canonical. In category theory, canonicity is usually formalised by a universal property. Unfortunately, the obvious universal property, the one that characterises real pullbacks, does not always hold for lens proxy pullbacks. However, $\mathscr{L}ens$ is not the only category with proxies for real pullbacks. Others include

- the category of Polish probability spaces and measure preserving maps [18],
- the category of smooth manifolds and smooth maps [20], and

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 400–416, doi:10.4204/EPTCS.380.23

- the category of comonoids in a symmetric monoidal category with nice coreflexive equalisers [5].

Several frameworks for understanding such pullback-like constructions have been proposed, including Simpson's local independent products [18], Böhm's relative pullbacks [5] and Yassine's F-pullbacks [20]. In particular, Simpson and Böhm's approaches are both based on the idea that although a proxy pullback of a cospan may not be universal amongst *all* spans that form a commuting square with the cospan, it should be universal amongst *some class* of those spans. Inspired by this idea, we answer the main question

"Amongst which lens spans is a lens proxy pullback universal?"

Although the notion of lens proxy pullback is fundamentally about modelling composites of bidirectional transformations, it has recently also become an important tool for understanding the theory of lenses. In the comprehensive account of the categorical properties of the category $\mathscr{L}ens$ of small categories and lenses given by Chollet et al. [6] and Di Meglio [9], the proxy pullback played the role of a real pullback in many of the proofs. With an answer to our main question, we address some of the open questions posed by Chollet et al. [7] about real pullbacks in $\mathscr{L}ens$ and their relationship with proxy pullbacks.

Outline

In Section 3, we reformulate the notion of proxy pullback in terms of *compatibility*—a commutativity-like property of squares of (asymmetric delta) lenses. Our new definition is better suited to category theory than the original one given by Johnson and Rosebrugh.

To answer our main question, our overall approach is to search for lens span properties that are possessed by proxy-pullback spans and that are preserved by precomposition with lenses, until we find enough of them that a lens span posessing all of these conditions is guaranteed to have a comparison lens to the proxy pullback. Only two such properties are needed: the first is compatibility with the cospan; the second is a new property of lens spans that we call *independence*, which is itself defined in terms of another new property of lens spans that we call *sync minimality*. We introduce these properties in Section 4, and we prove their necessity in Section 5. In Theorem 6.1, one of the main results of this paper, we see that for sync-minimal proxy-pullback spans, the possession of these two properties by a lens span is also sufficient for the existence of such a comparison lens to the proxy-pullback span.

A natural next step is then to determine whether the sync minimality of a proxy-pullback span is itself also a necessary condition for the existence of such a comparison lens. This is not in general true, however, in Theorem 6.2, we see that it is actually a necessary condition for the *simultaneous* existence of a comparison lens to the proxy pullback from all independent lens spans that are compatible with the cospan. Stated differently, if a proxy-pullback span of a lens cospan is terminal amongst the independent spans that are compatible with the cospan, then the proxy-pullback span is necessarily sync minimal.

Combining the above results, a proxy pullback is a real pullback if and only if it is sync minimal and every lens span that forms a commuting square with the cospan is compatible with the cospan and is also independent. Although this statement completely characterises when a proxy pullback is a real pullback, it is somewhat unsatisfactory, as it is not expressed in terms of properties of the cospan that are easily checked. There is, however, a tractable characterisation for lens cospans whose apex is the terminal category. Indeed, a proxy product of two categories is a real product if and only if at least one of the two categories is a discrete category. This result, Proposition 8.3, is the pinnacle of Sections 7 and 8. Finding such a tractable characterisation for general lens cospans is ongoing work.

Remark 1.1. This paper is based on Chapter 4 of the author's Master of Research thesis [10].

2 Background

2.1 Notation

Application of functions (functors, etc.) is written by juxtaposing the function name with its argument, and parentheses are only used when needed. Binary operators like $\circ$ have lower precedence than application, so an expression like $Fa \circ Fb$ parses as $(Fa) \circ (Fb)$.

Let $\mathscr{C}at$ denote the category whose objects are small categories and whose morphisms are functors. Categories with boldface names $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, etc. are always small. We write $|\mathbf{C}|$ for the set of objects of a small category $\mathbf{C}$, and, for all $X,Y \in |\mathbf{C}|$, we write $\mathbf{C}(X,Y)$ for the set of morphisms of $\mathbf{C}$ from X to Y. For each $X \in |\mathbf{C}|$, we write $\mathbf{C}(X,*)$ for the set $\bigsqcup_{Y\in|\mathbf{C}|}\mathbf{C}(X,Y)$ of all morphisms in $\mathbf{C}$ out of X. We write $\operatorname{src} f$ and $\operatorname{tgt} f$ for, respectively, the source and target of a morphism f. We also write $f\colon X \to Y$ to say that $X,Y \in |\mathbf{C}|$ and $f \in \mathbf{C}(X,Y)$. The composite of morphisms $f\colon X \to Y$ and $g\colon Y \to Z$ is denoted $g \circ f$.

The category with a single object 0 and no non-identity morphisms, also known as the *terminal category*, is denoted $\mathbf{1}$. The category with two objects 0 and 1 and a single non-identity morphism, namely $u\colon 0 \to 1$, also known as the *interval category*, is denoted $\mathbf{2}$. We will identify objects and morphisms of a small category $\mathbf{C}$ with the corresponding functors $\mathbf{1} \to \mathbf{C}$ and $\mathbf{2} \to \mathbf{C}$ respectively.

If the square in $\mathscr{C}at$

$$\begin{array}{ccc} \mathbf{D} & \xrightarrow{T} & \mathbf{B} \\ {\scriptstyle S}\downarrow & & \downarrow{\scriptstyle G} \\ \mathbf{A} & \xrightarrow[F]{} & \mathbf{C} \end{array}$$

is a pullback square and $S'\colon \mathbf{D}' \to \mathbf{A}$ and $T'\colon \mathbf{D}' \to \mathbf{B}$ are functors for which $F \circ S' = G \circ T'$, then we write $\langle S', T'\rangle$ for the functor $\mathbf{D}' \to \mathbf{D}$ induced from S' and T' by the universal property of the pullback. By our above identification of objects with functors from $\mathbf{1}$, if $A \in |\mathbf{A}|$ and $B \in |\mathbf{B}|$ are such that $FA = GB$, then $\langle A, B\rangle$ is the object of $\mathbf{D}$ selected by the functor $\mathbf{1} \to \mathbf{D}$ induced by the universal property of the pullback from the functors $\mathbf{1} \to \mathbf{A}$ and $\mathbf{1} \to \mathbf{B}$ that respectively select the objects A and B.

2.2 Cofunctors and Lenses

The definition of (asymmetric delta) lens most useful to us will be as a suitable pairing of a functor and a cofunctor [2]. Let us first recall the definition of a cofunctor [1, 7].

Definition 2.1. For small categories $\mathbf{A}$ and $\mathbf{B}$, a *cofunctor* $F\colon \mathbf{A} \to \mathbf{B}$ consists of

- a function $F\colon |\mathbf{A}| \to |\mathbf{B}|$, called the *object function*, and
- functions $F^A\colon \mathbf{B}(FA,*) \to \mathbf{A}(A,*)$ for all $A \in |\mathbf{A}|$, called *lifting functions*,

such that the equations

$$F \operatorname{tgt} F^A b = \operatorname{tgt} b \quad \text{(PutTgt)} \qquad F^A \operatorname{id}_{FA} = \operatorname{id}_A \quad \text{(PutId)} \qquad F^A(b' \circ b) = F^{A'} b' \circ F^A b \quad \text{(PutPut)}$$

hold whenever they are defined.

Warning 2.2. The notions of cofunctor and contravariant functor are distinct and unrelated.

There is a category $\mathscr{C}of$ whose objects are small categories and whose morphisms are cofunctors. The composite $G \circ F$ of cofunctors $F\colon \mathbf{A} \to \mathbf{B}$ and $G\colon \mathbf{B} \to \mathbf{C}$ has as its object function the composite of the object functions of F and G, and has $(G \circ F)^A c = F^A G^{FA} c$ for all $A \in |\mathbf{A}|$ and all $c \in \mathbf{C}(GFA,*)$.

In the following definition of a lens, although we use the name of the lens to refer both to its get functor and its put cofunctor, the equal object function requirement ensures that there is no ambiguity.

Definition 2.3. For small categories **A** and **B**, a *lens* $F\colon \mathbf{A} \to \mathbf{B}$ consists of

- a functor $F\colon \mathbf{A} \to \mathbf{B}$, called the *get functor*, and
- a cofunctor $F\colon \mathbf{A} \to \mathbf{B}$, called the *put cofunctor*,

with same object functions, such that the equation

$$FF^A b = b \tag{PutGet}$$

holds whenever it is defined.

There is a category $\mathscr{L}ens$ of small categories and lenses. There are also identity-on-objects functors $\mathscr{G}\colon \mathscr{L}ens \to \mathscr{C}at$ and $\mathscr{P}\colon \mathscr{L}ens \to \mathscr{C}of$ that respectively send a lens to its get functor and put cofunctor.

2.3 Discrete Opfibrations and Split Opfibrations

Definition 2.4. A functor $F\colon \mathbf{A} \to \mathbf{B}$ is a *discrete opfibration* if, for each $A \in |\mathbf{A}|$ and each $b \in \mathbf{B}(FA,*)$, there is a unique $a \in \mathbf{A}(A,*)$ such that $Fa = b$.

Definition 2.5. A lens $F\colon \mathbf{A} \to \mathbf{B}$ is a *discrete opfibration* if the equation

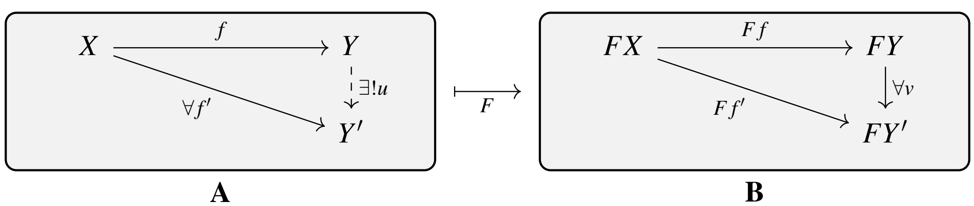
$$F^A Fa = a \tag{GetPut}$$

holds for each $A \in |\mathbf{A}|$ and each $a \in \mathbf{A}(A,*)$.

Warning 2.6. The name GetPut has, in the past, been used for what is now called PutId. The reader should note that lenses in general need not satisfy GetPut the way that we have defined it.

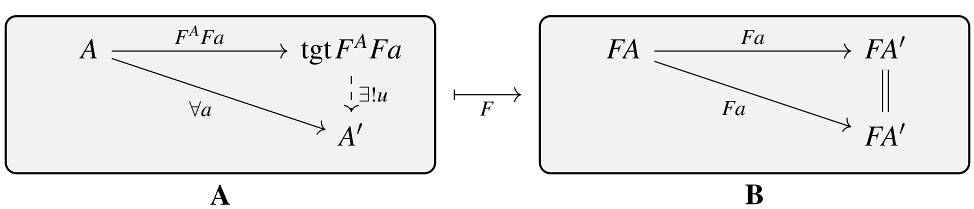
If $F\colon \mathbf{A} \to \mathbf{B}$ is a discrete opfibration, then there is a unique lens mapped by $\mathscr{G}$ to F, which we sometimes also refer to as F. A lens is a discrete opfibration if and only if its get functor is a discrete opfibration. Together, these results mean that we need not specify whether a discrete opfibration $F\colon \mathbf{A} \to \mathbf{B}$ is a functor or a lens, and we can use the name F in both functor and lens contexts without ambiguity.

Definition 2.7. For a functor $F\colon \mathbf{A} \to \mathbf{B}$, a morphism $f\colon X \to Y$ in **A** is *F-opcartesian* if, for all morphisms $f'\colon X \to Y'$ in **A** and all morphisms $v\colon FY \to FY'$ in **B** such that $Ff' = v \circ Ff$, there is a unique morphism $u\colon Y \to Y'$ in **A** such that $f' = u \circ f$ and $v = Fu$. For f to be *weakly F-opcartesian*, the property described in the previous sentence need only hold for $v = \mathrm{id}_{FY}$.



Definition 2.8. A lens $F\colon \mathbf{A} \to \mathbf{B}$ is a *split opfibration* if each morphism $F^A b$ is $\mathscr{G}F$-opcartesian.

Proposition 2.9. *A lens $F\colon \mathbf{A} \to \mathbf{B}$ is a split opfibration if and only if, for all $a\colon A \to A'$ in **A**, there is a unique $u\colon \operatorname{tgt} F^A Fa \to A'$ in **A** such that $a = u \circ F^A Fa$ and $Fu = \mathrm{id}_{FA'}$.*



Proof sketch. Having opcartesian lifts is equivalent to having weakly opcartesian lifts that are closed under composition. The chosen lifts of a *lens* are, by the PutPut axiom, closed under composition. □

In particular, every discrete opfibration is a split opfibration.

3 Compatible Squares and Proxy Pullbacks

Compatibility is a stronger notion of commutativity for a square of lenses. In addition to requiring that the underlying square of functors and the underlying square of cofunctors commute, it also imposes conditions on certain squares formed from a mix of the underlying functors and the underlying cofunctors.

Definition 3.1. A *compatible lens square* is a commuting lens square

$$\begin{array}{ccc} \mathbf{D} & \xrightarrow{\overline{F}} & \mathbf{B} \\ {\scriptstyle \overline{G}}\downarrow & & \downarrow{\scriptstyle G} \\ \mathbf{A} & \xrightarrow[F]{} & \mathbf{C} \end{array} \tag{1}$$

such that the compatibility equations

$$\overline{F}\,\overline{G}^D a = G^{\overline{F}D} Fa \qquad\qquad \overline{G}\,\overline{F}^D b = F^{\overline{G}D} Gb$$

hold whenever they are defined. We also say that $(\overline{G},\overline{F})$ is *compatible with* (F,G).

Proposition 3.2. *Every commuting lens square for which one leg of the cospan is a discrete opfibration is a compatible lens square.*

Proof. For all $D \in |\mathbf{D}|$, all $a \in \mathbf{A}(\overline{G}D,*)$ and all $b \in \mathbf{B}(\overline{F}D,*)$, we have

$$\overline{F}\,\overline{G}^D a = \overline{F}\,\overline{G}^D F^{\overline{G}D} Fa = \overline{F}(\overline{G}\circ F)^D Fa = \overline{F}(\overline{F}\circ G)^D Fa = \overline{F}\,\overline{F}^D \overline{G}^{\overline{F}D} Fa = \overline{G}^{\overline{F}D} Fa,$$
$$\overline{G}\,\overline{F}^D b = F^{\overline{G}D} F\overline{G}\,\overline{F}^D b = F^{\overline{G}D} G\overline{F}\,\overline{F}^D b = F^{\overline{G}D} Gb. \qquad \square$$

Remark 3.3. As identity lenses are discrete opfibrations, every commuting lens triangle becomes a compatible lens square by inserting an identity lens into the triangle in the appropriate place.

Definition 3.4. A *proxy-pullback square* is a compatible lens square sent by $\mathscr{G}$ to a pullback square. A *proxy pullback* of a lens cospan is a lens span forming a proxy-pullback square with the cospan. A *proxy product* is a proxy pullback of a cospan whose apex is the terminal category.

In diagrams, we will mark proxy-pullback squares with PPB.

Suppose that the lens square (1) is mapped by $\mathscr{G}$ to a pullback square. By the universal property of this pullback square, the compatibility conditions for (1) to be a proxy-pullback square are equivalent to the equations $\overline{G}^D a = \langle a, G^{\overline{F}D} Fa\rangle$ and $\overline{F}^D b = \langle F^{\overline{G}D} Gb, b\rangle$. Actually, starting with a pullback in $\mathscr{C}at$ of the get functors of a lens span, these equations define lifts on the pullback projection functors; one may check that this turns these functors into lenses and that the resulting lens square is compatible.

Proposition 3.5. *For each lens cospan, there is a unique proxy pullback of the cospan above each pullback of the get functors of the cospan.*

Proposition 3.6 ([10, Corollary 3.15])**.** *Proxy-pullback spans are unique up to unique span isomorphism.*

4 Sync-minimal and Independent Lens Spans

As was explained in the introduction to this paper, new notions of *sync minimality* and *independence* of lens spans give various necessary and sufficient conditions for the existence of comparison lenses into proxy pullbacks. In this section, we merely introduce these notions, delaying the development of their theory to when it is needed later in the paper.

Johnson and Rosebrugh [14] proposed that we regard a lens span $\mathbf{A} \xleftarrow{F} \mathbf{C} \xrightarrow{G} \mathbf{B}$ as a synchronisation protocol between the systems represented by the categories $\mathbf{A}$ and $\mathbf{B}$. From this perspective, the category $\mathbf{C}$ has the sole purpose of coordinating the propagation to $\mathbf{B}$ of transitions that occur in $\mathbf{A}$ and vice versa. As transitions always originate in $\mathbf{A}$ or $\mathbf{B}$, there may be morphisms in $\mathbf{C}$ that are never used—these are the ones that are not composites of a sequence of morphisms that are all lifts along F or G. If there are no such extraneous morphisms in $\mathbf{C}$, we call the lens span *sync minimal*.

Definition 4.1. A lens span

$$\mathbf{A} \xleftarrow{F} \mathbf{C} \xrightarrow{G} \mathbf{B}$$

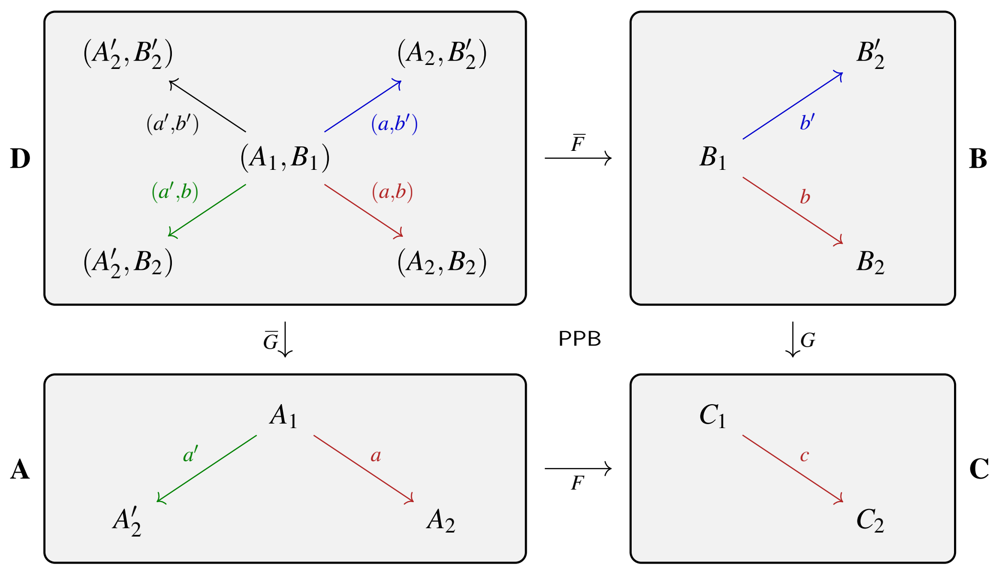
is *sync minimal* if each morphism in $\mathbf{C}$ is a composite of a sequence of morphisms

$$C_1 \xrightarrow{c_1} C_2 \xrightarrow{c_2} C_3 \cdots C_{n-1} \xrightarrow{c_{n-1}} C_n$$

that are all lifts along F or G, that is, for each k, either $c_k = F^{C_k} F c_k$ or $c_k = G^{C_k} G c_k$.

There are many sync-minimal lens spans, but not all proxy-pullback spans are sync minimal.

Example 4.2. Consider the proxy-pullback square depicted in the diagram below, where the lens lifts are indicated by the colouring of the morphisms.



The lens span $(\overline{G}, \overline{F})$ is not sync minimal as the morphism (a', b') is not a composite of lifts. Notice that removing (a', b') from $\mathbf{D}$ would make the span $(\overline{G}, \overline{F})$ sync minimal.

Starting with a lens span $\mathbf{A} \xleftarrow{F} \mathbf{C} \xrightarrow{G} \mathbf{B}$, by removing all morphisms in $\mathbf{C}$ that are not composites of a sequence of morphisms that are lifts along F or G, we obtain a sync-minimal lens span from $\mathbf{A}$ to $\mathbf{B}$ that encodes the same synchronisation protocol as (F, G). We call this sync-minimal lens span the *sync-minimal core* of (F, G) and denote it by $\mathscr{M}(F, G)$. Let $E_{(F,G)}$ denote the inclusion functor from the apex of $\mathscr{M}(F, G)$ to $\mathbf{C}$.

We are now ready to define the notion of *independence* for lens spans. It is similar to a jointly-monic condition, except only with respect to morphisms in the apex of the sync-minimal core of the span with the same source object. Defining independence with respect to the sync-minimal core is necessary for independence to be preserved by precomposition with lenses.

Definition 4.3. A lens span $\mathbf{A} \xleftarrow{F} \mathbf{C} \xrightarrow{G} \mathbf{B}$ is called *independent* if, for all morphisms c and c' in the apex of $\mathscr{M}(F,G)$ with the same source, whenever $Fc = Fc'$ and $Gc = Gc'$, also $c = c'$.

Remark 4.4. Simpson [18] defines the notion of *independent product* with respect to a chosen *independence structure*—a multicategory of multispans, called *independent* multispans, that satisfies certain additional properties. This is where our terminology for independent lens spans originates. We will have more to say about Simpson's independent products and local independent products at the end of this paper.

The lens span $(\overline{G}, \overline{F})$ in Example 4.2 is independent.

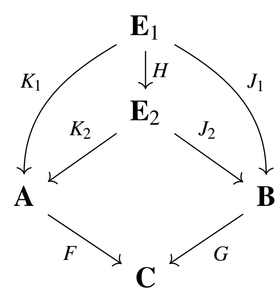
5 Necessity of Compatibility and Independence

Proxy-pullback spans of a lens cospan are, by definition, compatible with the cospan. In this section, we will show that proxy-pullback spans are also independent, and that compatibility and independence of lens spans are preserved by precomposition with lenses. It follows that whenever a lens span that commutes with a lens cospan has a comparison lens to the proxy pullback of the cospan, the span is necessarily independent and compatible with the cospan.

Proposition 5.1. *All proxy-pullback spans are independent.*

Proof. Let $\mathbf{A} \xleftarrow{\overline{G}} \mathbf{D} \xrightarrow{\overline{F}} \mathbf{B}$ be a proxy pullback of some lens cospan. For all $D \in |\mathbf{D}|$, and all $d, d' \in \mathbf{D}(D, *)$, if $\overline{F}d = \overline{F}d'$ and $\overline{G}d = \overline{G}d'$ then $d = d'$ by the universal property of the pullback in $\mathscr{C}at$ underlying the proxy pullback. In particular, this holds for those objects and morphisms in the apex of $\mathscr{M}(\overline{G}, \overline{F})$. $\square$

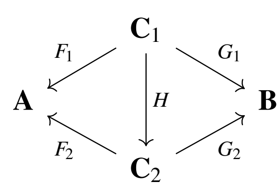
Proposition 5.2. *Consider the following diagram in $\mathscr{L}ens$, where $K_1 = K_2 \circ H$ and $J_1 = J_2 \circ H$.*



If the span (K_2, J_2) is compatible with the cospan (F, G), then so is the span (K_1, J_1).

Proof. Suppose that the span (K_2, J_2) is compatible with the cospan (F, G). Then the span (K_1, J_1) forms a commuting square with the cospan (F, G). One of the compatibility conditions holds because $J_1 {K_1}^E a = J_2 H H^E {K_2}^{HE} a = J_2 {K_2}^{HE} a = G^{J_2 HE} Fa = G^{J_1 E} Fa$, and the other holds similarly. $\square$

Proposition 5.3. *Consider the following commuting diagram in $\mathscr{L}ens$.*



If the span (F_2, G_2) is independent, then the span (F_1, G_1) is also independent.

Proof. Suppose that c and c' are morphisms in the apex of $\mathscr{M}(F_1, G_1)$ with the same source object C such that $F_1c = F_1c'$ and $G_1c = G_1c'$. Then Hc and Hc' are morphisms in the apex of $\mathscr{M}(F_2, G_2)$ with the same source object HC such that $F_2Hc = F_2Hc'$ and $G_2Hc = G_2Hc'$. As (F_2, G_2) is independent, actually $Hc = Hc'$. But c and c' are both composites of lifts along $H \circ F_2$ and $H \circ G_2$, so they are both lifts along H, and thus $c = H^CHc = H^CHc' = c'$. □

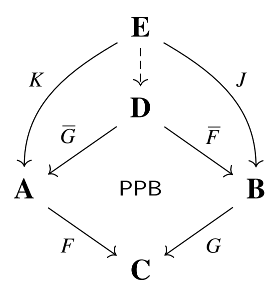
Although compatibility and independence give necessary conditions for the existence of a comparison lens, these conditions are not sufficient ones.

Example 5.4. Consider again the proxy-pullback square in Example 4.2. The sync-minimal core $\mathscr{M}(\overline{G}, \overline{F})$ of $(\overline{G}, \overline{F})$ is obtained by removing the morphism (a', b') from $\mathbf{D}$. Although the span $\mathscr{M}(\overline{G}, \overline{F})$ is independent and compatible with the cospan (F, G), there is no comparison *lens* from it to the proxy-pullback span $(\overline{G}, \overline{F})$. Assume that such a comparison lens exists. Then, as the put cofunctor of the comparison lens commutes with the put cofunctors of the legs of both spans, all of the morphisms in the apex of $\mathscr{M}(\overline{G}, \overline{F})$ are necessarily lifts of the corresponding morphisms in $\mathbf{D}$. The PutGet axiom necessitates that the lift by such a comparison lens of the morphism (a', b') into the apex of $\mathscr{M}(\overline{G}, \overline{F})$ be distinct from the lifts of the other morphisms (a', b), (a, b') and (a, b), but there is no such morphism.

6 Necessity and Sufficiency of Sync Minimality

In the previous section we saw that a lens span that commutes with a lens cospan and has a comparison lens to the proxy pullback of the cospan is necessarily independent and compatible with the cospan. It turns out that if the proxy pullback is also sync minimal, then these conditions are also sufficient.

Theorem 6.1. *Consider the following commuting diagram in $\mathscr{L}ens$.*



Suppose that the span (K, J) is independent and is compatible with the cospan (F, G). If the span $(\overline{G}, \overline{F})$ is sync minimal, then there is a unique lens $\mathbf{E} \to \mathbf{D}$ such that the triangles commute.

Proof. Suppose that $(\overline{G}, \overline{F})$ is sync minimal. Let $L\colon \mathbf{E} \to \mathbf{D}$ be the unique comparison functor from the span $(\mathscr{G}K, \mathscr{G}J)$ to the pullback span $(\mathscr{G}\overline{G}, \mathscr{G}\overline{F})$. If there is a lens structure on L that makes the triangles commute, then, for all $E \in |\mathbf{E}|$, all $a \in \mathbf{A}(KE, *)$ and all $b \in \mathbf{B}(JE, *)$, we necessarily have

$$L^E \overline{G}^{LE} a = K^E a \qquad \text{and} \qquad L^E \overline{F}^{LE} b = J^E b;$$

that is, the lifts by L of those morphisms of $\mathbf{D}$ that are lifts by $\overline{G}$ and $\overline{F}$ are determined by the lifts by K and J. As $(\overline{G}, \overline{F})$ is sync minimal, each morphism in $\mathbf{D}$ is a composite of lifts by $\overline{G}$ and lifts by $\overline{F}$, and so the above equations and the PutPut axiom for L determine the lifts by L of all morphisms of $\mathbf{D}$. Such a lens structure on L is thus uniquely determined if it exists.

In order to define $L^E d$ using the above equations, we need to check, for any two decompositions

$$\overline{F}^{D_n} b_n \circ \overline{G}^{D_{n-1}} a_{n-1} \circ \cdots \circ \overline{F}^{D_1} b_1 \circ \overline{G}^{LE} a_0 \qquad \text{and} \qquad \overline{F}^{D'_m} b'_m \circ \overline{G}^{D'_{m-1}} a'_{m-1} \cdots \circ \overline{F}^{D'_1} b'_1 \circ \overline{G}^{LE} a'_0$$

of d, that

$$J^{E_n} b_n \circ K^{E_{n-1}} a_{n-1} \circ \cdots \circ J^{E_1} b_1 \circ K^E a_0 = J^{E'_m} b'_m \circ K^{E'_{m-1}} a'_{m-1} \cdots \circ J^{E'_1} b'_1 \circ K^E a'_0.$$

Using the compatibility of both (K,J) and $(\overline{G},\overline{F})$ with (F,G), we see that

$$\begin{aligned}
\overline{F} d &= \overline{F}\big(\overline{F}^{D_n} b_n \circ \overline{G}^{D_{n-1}} a_{n-1} \circ \cdots \circ \overline{F}^{D_1} b_1 \circ \overline{G}^{LE} a_0\big) \\
&= \overline{F}\,\overline{F}^{D_n} b_n \circ \overline{F}\,\overline{G}^{D_{n-1}} a_{n-1} \circ \cdots \circ \overline{F}\,\overline{F}^{D_1} b_1 \circ \overline{F}\,\overline{G}^{LE} F a_0 \\
&= b_n \circ G^{\overline{F} D_{n-1}} F a_{n-1} \circ \cdots \circ b_1 \circ G^{\overline{F} LE} F a_0 \\
&= b_n \circ J K^{E_{n-1}} a_{n-1} \circ \cdots \circ b_1 \circ J K^E a_0 \\
&= J J^{E_n} b_n \circ J K^{E_{n-1}} a_{n-1} \circ \cdots \circ J J^{E_1} b_1 \circ J K^E a_0 \\
&= J\big(J^{E_n} b_n \circ K^{E_{n-1}} a_{n-1} \circ \cdots \circ J^{E_1} b_1 \circ K^E a_0\big)
\end{aligned}$$

Similarly, we see that

$$\begin{aligned}
J\big(J^{E'_m} b'_m \circ K^{E'_{m-1}} a'_{m-1} \cdots \circ J^{E'_1} b'_1 \circ K^E a'_0\big) &= \overline{F} d \\
K\big(J^{E_n} b_n \circ K^{E_{n-1}} a_{n-1} \circ \cdots \circ J^{E_1} b_1 \circ K^E a_0\big) &= \overline{G} d \\
K\big(J^{E'_m} b'_m \circ K^{E'_{m-1}} a'_{m-1} \cdots \circ J^{E'_1} b'_1 \circ K^E a'_0\big) &= \overline{G} d,
\end{aligned}$$

and so

$$J^{E_n} b_n \circ K^{E_{n-1}} a_{n-1} \circ \cdots \circ J^{E_1} b_1 \circ K^E a_0 = J^{E'_m} b'_m \circ K^{E'_{m-1}} a'_{m-1} \cdots \circ J^{E'_1} b'_1 \circ K^E a'_0$$

by the independence of (K,J).

We now check that L, with lifts defined in this way, satisfies the lens axioms. The PutPut axiom is immediate from the definition of L, the PutId axiom follows from that of K (or of J), and the PutGet axiom holds because

$$LK^E a = \langle \overline{G} L K^E a, \overline{F} L K^E a \rangle = \langle K K^E a, J K^E a \rangle = \langle a, G^{JE} F a \rangle = \langle a, G^{\overline{F} LE} F a \rangle = \overline{G}^{LE} a$$

and similarly $LJ^E b = \overline{F}^{LE} b$ for each $E \in |\mathbf{E}|$, each $a \in \mathbf{A}(KE,*)$ and each $b \in \mathbf{B}(JE,*)$. The relevant triangles of lenses commute by definition. □

Although the sync minimality of a proxy pullback of a lens cospan is sufficient for the existence of a comparison lens to the proxy-pullback span from an independent lens span that is compatible with the lens cospan, it is not in general necessary. For example, there is always a comparison lens from any proxy-pullback span to itself, namely, the identity lens on its apex. However, sync minimality is in fact necessary for there to be such comparison lenses simultaneously from all of the independent lens spans that are compatible with the lens cospan.

Theorem 6.2. *Consider the proxy-pullback square in $\mathscr{L}ens$ depicted below.*

$$\begin{array}{ccc}
\mathbf{D} & \xrightarrow{\overline{F}} & \mathbf{B} \\
{\scriptstyle \overline{G}}\big\downarrow & \mathsf{PPB} & \big\downarrow{\scriptstyle G} \\
\mathbf{A} & \xrightarrow[F]{} & \mathbf{C}
\end{array}$$

If the proxy-pullback span $(\overline{G},\overline{F})$ is terminal amongst the independent spans that are compatible with the cospan (F,G), then the proxy-pullback span $(\overline{G},\overline{F})$ is sync minimal.

To prove this proposition, we will consider what happens when there is a comparison lens to a proxy pullback from its sync-minimal core.

Lemma 6.3. *Let* $\mathbf{A} \xleftarrow{\overline{G}} \mathbf{D} \xrightarrow{\overline{F}} \mathbf{B}$ *be a lens span. The functor* $E_{(\overline{G},\overline{F})}$ *has a lens structure if and only if it is the identity functor on* $\mathbf{D}$*, in which case* $(\overline{G},\overline{F}) = \mathscr{M}(\overline{G},\overline{F})$ *is sync minimal.*

Proof. By construction, the functor $E_{(\overline{G},\overline{F})}$ is an identity function on objects and is a subset inclusion on morphisms. A lens that is surjective on objects is also surjective on morphisms [6]. Hence, if $E_{(\overline{G},\overline{F})}$ is the get functor of a lens, then it is surjective on morphisms and thus actually the identity functor. Conversely, if $E_{(\overline{G},\overline{F})}$ is the identity functor on $\mathbf{D}$, then it is the get functor of the identity lens on $\mathbf{D}$. □

Proof of Theorem 6.2. Suppose that $(\overline{G},\overline{F})$ is terminal amongst the independent spans that are compatible with (F,G). As the span $(\overline{G},\overline{F})$ is a proxy pullback, it is by definition compatible with the cospan (F,G), and it is independent by Proposition 5.1. The independence of the span $\mathscr{M}(\overline{G},\overline{F})$ and its compatibility with the cospan (F,G) follows from these properties of the span $(\overline{G},\overline{F})$; the former from the way that the sync-minimal core is defined, and the latter because independence is defined in terms of the sync-minimal core. By our assumption, there is thus a comparison lens H from the span $\mathscr{M}(\overline{G},\overline{F})$ to the span $(\overline{G},\overline{F})$. By the universal property of the pullback span $(\mathscr{G}\overline{G},\mathscr{G}\overline{F})$ in $\mathscr{C}at$, the functors $\mathscr{G}H$ and $E_{(\overline{G},\overline{F})}$ are both the unique comparison functor from the span of get functors of $\mathscr{M}(\overline{G},\overline{F})$ to the pullback span $(\mathscr{G}\overline{G},\mathscr{G}\overline{F})$, and so they are necessarily equal. The result then follows by Lemma 6.3. □

7 Proxy Pullbacks of Split Opfibrations

In the remainder of this paper, we unpack the results in the previous two sections for the proxy pullback of a lens cospan with additional known properties. In this section, we consider what happens when one of the legs of the cospan is a split opfibration.

Proposition 7.1. *A proxy-pullback span of a lens cospan with one leg a split opfibration is terminal amongst the independent lens spans that are compatible with the cospan.*

Proposition 7.1 follows directly from Theorem 6.1 and the following lemma.

Lemma 7.2. *Consider a proxy-pullback square*

$$\begin{array}{ccc} \mathbf{D} & \xrightarrow{\overline{F}} & \mathbf{B} \\ {\scriptstyle \overline{G}}\downarrow & \text{PPB} & \downarrow{\scriptstyle G} \\ \mathbf{A} & \xrightarrow[F]{} & \mathbf{C} \end{array} .$$

If F or G is a split opfibration then the lens span $(\overline{G},\overline{F})$ *is sync minimal.*

Proof. Without loss of generality, suppose that F is a split opfibration. Let $d\colon D_1 \to D_2$ be a morphism in $\mathbf{D}$, and let $a = \overline{G}d\colon A_1 \to A_2$ and $b = \overline{F}d\colon B_1 \to B_2$. Let u be the unique comparison morphism from the F-opcartesian morphism $F^{A_1}Fa$ to a, as in the diagram

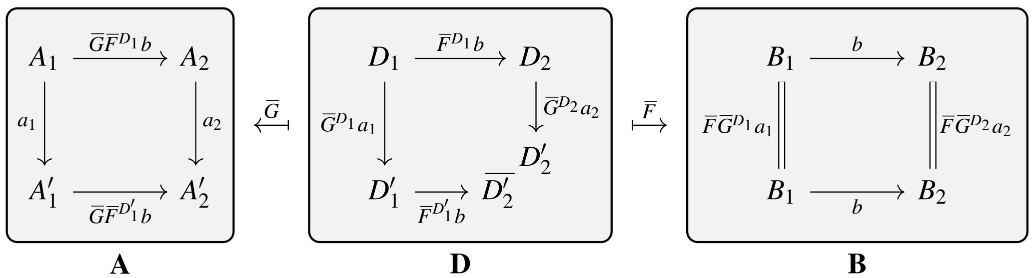
$$\begin{array}{ccc} A_1 & \xrightarrow{F^{A_1}Fa} & A_2' \\ & {\scriptstyle a}\searrow & \downarrow{\scriptstyle u} \\ & & A_2 \end{array} .$$

Then $d = \langle a, b\rangle = \langle u, \mathrm{id}_{B_2}\rangle \circ \langle F^{A_1}Fa, b\rangle = \langle u, G^{B_2}Fu\rangle \circ \langle F^{A_1}Gb, b\rangle = \overline{G}^{\langle A_2', B_2\rangle}u \circ \overline{F}^{\langle A_1, B_1\rangle}b.$ □

Remark 7.3. Split opfibrations are pullback stable, so in the proof above $\overline{F}$ is actually a split opfibration. However, lens spans with one leg a split opfibration are not in general sync minimal.

Shortly we will see that for a lens cospan with one leg a split opfibration, the independence condition for lens spans forming compatible squares with the cospan is equivalent to a simpler notion of independence, which we will call *split independence*.

Definition 7.4. A lens span $\mathbf{A} \xleftarrow{\overline{G}} \mathbf{D} \xrightarrow{\overline{F}} \mathbf{B}$ is called $\overline{F}$-*split independent* if for all $D_1 \in |\mathbf{D}|$, all $a_1\colon \overline{G}D_1 = A_1 \to A_1'$ in $\mathbf{A}$, all $b\colon \overline{F}D_1 = B_2 \to B_2$ in $\mathbf{B}$, and all $a_2\colon \operatorname{tgt}\overline{G}\overline{F}^{D_1}b = A_2 \to A_2'$, as shown in the diagram



, (2)

whenever the square in $\mathbf{A}$ commutes and $\overline{F}\overline{G}^{D_1}a_1 = \mathrm{id}_{B_1}$ and $\overline{F}\overline{G}^{D_2}a_2 = \mathrm{id}_{B_2}$, then also $D_2' = \overline{D_2'}$ and the resulting square in $\mathbf{D}$ commutes.

Proposition 7.5. *Consider a compatible lens square*

$$\begin{array}{ccc} \mathbf{D} & \xrightarrow{\overline{F}} & \mathbf{B} \\ {\scriptstyle \overline{G}}\downarrow & & \downarrow{\scriptstyle G} \\ \mathbf{A} & \xrightarrow[F]{} & \mathbf{C} \end{array} .$$

If F is a split opfibration, then $(\overline{G},\overline{F})$ is independent if and only if it is $\overline{F}$-split independent.

The *only if* direction follows directly from the definition of independence. Essential to the proof of the *if* direction is the following lemma.

Lemma 7.6. *Suppose that F is a split opfibration and $(\overline{G},\overline{F})$ is $\overline{F}$-split independent. Then, each morphism $d\colon D_1 \to D_2$ in the apex of $\mathscr{M}(\overline{G},\overline{F})$ has the factorisation*

$$\begin{array}{ccc} D_1 & \xrightarrow{\overline{F}^{D_1}\overline{F}d} & D_2' \\ & {\scriptstyle d}\searrow & \downarrow{\scriptstyle \overline{G}^{D_2'}u} \\ & & D_2 \end{array} \tag{3}$$

where $u\colon \overline{G}D_3 \to \overline{G}D_2$ comes from the universal property of the F-opcartesian morphism $F^{\overline{G}D_1}G\overline{F}d$, that is, u is the unique morphism of $\mathbf{A}$ for which $Fu = \mathrm{id}_{F\overline{G}D_2}$ and the diagram

$$\begin{array}{ccc} \overline{G}D_1 & \xrightarrow{F^{\overline{G}D_1}G\overline{F}d} & \overline{G}D_2' \\ & {\scriptstyle \overline{G}d}\searrow & \downarrow{\scriptstyle u} \\ & & \overline{G}D_2 \end{array} \tag{4}$$

commutes. In particular, the right leg of $\mathscr{M}(\overline{G},\overline{F})$ is also a split opfibration.

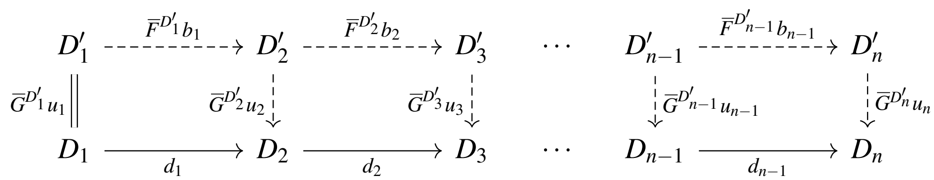
We will return to prove the lemma shortly, but let us first finish the proof of the proposition.

Proof of if *direction of Proposition 7.5.* If F is a split opfibration and $(\overline{G},\overline{F})$ is $\overline{F}$-split independent, then Lemma 7.6 implies that each morphism d in the apex of $\mathscr{M}(\overline{G},\overline{F})$ is uniquely determined by the data $\overline{G}d$ and $\overline{F}d$. Indeed, from (3), d is a composite of morphisms expressed in terms of $\overline{F}d$ and u, and u itself is uniquely determined by the top side and hypotenuse of the triangle (4), which are themselves expressed in terms of $\overline{G}d$ and $\overline{F}d$. □

Proof of Lemma 7.6. Recall that each morphism in the apex of $\mathscr{M}(\overline{G},\overline{F})$ is a composite

$$D_1 \xrightarrow{d_1} D_2 \xrightarrow{d_2} D_3 \quad \cdots \quad D_{n-1} \xrightarrow{d_{n-1}} D_n$$

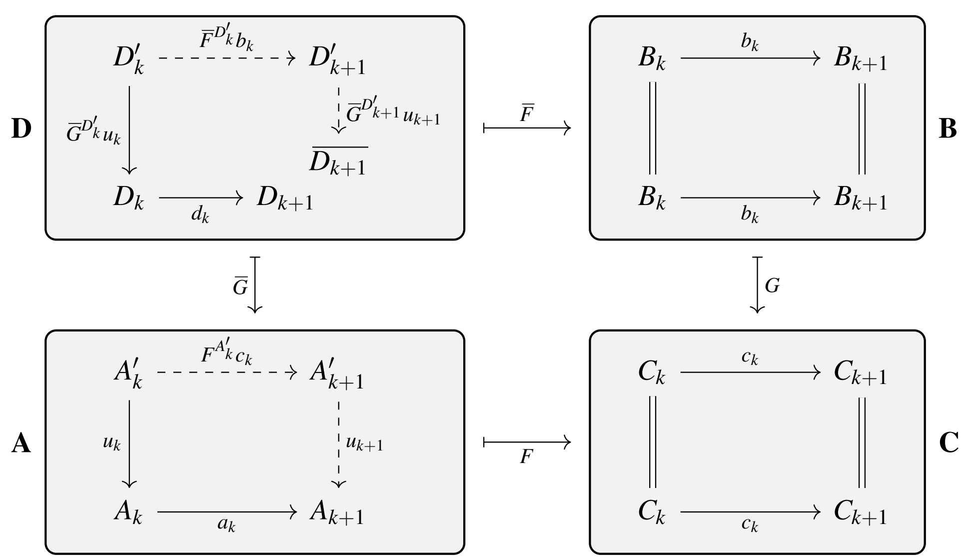
of morphisms in **D** such that, for each k, either $d_k = \overline{G}^{D_k} a_k$ or $d_k = \overline{F}^{D_k} b_k$, where $a_k = \overline{G} d_k$ and $b_k = \overline{F} d_k$. We will inductively construct the dashed morphisms in the diagram



in **D** such that the resulting diagram in **D** commutes and $\overline{F}\,\overline{G}^{D'_k} u_k = \mathrm{id}_{\overline{F}D_k}$ for each k.

For the base step, we may set $D'_1 = D_1$ and $u_1 = \mathrm{id}_{\overline{G}D_1}$, so that $\overline{F}\,\overline{G}^{D'_1} u_1 = \overline{F}\,\mathrm{id}_{D_1} = \mathrm{id}_{\overline{F}D_1}$.

For the inductive step, suppose that we have already constructed D'_k and u_k, and wish now to construct D'_{k+1} and u_{k+1}. Consider the diagram



From the universal property of the F-opcartesian morphism $F^{A'_k} c_k$, there is a unique morphism $u_{k+1} \colon A'_{k+1} \to A_{k+1}$ in **A** above $\mathrm{id}_{C_{k+1}}$ such that the square in **A** above commutes.

Suppose that $d_k = \overline{G}^{D_k} a_k$. By the PutPut axiom, and commutativity of the lens square,

$$d_k \circ \overline{G}^{D'_k} u_k = \overline{G}^{D'_{k+1}} u_{k+1} \circ \overline{G}^{D'_k} F^{A'_k} c_k = \overline{G}^{D'_{k+1}} u_{k+1} \circ \overline{F}^{D'_k} G^{B_k} c_k. \tag{5}$$

We also have $\overline{F}\overline{G}^{D'_k}u_k = G^{B_k}Fu_k = G^{B_k}\mathrm{id}_{C_k} = \mathrm{id}_{B_k}$ by compatibility of the lens square, and similarly $\overline{F}\overline{G}^{D'_{k+1}}u_{k+1} = \mathrm{id}_{B_{k+1}}$. Hence, applying $\overline{F}$ to both sides of (5), we see that $b_k = G^{B_k}c_k$. Thus (5) actually says that $\overline{D_{k+1}} = D_{k+1}$ and the square in $\mathbf{D}$ above commutes.

Otherwise, $d_k = \overline{F}^{D_k}b_k$, and thus also $a_k = \overline{G}\overline{F}^{D_k}b_k$. Also, as the lens square is compatible, $F^{A'_k}c_k = \overline{G}\overline{F}^{D'_k}b_k$. As $(\overline{G},\overline{F})$ is split independent, again $\overline{D_{k+1}} = D_{k+1}$ and the square in $\mathbf{D}$ above commutes. □

8 Proxy Pullbacks of Discrete Opfibrations and Proxy Products

The results in the previous section apply in particular to proxy pullbacks of lens cospans with one leg a discrete opfibration. For proxy pullbacks of such cospans, Proposition 7.1 simplifies as follows.

Proposition 8.1. *Proxy pullbacks of discrete opfibrations are real pullbacks in $\mathscr{L}ens$.*

This result was actually first proved by Chollet et al. [6], but in a nuts-and-bolts manner rather than as a consequence of the general theory that we present in this paper.

Lemma 8.2. *Consider a compatible lens square*

$$\begin{array}{ccc} \mathbf{D} & \xrightarrow{\overline{F}} & \mathbf{B} \\ {\scriptstyle \overline{G}}\big\downarrow & & \big\downarrow{\scriptstyle G} \\ \mathbf{A} & \xrightarrow[F]{} & \mathbf{C} \end{array} \cdot$$

If F or G is a discrete opfibration, then $(\overline{G},\overline{F})$ is independent.

Proof. Without loss of generality, suppose that F is a discrete opfibration. Let $D_1 \in |\mathbf{D}|$, let $a_1 \colon \overline{G}D_1 = A_1 \to A'_1$ in $\mathbf{A}$, let $b \colon \overline{F}D_1 = B_2 \to B_2$ in $\mathbf{B}$, and let $a_2 \colon \operatorname{tgt}\overline{G}\overline{F}^{D_1}b = A_2 \to A'_2$, as shown in the diagram (2). Suppose also that the square in $\mathbf{A}$ commutes, and that $\overline{F}\overline{G}^{D_1}a_1 = \mathrm{id}_{B_1}$ and $\overline{F}\overline{G}^{D_2}a_2 = \mathrm{id}_{B_2}$. We have

$$Fa_1 = GG^{B_1}Fa_1 = G\overline{F}\overline{G}^{D_1}a_1 = G\,\mathrm{id}_{B_1} = \mathrm{id}_{GB_1} = \mathrm{id}_{FA_1}$$

by compatibility of the lens square, and so $a_1 = \mathrm{id}_{A_1}$ as F is a discrete opfibration. Hence $\overline{G}^{D_1}a_1 = \overline{G}^{D_1}\mathrm{id}_{A_1} = \mathrm{id}_{D_1}$ and $D'_1 = D_1$. Similarly, $\overline{G}^{D_2}a_2 = \mathrm{id}_{D_2}$ and $D'_2 = D_2$. As $D'_1 = D_1$, we have $\overline{F}^{D_1}b = \overline{F}^{D'_1}b$. Hence the square in $\mathbf{D}$ commutes. □

Proof of Proposition 8.1. By Proposition 7.1, a proxy-pullback span of a lens cospan with one leg a discrete opfibration is terminal amongst the independent lens spans that are compatible with the cospan. Every lens span that forms a commuting square with such a cospan is actually compatible with the cospan by Proposition 3.2 and independent by Lemma 8.2. □

Specialising further, we now consider the proxy pullbacks of those cospans whose apex is the terminal category, that is, proxy products. Recall that the unique lens from a category $\mathbf{C}$ to the terminal category is a discrete opfibration if and only if $\mathbf{C}$ is a discrete category. The specialisation of Proposition 8.1 then says that the proxy product of a category with a discrete category is a real product. Actually, in this case, the converse also holds.

Proposition 8.3. *The proxy product of two categories is a real product if and only if at least one of the two categories is a discrete category.*

To prove the converse, it suffices to show, for all non-discrete categories $\mathbf{A}$ and $\mathbf{B}$, that there is a non-independent lens span from $\mathbf{A}$ to $\mathbf{B}$. We may explicitly describe such a non-independent lens span; it is merely the so-called *funny tensor product* $\mathbf{A} \,\square\, \mathbf{B}$ of $\mathbf{A}$ and $\mathbf{B}$ with a canonical lens structure on the projection functors. Henceforth, we will refer to the funny tensor product as the *free product* of categories, as it generalises the well-known free product of groups. The free product of categories has several different descriptions; we will use the following one.

Definition 8.4. The *free product* $\mathbf{A} \,\square\, \mathbf{B}$ of categories $\mathbf{A}$ and $\mathbf{B}$ is the category with object set $|\mathbf{A}| \times |\mathbf{B}|$ whose morphisms are freely generated by those of the form

$$(A_1, B) \xrightarrow{(a,B)} (A_2, B) \qquad \text{and} \qquad (A, B_1) \xrightarrow{(A,b)} (A, B_2),$$

subject to the equations

$$\begin{aligned} (\mathrm{id}_A, B) &= \mathrm{id}_{(A,B)} & \qquad (a_2, B) \circ (a_1, B) &= (a_2 \circ a_1, B) \\ (A, \mathrm{id}_B) &= \mathrm{id}_{(A,B)} & \qquad (A, b_2) \circ (A, b_1) &= (A, b_2 \circ b_1). \end{aligned}$$

There are projection lenses $P_1 \colon \mathbf{A} \,\square\, \mathbf{B} \to \mathbf{A}$ and $P_2 \colon \mathbf{A} \,\square\, \mathbf{B} \to \mathbf{B}$, defined by the equations

$$\begin{array}{llll} P_1(A,B) = A & P_1(a,B) = a & P_1(A,b) = \mathrm{id}_A & P_1{}^{(A,B)}a = (a,B) \\ P_2(A,B) = B & P_2(A,b) = b & P_2(a,B) = \mathrm{id}_B & P_2{}^{(A,B)}b = (A,b), \end{array}$$

whose get functors are the usual projection functors.

Proof of Proposition 8.3. The *if* direction is a particular case of Proposition 8.1. For the *only if* direction, suppose that $\mathbf{A}$ and $\mathbf{B}$ are both non-discrete categories, that is, that there are non-identity morphisms $a \colon A_1 \to A_2$ in $\mathbf{A}$ and $b \colon B_1 \to B_2$ in $\mathbf{B}$. Then the morphisms

$$(A_1, B_1) \xrightarrow{(a,B_1)} (A_2, B_1) \xrightarrow{(A_2,b)} (A_2, B_2) \qquad \text{and} \qquad (A_1, B_1) \xrightarrow{(A_1,b)} (A_1, B_2) \xrightarrow{(a,B_2)} (A_2, B_2)$$

in $\mathbf{A} \,\square\, \mathbf{B}$ both have the same source object, and both are mapped by P_1 to a and P_2 to b, but they are not equal. Hence the lens span $\mathbf{A} \xleftarrow{P_1} \mathbf{A} \,\square\, \mathbf{B} \xrightarrow{P_2} \mathbf{B}$ is not independent. □

9 Conclusion

In this paper, we gave necessary and sufficient conditions for when a lens span that forms a commuting square with a lens cospan has a comparison lens to a proxy pullback of the cospan. These conditions involved the new notions of compatibility, sync minimality and independence. They enabled us to describe exactly when a proxy pullback is a real pullback, and this description simplified further for proxy products. A search for such a simplified description for general proxy pullbacks is ongoing.

We would like to obtain categorical characterisations of the notions of sync minimality and independence, perhaps in terms of some universal property. Whilst the author is yet to discover a compelling such characterisation of independent lens spans, some interesting progress has already been made for sync minimal ones. The key observation is that being sync minimal is really a property of the put cofunctors of a lens span, and that the process of taking the sync minimal core actually gives a factorisation of this span of put cofunctors. Spivak and Niu [19] show that $\mathscr{C}of$ has products; the explicit description of these products is unfortunately rather complicated—the objects of the product of two categories are certain pairs of rooted infinite trees whose edges are morphisms from either category, and the morphisms out of such an object are the paths in either tree from its root. It turns out that a cofunctor span is sync minimal exactly when its product pairing in $\mathscr{C}of$ has surjective lifting functions. We will call a cofunctor with surjective lifting functions *cofull* and one with injective lifting functions *cofaithful*. There is a well-known factorisation system on $\mathscr{C}of$ whose left class is the bijective-on-objects cofunctors and whose right class is the discrete opfibrations [7], which, in this context, we might also call the cofull cofaithful cofunctors. The factorisation system on $\mathscr{C}of$ that we are actually interested in has as its left class the cofaithful bijective-on-objects cofunctors, and its right class the cofull cofunctors; this factorisation of the put cofunctor of a lens coincides with the other factorisation. If we factor the product pairing of a cofunctor span using this factorisation system, then the sync-minimal core of the cospan is obtained by composing the second factor with the appropriate product projection cofunctors.

We have already recalled that symmetric lenses between two categories correspond to the equivalence classes of a certain equivalence relation on asymmetric lens spans between the two categories [14]. Clarke, with a different definition of symmetric lens, constructed an adjoint triple[1]

$$\mathscr{S}ym\mathscr{L}ens(\mathbf{A},\mathbf{B}) \underset{\mathscr{R}}{\overset{\mathscr{L}}{\underset{\perp}{\overset{\perp}{\leftrightarrows}}}} \mathscr{S}pan\mathscr{L}ens(\mathbf{A},\mathbf{B})$$

between his category $\mathscr{S}ym\mathscr{L}ens(\mathbf{A},\mathbf{B})$ of symmetric lenses from $\mathbf{A}$ to $\mathbf{B}$ and the category $\mathscr{S}pan\mathscr{L}ens(\mathbf{A},\mathbf{B})$ whose objects are lens spans from $\mathbf{A}$ to $\mathbf{B}$ and whose morphisms are functors satisfying certain compatibility conditions [8]. The comonad $\mathscr{L}\circ\mathscr{M}$ on $\mathscr{S}pan\mathscr{L}ens(\mathbf{A},\mathbf{B})$ induced by the adjoint triple appears to be closely related to our process that sends a lens span to its sync minimal core. Additionally, as $\mathscr{L}$ is fully faithful, we may think of those lens spans in the image of $\mathscr{L}$ as representing symmetric lenses. It might thus be reasonable to think of the sync-minimal lens spans as being the symmetric lenses, an idea that is reinforced by the interpretation of the sync-minimal property that was given in Section 4.

The original proposal for the *Categories of Maintainable Relations* project of the Applied Category Theory Adjoint School 2020, which did not end up being the actual focus of the project, was to work out how to view symmetric lenses as some kind of generalised relations in $\mathscr{L}ens$. A *relation* in a category from object X to object Y is usually defined as a jointly monic span from X to Y. A *regular category* [3] is a finitely complete category with a pullback-stable regular-epi mono factorisation system. Relations in regular categories are particularly nice as they form the morphisms of a bicategory; the composite of two relations is the image (from the factorisation system) of their composite as spans (from the pullback). Given a not-necessarily-proper orthogonal factorisation system $(\mathscr{E},\mathscr{M})$ on a category with products, an $\mathscr{M}$-relation from X to Y is a span from X to Y whose product pairing is in $\mathscr{M}$. If the factorisation system is pullback-stable, then the $\mathscr{M}$-relations still form the morphisms of a bicategory with nice properties [16, 15, 17], where composition of $\mathscr{M}$-relations is defined similarly to that of relations in a regular category. As $\mathscr{C}of$ is finitely complete, we may consider the $\mathscr{M}$-relations in $\mathscr{C}of$ for the factorisation system where $\mathscr{E}$ is the class of cofaithful bijective-on-objects cofunctors and $\mathscr{M}$ is the class of cofull cofunctors. From our earlier discussion, these $\mathscr{M}$-relations are exactly the sync-minimal cofunctor spans. It would be interesting to work out what the composition of such $\mathscr{M}$-relations is, as the pullback in $\mathscr{C}of$ is very different to the proxy pullback in $\mathscr{L}ens$. Returning to the question of whether symmetric lenses may be viewed as some kind of relations in $\mathscr{L}ens$, we seem to need a further generalisation of the notion of internal relation as the sync-minimal core of a lens span is not obtained from a factorisation system on $\mathscr{L}ens$ itself.

[1] Clarke's functor $\mathscr{M}$ is not to be confused with our $\mathscr{M}$ that sends a lens span to its sync-minimal core.

Acknowlegdements

Many thanks go to Michael Johnson, Bryce Clarke, Richard Garner, Samuel Muller, Chris Heunen, Paolo Perrone and Perdita Stevens for their helpful feedback and suggestions.

References

[1] Marcelo Aguiar (1997): *Internal Categories and Quantum Groups*. Ph.D. thesis, Cornell University. Available at `http://pi.math.cornell.edu/~maguiar/thesis2.pdf`.

[2] Danel Ahman & Tarmo Uustalu (2017): *Taking Updates Seriously*. In Romina Eramo & Michael Johnson, editors: *Proceedings of the 6th International Workshop on Bidirectional Transformations*, pp. 59–73. Available at `http://ceur-ws.org/Vol-1827/`.

[3] M. Barr, P. A. Grillet & D. H. van Osdol (1971): *Exact Categories and Categories of Sheaves*, 1 edition. Lecture Notes in Mathematics, Springer, Berlin, Heidelberg, doi:10.1007/BFb0058579.

[4] Benjamin Merlin Bumpus & Zoltan A. Kocsis (2021): *Spined categories: generalizing tree-width beyond graphs*, doi:10.48550/arXiv.2104.01841.

[5] Gabriella Böhm (2019): *Crossed Modules of Monoids I: Relative Categories*. *Applied Categorical Structures* 27(6), pp. 641–662, doi:10.1007/s10485-019-09570-0.

[6] Emma Chollet, Bryce Clarke, Michael Johnson, Maurine Songa, Vincent Wang & Gioele Zardini: *Limits and colimits in a category of lenses*. Available at `https://arxiv.org/abs/2105.05422`. To appear in: Applied Category Theory 2021.

[7] Bryce Clarke (2020): *Internal lenses as functors and cofunctors*. In John Baez & Bob Coecke, editors: Proceedings *Applied Category Theory 2019*, University of Oxford, UK, 15–19 July 2019, *Electronic Proceedings in Theoretical Computer Science* 323, pp. 183–195, doi:10.4204/EPTCS.323.13.

[8] Bryce Clarke (2021): *A diagrammatic approach to symmetric lenses*. In David I. Spivak & Jamie Vicary, editors: Proceedings of the 3rd Annual International *Applied Category Theory Conference 2020*, Cambridge, USA, 6–10th July 2020, *Electronic Proceedings in Theoretical Computer Science* 333, Open Publishing Association, pp. 79–91, doi:10.4204/EPTCS.333.6.

[9] Matthew Di Meglio: *Coequalisers under the lens*. To appear in: Applied Category Theory 2021.

[10] Matthew Di Meglio (2021): *The category of asymmetric lenses and its proxy pullbacks*, doi:10.25949/20236449.v1. MRES Thesis, Macquarie University.

[11] Zinovy Diskin, Yingfei Xiong & Krzysztof Czarnecki (2011): *From State- to Delta-Based Bidirectional Model Transformations: the Asymmetric Case*. *Journal of Object Technology* 10, pp. 6:1–25, doi:10.5381/jot.2011.10.1.a6.

[12] Zinovy Diskin, Yingfei Xiong, Krzysztof Czarnecki, Hartmut Ehrig, Frank Hermann & Fernando Orejas (2011): *From State- to Delta-Based Bidirectional Model Transformations: The Symmetric Case*. In Jon Whittle, Tony Clark & Thomas Kühne, editors: *Model Driven Engineering Languages and Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 304–318, doi:10.1007/978-3-642-24485-8_22.

[13] Jeremy Gibbons & Perdita Stevens, editors (2018): *Bidirectional Transformations*. *Lecture Notes in Computer Science* 9715, Springer International Publishing, doi:10.1007/978-3-319-79108-1.

[14] Michael Johnson & Robert Rosebrugh (2015): *Spans of Delta Lenses*. In Alcino Cunha & Ekkart Kindler, editors: *Proceedings of the 4th International Workshop on Bidirectional Transformations*, pp. 1–15. Available at `http://ceur-ws.org/Vol-1396/`.

[15] G. M. Kelly (1991): *A note on relations relative to a factorization system*. In Aurelio Carboni, Maria Cristina Pedicchio & Guiseppe Rosolini, editors: *Category Theory*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 249–261, doi:10.1007/BFb0084224.

[16] Jeanne Meisen (1974): *On Bicategories of Relations and Pullback Spans*. *Communications in Algebra* 1(5), pp. 377–401, doi:10.1080/00927877408548625.

[17] Duško Pavlović (1995): *Maps I: relative to a factorisation system*. *Journal of Pure and Applied Algebra* 99(1), pp. 9–34, doi:10.1016/0022-4049(94)00054-M.

[18] Alex Simpson (2018): *Category-theoretic Structure for Independence and Conditional Independence*. *Electronic Notes in Theoretical Computer Science* 336, pp. 281–297, doi:10.1016/j.entcs.2018.03.028.

[19] David I. Spivak & Nelson Niu (2021): *Polynomial Functors: A General Theory of Interaction*. Available at `https://topos.site/poly-book.pdf`.

[20] Adam Maher Yassine (2020): *Generalized Span Categories in Classical Mechanics and the Functoriality of the Legendre Transformation*. Ph.D. thesis, University of California Riverside. Available at `https://escholarship.org/uc/item/1zk8372c`.

Value Iteration is Optic Composition

Jules Hedges   Riu Rodríguez Sakamoto

Dynamic programming is a class of algorithms used to compute optimal control policies for Markov decision processes. Dynamic programming is ubiquitous in control theory, and is also the foundation of reinforcement learning. In this paper, we show that value improvement, one of the main steps of dynamic programming, can be naturally seen as composition in a category of optics, and intuitively, the optimal value function is the limit of a chain of optic compositions. We illustrate this with three classic examples: the gridworld, the inverted pendulum and the savings problem. This is a first step towards a complete account of reinforcement learning in terms of parametrised optics.

1 Introduction

In this paper we describe basic concepts of dynamic programming in terms of categories of optics. The class of models we consider are discrete-time Markov decision processes, aka. discrete-time controlled Markov chains. There are classical methods of computing optimal control policies, underlying much of both classical control theory and modern reinforcement learning, known collectively as *dynamic programming*. These are based on two operations that can be interleaved in many different ways: *value improvement* and *policy improvement*. The central idea of this paper is the slogan *value improvement is optic precomposition*, or said differently, *value improvement is a representable functor on optics*.

Given a control problem with state space X, a *value function* $V : X \to \mathbb{R}$ represents an estimate of the long-run payoff of following a policy starting from any state, and can be equivalently represented as a costate $V : \binom{X}{\mathbb{R}} \to I$ in a category of optics. Every control policy π also induces an optic $\lambda(\pi) : \binom{X}{\mathbb{R}} \to \binom{X}{\mathbb{R}}$. The general idea is that the forwards pass of the optic is a morphism $X \to X$ describing the dynamics of the Markov chain given the policy, and the backwards pass is a morphism $X \otimes \mathbb{R} \to \mathbb{R}$ which given the current state and the *continuation payoff*, describing the total payoff from all future stages, returns the total payoff for the current stage given the policy, plus all future stages.

Given a policy π and a value function $V : \binom{X}{\mathbb{R}} \to I$, the costate $\binom{X}{\mathbb{R}} \xrightarrow{\lambda(\pi)} \binom{X}{\mathbb{R}} \xrightarrow{V} I$ is a closer approximation of the value of π. This is called *value improvement*. Iterating this operation

$$\cdots \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \xrightarrow{\lambda(\pi)} \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \xrightarrow{\lambda(\pi)} \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \xrightarrow{V} I$$

converges efficiently to the true value function of the policy π.

Replacing π with a new policy that is optimal for its value function is called *policy improvement*. Repeating these steps is known as *policy iteration*, and converges to the optimal policy and value function.

Alternatively, instead of repeating value improvement until convergence before each step of policy improvement, we can also alternate them, giving the composition of optics

$$\cdots \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \xrightarrow{\lambda(\pi_2)} \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \xrightarrow{\lambda(\pi_1)} \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \xrightarrow{V} I$$

where each policy π_i is optimal for the value function to the right of it. This is known as *value iteration*, and also converges to the optimal policy and value function. For an account of convergence properties of these algorithms, classic textbooks are [32, Sec.6], [6, Ch.1].

J. Master & M. Lewis (Eds.): Fifth International
Conference on Applied Category Theory (ACT 2022).
EPTCS 380, 2023, pp. 417–432, doi:10.4204/EPTCS.380.24

In this paper we illustrate this idea, using mixed optics to account for the categorical structure of transitions in a Markov chain and the convex structure of expected payoffs, which typically form the kleisli and Eilenberg-Moore categories of a probability monad. This paper is partially intended as an introduction to dynamic programming for category theorists, focussing on illustrative examples rather than on heavy theory.

1.1 Related work

The precursor of this paper was early work on value iteration using open games [22]. The idea originally arose around 2016 during discussions of the first author with Viktor Winschel and Philipp Zahn. An early version was planned as a section of [25] but cut partly for page limit reasons, and partly because the idea was quite uninteresting until it was understood how to model stochastic transitions in open games [7] via optics [33]. In this paper we have chosen to present the idea without any explicit use of open games, both in order to clarify the essential idea and also to bring it closer to the more recent framework of categorical cybernetics [11], which largely subsumes open games [12]. (Although, actually using this framework properly is left for future work.)

A proof-of-concept implementation of value iteration with open games was done in 2019 by the first author and Wolfram Barfuss[1], implementing a model from [4] - a model of the social dilemma of emissions cuts and climate collapse as a stochastic game, or jointly controlled MDP - and verifying it against Barfuss' Matlab implementation. A far more advanced implementation of reinforcement learning using open games was developed recently by Philipp Zahn, currently closed-source, and was used for the paper [16].

The most closely related work to ours is [31], which formulates MDPs in terms of F-lenses [35] of the functor $\mathrm{BiKl}(C\times -, \Delta(\mathbb{R}\times -))^{\mathrm{op}}$, where $C\times -$ is the reader comonad and $\Delta(\mathbb{R}\times -)$ is a probability monad over actions with their expected value. A MDP there is a lens from states and potential state changes and rewards to the agents observation and input $\binom{X}{\Delta(X\times\mathbb{R})} \to \binom{O}{I}$. Our approach differs in two ways. We firstly assume that the readout function is the identity, as we are not dealing with partial observability [1]. Secondly, we specify a concrete structure of the backwards update map $f^* : X\times I \to \Delta(X\times R)$, which allows us to rearrange the interface of this lens from policies to value functions. Doing so opens up the possibility of composing these lenses sequentially, which is the heart of the dynamic programming approach explored in this paper.

Bakirtzis et al. propose a category of MPDs as models of tasks [3]. This emphasis on models allow them to compose different MPDs using fiber products and pushouts, and is agnostic to the control and RL algorithms that operate on them, which they take as given. Since our work focuses on a particular family of algorithms, we believe this approach is orthogonal to ours, and both could potentially be done simultaneously.

Another approach is to model MDPs as coalgebras from states to rewards and potential transitions, as done by Feys et al. [17]. They observe that the Bellman optimality condition for value iteration is a certain coalgebra-to-algebra morphism. We similarly believe this is orthogonal to our work and could potentially be unified.

A series of papers by Botta et al (for example [8]) formulates dynamic programming in dependent type theory, accounting in a serious way for how different actions can be available in different states, a complication that we ignore in this paper. It may be possible to unify these approaches using dependent optics [9, 40].

[1]Source currently available at `https://github.com/jules-hedges/open-games-hs/blob/og-v0.1/src/OpenGames/Examples/EcologicalPublicGood/EcologicalPublicGood.hs`

Finally, [2] builds a category of signal flow diagrams, a widely used tool in control theory. Besides the common application to control theory there is little connection to this paper. In particular, time is implicit in their string diagrams, meaning their models have continuous time, whereas our approach is inherently discrete time. Said another way, composition in their category is 'space-like' whereas ours is 'time-like' - their morphisms are (open) systems whereas ours are processes.

2 Dynamic programming

2.1 Markov Decision Processes

A *Markov decision process* (MDP) consists of a state space X, an action space A, a state transition function $f : X \times A \to X$, and a utility or reward function $U : X \times A \to \mathbb{R}$. The state transition function is often taken to be stochastic, that is, to be given by probabilities $f(x' \mid x, a)$. In the stochastic case the utility function can be taken without loss of generality to be an expected utility function. We imagine actions to be chosen by an agent, who is trying to *control* the Markov chain with the objective of optimising the long-run reward.

A *policy* for an MDP is a function $\pi : X \to A$, which can also be taken to be either deterministic or stochastic. The type of policies encodes the Markov property: the choice of action depends only on the current state, and may not depend on any memory of past states.

Given an initial state $x_0 \in X$, a policy π determines (possibly stochastically) a sequence of states

$$x_0, \quad x_1 = f(x_0, \pi(x_0)), \quad x_2 = f(x_1, \pi(x_1)), \quad \ldots$$

The total payoff is given by an infinite geometric sum of individual payoffs for each transition:

$$V_\pi(x_0) = \sum_{k=0}^{\infty} \beta^k U(x_k, \pi(x_k)) \tag{1}$$

where $0 < \beta < 1$ is a fixed *discount factor* which balances the relevance of present and future payoffs. (There are other methods of obtaining a single objective from an infinite sequence of transitions, such as averaging, but we focus on discounting in this paper.) A key idea behind dynamic programming is that this geometric sum can be equivalently written as a telescoping sum:

$$V_\pi(x_0) = U(x_0, \pi(x_0)) + \beta(U(x_1, \pi(x_1)) + \beta(U(x_2, \pi(x_2) + \cdots)))$$

The *control problem* is to choose a policy π in order to maximise (the expected value of) $V_\pi(x_0)$. In terms of decision theory, we assume that the agent choosing the policy operates under rational behaviour. Continuous and independent preferences of outcome implies by the von Neumann-Morgenstern expected utility theorem that the utility function has as codomain the reals.

2.2 Deterministic dynamic programming

In dynamic programming, the agent's objective of maximizing the overall utility can be divided into two orthogonal goals: to determine the value of a given policy π (which we call the *value improvement* step), and to determine the optimal policy π^* (the *policy improvement* step). Bellman's equation is used as an update rule for both:

$$\textbf{Value improvement:} \qquad V'(x) = U(x, \pi(x)) + \beta V(f(x, \pi(x))) \tag{2}$$

$$\textbf{Policy improvement:} \qquad \pi'(x) = \arg\max_{a \in A} U(x, a) + \beta V(f(x, a)) \tag{3}$$

A Bellman optimality condition on the other hand determines the fixpoint of this update rule, and is met when $V' = V$ and $\pi' = \pi$ respectively.

The update rule (2) is the discounted sum (1) where the stream of states is co-recursively fixed by the policy π and transition function f. The co-recursive structure refers to the calculation of the utility of a state x, where one needs the utility of the *next* state, while in a recursive structure, x needs the *previous* state, starting from an initial state as a base case.

Two classical algorithms use these two steps differently: Policy iteration iterates value improvement until the current policy value is optimal before performing a policy improvement step, and value iteration interleaves both steps one after another.

In policy iteration, a initial value function is chosen (usually $V(x) = 0$), and a randomly chosen policy π is evaluated by (2) repeatedly until the value reaches a fixpoint, which is assured by the contraction mapping [15]. Once V reaches (or in practice gets close to) a fixpoint $V' = V$ or another convergence condition, the policy improvement step (3) chooses a greedy policy as an improvement to π.

A q-function or *state-action value function* $q_\pi : X \times A \to \mathbb{R}$ describes the value of being in state x and then taking action a, assuming that subsequent actions are taken by the policy π

$$q_\pi(x,a) = U(x,a) + \beta V(f(x,a)) \tag{4}$$

The *policy improvement theorem* [5] states that if a pair of deterministic policies $\pi, \pi' : X \to A$ satisfies for all $x \in X$

$$q_\pi(x, \pi'(x)) \geq V_\pi(x)$$

then $V_{\pi'}(x) \geq V_\pi(x)$ for all $x \in X$.

The optimal policy π^*, if it exists, is the policy which if followed from any state, generates the maximum value. This is a Bellman optimality condition which fuses the two steps (2), (3):

$$V_{\pi^*}(x) = \max_{a \in A} U(x,a) + \beta V_{\pi^*}(f(x,a)) \tag{5}$$

Value iteration is a special policy iteration algorithm insofar it stops the update rule for value improvement to one step, by truncating the sum (1) to the first summand. Moreover, it introduces the value improvement step implicitly in the policy improvement, which assigns a value to states

$$V'(x) = \max_{a \in A} U(x,a) + \beta V(f(x,a))$$

while the policy in each iteration is still recoverable as

$$\pi'(x) = \arg\max_{a \in A} U(x,a) + \beta V(f(x,a))$$

2.3 Stochastic dynamic programming

Stochasticity can be introduced in different places in a MDP:

1. in the policy $\pi : X \to \Delta A$, where the probability of the policy π taking action a in a state x is now notated $\pi(a \mid x)$.
2. in the transition function $f : X \times A \to \Delta X$ and potentially the reward function $U : X \times A \to \Delta\mathbb{R}$ independently.

3. usually the reward is included inside the transition function $f : X \times A \to \Delta(X \times \mathbb{R})$, allowing correlated next states and rewards. This is relevant when the reward is morally from the next state, rather than the current state and action. If the reward were truly from the current state and action, the transition function can be decomposed into a function $f : X \times A \to \Delta X \times \Delta\mathbb{R}$.

In this section we assume for simplicity that Δ is the finite support distribution monad, although the equations in the following can be formulated for arbitrary distributions by replacing the sum with an appropriate integral.

The policy value update rule (2) becomes stochastic, and adopts a slightly different form depending on which part of the MDP is stochastic. For the cases 1. and 2.:

$$V'(x) = \sum_a \pi(a \mid x)(U(x,a) + \beta V(f(x,a))) \tag{6}$$

$$V'(x) = \sum_r U(r \mid x,a)r + \sum_{x'} f(x' \mid x,a)\beta V(x') \tag{7}$$

In the most general case, that is 1. together with 3.:

$$V'(x) = \sum_{a \in A} \pi(a \mid x) \sum_{x',r} f(x',r \mid x,a)(r + \beta V(x')) \tag{8}$$

(Note that the sum over r is over the support of $f(- \mid x,a)$, which we assume here to be finite, although in general it can be replaced with an integral.)

The policy improvement theorem holds in the stochastic setting [38, Sec.4.2] by defining

$$q_\pi(s, \pi'(s)) = \sum_a \pi'(a \mid s) q_\pi(s,a)$$

2.4 Gridworld example

A classic example in reinforcement learning is the Gridworld environment, where an agent moves in the four cardinal directions in a rectangular grid. States of this finite MDP correspond to the positions that the agent can be in.

Assume that all transitions and policies are deterministic, and that the transition function prevents the agent from moving outside the boundary. Suppose that the environment rewards 0 value for all states except the top left corner, where the reward is 1 (see figure 1).

Starting with a policy which moves upwards in all states and a value function which rewards 1 only in the top left corner, a policy iteration algorithm would improve the value of the current policy until converging to the optimal values in the leftmost column, before updating the policy, while a value iteration algorithm would update the value function and also update the policy.

Take the finite set of positions as the state space X, and $A = \{\leftarrow, \rightarrow, \uparrow, \downarrow\}$ as the action space.

This example can be made stochastic if we add stochastic policies like ε-greedy, where the action that the agent takes is the one with maximum value with probability $1 - \varepsilon$ and a random one with probability ε. Another way is for the transition function to be stochastic, for example with a wind current that shifts the next state to the right with some probability ε.

2.5 Inverted pendulum example

A task that illustrates a continuous state space MDP is the control of a pendulum balanced over a cart, which can be described in continuous-time exactly by two non-linear differential equations [18, Example

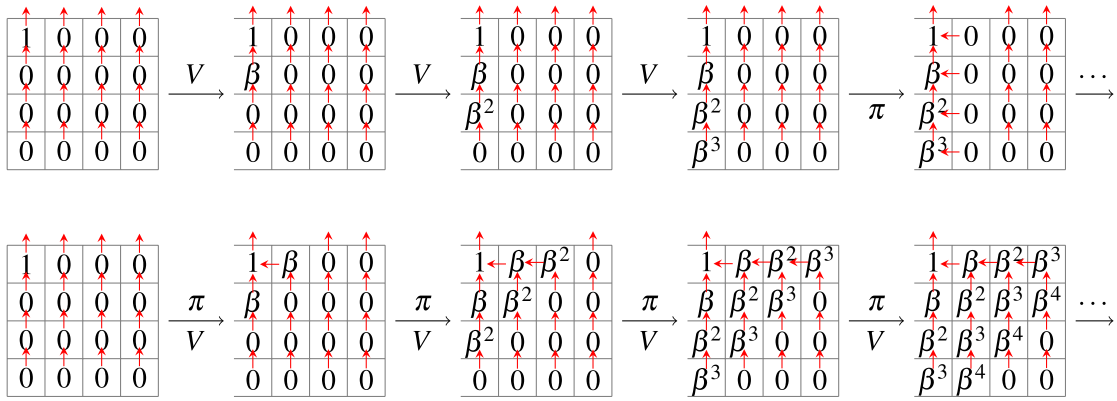


Figure 1: Difference between policy iteration (above) and value iteration (below). The numbers in the cells are state values and the red arrows are the directions dictated by the policy at each stage. The arrows between grids indicate what kind of update the algorithm does, either value improvement (V) or policy improvement (π). Notice how policy iteration performs value improvement three times before updating the policy, whereas value iteration improves the value and the policy at each stage.

2E]:

$$(M+m)\ddot{y} + mL\ddot{\theta}\cos\theta - mL\dot{\theta}^2\sin\theta = a$$
$$mL\ddot{y}\cos\theta + mL^2\ddot{\theta} - mLg\sin\theta = 0$$

where M is the mass of the cart, m the mass of the pendulum, L the length of the pendulum, θ the angle of the pendulum with respect to the upwards position, y the carts horizontal position, g the gravitational constant and a is our control function (usually denoted u). We rewrite the state variables as $x = [y, \dot{y}, \theta, \dot{\theta}]^\top$.

Sampling the trajectory of continuous-time dynamics $\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f(x(t))$ by $x_k = x(k\Delta t)$, one can define the discrete-time propagator $F_{\Delta t}$ by

$$F_{\Delta t}(x(t)) = x(t) + \int_t^{t+\Delta t} f(x(\tau))\mathrm{d}\tau$$

which allows to model the system with $x_{k+1} = F_{\Delta t}(x_k)$.

A more common approach is to observe that the system of equations $\dot{x} = A(x) + B(x)a$ with A and B being non-linear functions of the state space, can be *linearized* near a (not necessarily stable) equilibrium state, like the pendulum being in the upwards position. There we can assume certain approximations like $\cos\theta \approx 1$ and $\sin\theta \approx \theta$, as well as small velocities leading to negligible quadratic terms $\dot{\theta}^2 \approx 0$ and $\dot{y}^2 \approx 0$. This linearization around a fixpoint allows for the expression $\dot{x} = Ax(t) + Ba(t)$, where the matrix A and vector B are constants given by

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)g}{ML} & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{ML} \end{pmatrix}$$

If we assume that the observation of the pendulum angle and cart position is discretized in time, an a priori time-discretization of this model using Euler approximation follows $x_{k+1} = x_k + \Delta t(Ax_k + Ba_k)$, with the same constants, where k indexes time steps. Therefore we can say that the time-discretized, linearized model of the inverted pendulum over a cart follows a deterministic MDP for which a controller u can be learned. We take the state space as $\mathbb{R}^4$ and the action space of the force exerted to the cart as $\mathbb{R}$.

The time-discretised formulation of this problem is more common in reinforcement learning settings than in 'classical' control theory. In that case, a a common payoff function is to obtain one unit of reward for each time step that the pendulum is maintained within a threshold of angles. The not linearized, not time-discretized setting, which is more common in optimal control theory, allows the reward, which is usually termed negatively as a cost function J, to have a much more flexible expression, in terms of time spent towards the equilibrium, energy spent to control the device, etc.

$$J(x,a) = \int_0^\infty C(x(t),a(t))\mathrm{d}t$$

2.6 Savings problem example

The *savings problem* is one of the most important models in economics, modelling the dilemma between saving and consumption of resources [28, part IV] (see also [37]). It is also mathematically closely related to the problem of charging a battery, for example choosing when to draw electricity from a power grid to raise the water level in a reservoir [30].

At each discrete time step k, an agent receives an income i_k. They also have a bank balance x_k, which accumulates interest over time (this could also be, for example, an investment portfolio yielding returns). At each time step the agent makes a choice of *consumption*, which means converting their income into utility (or, more literally, things from which they derive utility). If the consumption in some stage is less than their income then the difference is added to the bank balance, and if it is more than the difference is taken from the bank balance. The dilemma is that the agent receives utility only from consumption, but saving gives the possibility of higher consumption later due to interest. The optimal balance between consumption and saving depends on the discount factor, which models the agent's preference between consumption now and consumption in the future.

In the most basic version of the model, all values can be taken as deterministic, and the income i_k can also be taken as constant. This basic model can be expanded in many ways, for example with forecasts and uncertainty about income and interest rates. A straightforward extension, which we will consider in this paper, is that income is normally distributed $i \sim \mathcal{N}(\mu,\sigma)$, independently in each time step.

We take the state space and action space both as $X = A = [0,\infty)$. Given the current bank balance x and consumption decision a, the utility in the current stage is $U(x,a) = \min\{a, x+i\}$. (That is, the agent's consumption is capped by their current bank balance.) The state transition is given by $f(x,a) = \max\{(1+\gamma)x - a + i, 0\}$, where γ is the interest rate.

3 Optics

In this section we recall material on categories of mixed optics, mostly taken from [13].

3.1 Categories of optics

Given a monoidal category $\mathscr{M}$ and a category $\mathscr{C}$, an action of $\mathscr{M}$ on $\mathscr{C}$ is given by a functor $\bullet : \mathscr{M} \times \mathscr{C} \to \mathscr{C}$ with coherence isomorphisms $I \bullet X \cong X$ and $(M \otimes N) \bullet X \cong M \bullet (N \bullet X)$. $\mathscr{C}$ is called an $\mathscr{M}$-actegory.

Given a pair of $\mathscr{M}$-actegories $\mathscr{C},\mathscr{D}$, we can form the category of optics $\mathbf{Optic}_{\mathscr{C},\mathscr{D}}$. Its objects are pairs $\binom{X}{X'}$ where X is an object of $\mathscr{C}$ and X' is an object of $\mathscr{D}$. Hom-sets are defined by the coend

$$\mathbf{Optic}_{\mathscr{C},\mathscr{D}}\left(\begin{pmatrix}X\\X'\end{pmatrix},\begin{pmatrix}Y\\Y'\end{pmatrix}\right) = \int^{M:\mathscr{M}} \mathscr{C}(X, M \bullet Y) \times \mathscr{D}(M \bullet Y', X')$$

in the category **Set**. Such a morphism is called an optic, and consists of an equivalence class of triples (M, f, f') where M is an object of $\mathscr{M}$, $f : X \to M \bullet Y$ in $\mathscr{C}$ and $g : M \bullet Y' \to X'$ in $\mathscr{D}$. We call M the residual, f the forwards pass and f' the backwards pass, so we think of the residual as mediating communication from the forward pass to the backward pass. Composition of optics works by taking the monoidal product in $\mathscr{M}$ of the residuals.

A common example is a monoidal category $\mathscr{M} = \mathscr{C} = \mathscr{D}$ acting on itself by the monoidal product, so

$$\mathbf{Optic}_{\mathscr{C}}\left(\begin{pmatrix}X\\X'\end{pmatrix},\begin{pmatrix}Y\\Y\end{pmatrix}\right) = \int^{M:\mathscr{C}} \mathscr{C}(X, M \otimes Y) \times \mathscr{C}(M \otimes Y', X')$$

This is the original definition of optics from [33]. If $\mathscr{C}$ is additionally cartesian monoidal then we can eliminate the coend to produce *concrete lenses*:

$$\begin{aligned}\int^{M:\mathscr{C}} \mathscr{C}(X, M \times Y) \times \mathscr{C}(M \times Y', X') &\cong \int^{M:\mathscr{C}} \mathscr{C}(X, M) \times \mathscr{C}(X, Y) \times \mathscr{C}(M \times Y', X')\\ &\cong \mathscr{C}(X, Y) \times \mathscr{C}(X \times Y', X')\end{aligned}$$

On the other hand, if $\mathscr{C}$ is monoidal closed then we can eliminate the coend in a different way to produce *linear lenses*:

$$\begin{aligned}\int^{M:\mathscr{C}} \mathscr{C}(X, M \otimes Y) \times \mathscr{C}(M \otimes Y', X') &\cong \int^{M:\mathscr{C}} \mathscr{C}(X, M \otimes Y) \times \mathscr{C}(M, [Y', X'])\\ &\cong \mathscr{C}(X, [Y', X'] \otimes Y)\end{aligned}$$

Both of these proofs use the *ninja Yoneda lemma* for coends [29].

Example 1. Let **Set** act on itself by cartesian product. Optics $\binom{X}{X'} \to \binom{Y}{Y'}$ in $\mathbf{Optic}_{\mathbf{Set}}$ can be written equivalently as pairs of functions $X \to Y$ and $X \times Y' \to X'$, or as a single function $X \to Y \times (Y' \to X')$.

Example 2. Let **Euc** be the category of Euclidean spaces and smooth functions, which is cartesian but not cartesian closed. Optics $\binom{X}{X'} \to \binom{Y}{Y'}$ in $\mathbf{Optic}_{\mathbf{Euc}}$ can be written as pairs of smooth functions $X \to Y$ and $X \times Y' \to X'$.

Example 3. Let **Mark** be the category of sets and finite support Markov kernels, which is the kleisli category of the finite support probability monad $\Delta : \mathbf{Set} \to \mathbf{Set}$. It is a prototypical example of a Markov category [20], and it is neither cartesian monoidal nor monoidal closed. Optics $\binom{X}{X'} \to \binom{Y}{Y'}$ in $\mathbf{Optic}_{\mathbf{Mark}}$ can only be written as optics, it is not possible to eliminate the coend. This is the setting used for Bayesian open games [7].

Example 4. Let **Conv** be the category of convex sets, which is the Eilenberg-Moore category of the finite support probability monad [19]. A convex set can be thought of a set with an abstract expectation operator $\mathbb{E} : \Delta X \to X$. Thus the functor $\Delta : \mathbf{Mark} \to \mathbf{Conv}$ given by $X \mapsto \Delta(X)$ on objects is fully faithful. **Conv** has finite products which are given by tupling in the usual way. **Conv** also has a closed structure: the set of convex functions $X \to Y$ themselves form a convex set $[X, Y]$ pointwise. However **Conv** is not cartesian closed: instead there is a different monoidal product making it monoidal closed [36, section

2.2] (see also [27]). This monoidal product "classifies biconvex maps" in the same sense that the tensor product of vector spaces classifies bilinear maps. The embedding $\Delta : \mathbf{Mark} \to \mathbf{Conv}$ is strong monoidal for this monoidal product, not for the cartesian product of convex sets.

We can define an action of **Mark** on **Conv**, by $M \bullet X = \Delta(M) \otimes X$ [10, section 5.5]. Together with the self-action of **Mark**, we get a category $\mathbf{Optic}_{\mathbf{Mark},\mathbf{Conv}}$ given by

$$\mathbf{Optic}_{\mathbf{Mark},\mathbf{Conv}}\left(\begin{pmatrix}X\\X'\end{pmatrix},\begin{pmatrix}Y\\Y'\end{pmatrix}\right) = \int^{M:\mathbf{Mark}} \mathbf{Mark}(X, M \otimes Y) \times \mathbf{Conv}(\Delta(M) \otimes Y', X')$$
$$\cong \int^{M:\mathbf{Mark}} \mathbf{Mark}(X, M \otimes Y) \times \mathbf{Conv}(\Delta(M), [Y', X'])$$

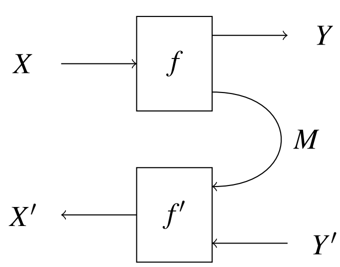
(This coend cannot be eliminated because the embedding $\Delta : \mathbf{Mark} \to \mathbf{Conv}$ does not have a right adjoint.)

This category of optics will be very useful for Markov decision processes, where the forwards direction is a Markov kernel and the backwards direction is a function involving expectations.

3.2 Monoidal structure of optics

A category of optics $\mathbf{Optic}_{\mathscr{C},\mathscr{D}}$ is itself (symmetric) monoidal, when $\mathscr{C}$ and $\mathscr{D}$ are (symmetric) monoidal in a way that is compatible with the actions of $\mathscr{M}$. The details of this have been recently worked out in [10]. The monoidal product on objects of $\mathbf{Optic}_{\mathscr{C},\mathscr{D}}$ is given by pairwise monoidal product. All of the above examples are symmetric monoidal.

A monoidal category of optics comes equipped with a string diagram syntax [24]. This has directed arrows representing the forwards and backwards passes, and right-to-left bending wires but not left-to-right bending wires. The residual of the denoted optic can be read off from a diagram, as the monoidal product of the wire labels of all right-to-left bending wires. For example, a typical optic $(M, f, f') \in \mathbf{Optic}_{\mathscr{C}}\left(\binom{X}{X'},\binom{Y}{Y'}\right)$ is denoted by the diagram



These diagrams have only been properly formalised for a monoidal category acting on itself, so for mixed optics we need to be very careful and are technically being informal.

Costates in monoidal categories of optics, that is optics $\binom{X}{X'} \to I$ (where $I = \binom{I}{I}$ is the monoidal unit of $\mathbf{Optic}_{\mathscr{C},\mathscr{D}}$), are a central theme of this paper. When we have a monoidal category acting on itself, costates in $\mathbf{Optic}_{\mathscr{C}}$ are given by

$$\mathbf{Optic}_{\mathscr{C}}\left(\begin{pmatrix}X\\X'\end{pmatrix}, I\right) = \int^{M:\mathscr{C}} \mathscr{C}(X, M \otimes I) \times \mathscr{C}(M \otimes I, X') \cong \mathscr{C}(X, X')$$

Thus *costates in optics are functions*. A different way of phrasing this is by defining a functor $K : \mathbf{Optic}_{\mathscr{C}}^{\mathrm{op}} \to \mathbf{Set}$ given on objects by $K\binom{X}{X'} = \mathscr{C}(X, X')$, and then showing that K is representable [25]. We will generally treat this isomorphism as implicit, sometimes referring to costates as though they are functions.

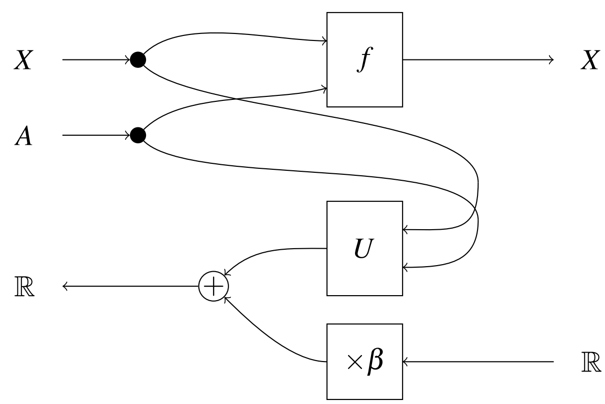
In the case of a cartesian monoidal category $\mathscr{C}$, given a concrete lens $f : X \to Y$, $f' : X \times Y' \to X'$ and a function $k : Y \to Y'$, the action of K gives us the function $X \to X'$ given by $x \mapsto f'(x, k(f(x)))$.

When we have $\mathscr{M} = \mathscr{C}$ acting on both itself and on $\mathscr{D}$ (which includes all of the examples above) then similarly

$$\mathbf{Optic}_{\mathscr{C},D}\left(\begin{pmatrix} X \\ X' \end{pmatrix}, I\right) = \int^{M:\mathscr{C}} \mathscr{C}(X, M \otimes I) \times \mathscr{D}(M \bullet I, X') \cong \mathscr{D}(X \bullet I, X')$$

4 Dynamic programming with optics

Given an MDP with state space X and action space A, we can convert it to an optic $\binom{X \otimes A}{\mathbb{R}} \to \binom{X}{\mathbb{R}}$. The category of optics in which this lives can be 'customised' to some extent, and depends on the class of MDPs that we are considering and how much typing information we choose to include. The definition of this optic is given by the following string diagram:

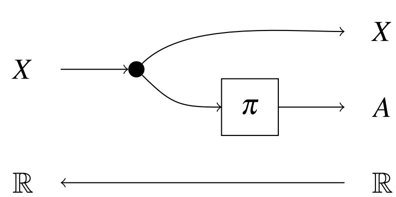


To be clear, this diagram is not completely formal because we are making some assumptions about the category of optics we work in. In general, we require the forwards category $\mathscr{C}$ to be a Markov category (giving us copy morphisms Δ_X and Δ_A), and the backwards category $\mathscr{D}$ must have a suitable object $\mathbb{R}$ together with morphisms $\times\beta : \mathbb{R} \to \mathbb{R}$ and $+ : \mathbb{R} \otimes \mathbb{R} \to \mathbb{R}$. Specific examples of interpretations of this diagram will be explored below. When the forwards category acts on the backwards category, then the forwards pass is a morphism $g : X \otimes A \to X \otimes X \otimes A$ in $\mathscr{C}$ where

$$g = \Delta_{X \otimes A} \fatsemi (f \otimes \mathrm{id}_{X \otimes A})$$

and the backwards pass is a morphism $g' : X \bullet A \bullet \mathbb{R} \to \mathbb{R}$ in $\mathscr{D}$ encoding the function $g'(x,a,r) = \mathbb{E}U(x,a) + \beta r$. The resulting optic is given by $\lambda = (X \otimes A, g, g') : \binom{X \otimes A}{\mathbb{R}} \to \binom{X}{\mathbb{R}}$ in $\mathbf{Optic}_{\mathscr{C},\mathscr{D}}$.

Given a policy $\pi : X \to A$, we lift it to an optic $\overline{\pi} : \binom{X}{\mathbb{R}} \to \binom{X \otimes A}{\mathbb{R}}$, by



Here we are also assuming that the forwards category has a copy morphisms Δ_X (for example, because it is a Markov category), and the backwards category has a suitable object of real numbers. The interpretation of this diagram is the optic $(I, \Delta_X \fatsemi (\mathrm{id}_X \otimes \pi), \mathrm{id}_{\mathbb{R}})$.

4.1 Discrete-space deterministic decision processes

Consider a deterministic decision process with a discrete set of states X, discrete and finite set of actions A, transition function $f : X \times A \to X$, payoff function $U : X \times A \to \mathbb{R}$ and discount factor $\beta \in (0,1)$. We convert this into an optic $\lambda = (X \times A, g, g') : \binom{X \times A}{\mathbb{R}} \to \binom{X}{\mathbb{R}}$ in $\mathbf{Optic}_{\mathbf{Set}}$, whose forwards pass is $g(x,a) = (x,a,f(x,a))$ and whose backwards pass is $g'(x,a,r) = U(x,a) + \beta r$.

Consider a dynamical system with the state space $A^X \times \mathbf{Optic}_{\mathbf{Set}}\left(\binom{X}{\mathbb{R}}, I\right)$. Elements of this are pairs (π, V) of a policy $\pi : X \to A$ and a value function $V : X \to \mathbb{R}$. We can define two update steps:

Value improvement:	$(\pi, V) \mapsto (\pi, \overline{\pi} \mathbin{⨾} \lambda \mathbin{⨾} V)$
Policy improvement:	$(\pi, V) \mapsto (x \mapsto \arg\max_{a \in A} (\lambda \mathbin{⨾} V)(x,a), V)$

(We assume that arg max is canonically defined, for example because A is equipped with an enumeration so that we can always choose the first maximiser.)

Unpacking and applying the isomorphism between costates in lenses and functions, a step of value improvement replaces V with

$$V'(x) = U(x, \pi(x)) + \beta V(f(x, \pi(x)))$$

and a step of policy improvement replaces π with

$$\pi'(x) = \arg\max_{a \in A} U(x,a) + \beta V(f(x,a))$$

Iterating the value improvement step converges to a value function which is the optimal value function for the current (not necessarily optimal) policy π. A fixpoint of alternating steps of value improvement and policy improvement is a pair (π^*, V^*) satisfying

$$V^*(x) = \max_{a \in A} (\lambda \mathbin{⨾} V^*)(x,a) = \max_{a \in A} U(x,a) + \beta V^*(f(x,a))$$
$$\pi^*(x) = \arg\max_{a \in A} (\lambda \mathbin{⨾} V^*)(x,a) = \arg\max_{a \in A} U(x,a) + \beta V^*(f(x,a))$$

Example 5 (Gridworld example)**.** A policy of an agent in our version of Gridworld (Figure 1) is a function from the 4×4 set of states $X = \{1,2,3,4\}^2$ that we index by (i,j) to the four-element set of actions $A = \{\leftarrow, \rightarrow, \uparrow, \downarrow\}$, i.e. an element of A^X. Initializing the value function V with the environments immediate reward whose only non-zero value is $V(0,0) = 1$ (top-left corner) and the policy with a upwards facing constant action $\pi(i,j) = \uparrow$ for all $(i,j) \in X$, a value improvement step would leave the policy unchanged while updating V to $\overline{\pi} \mathbin{⨾} \lambda \mathbin{⨾} V$, which differs with V only at $(0,1) \mapsto \beta$.

If we instead perform a policy improvement step, the value function remains unchanged while the new policy differs with π at $(1,0) \mapsto \arg\max_{a \in A} (\lambda \mathbin{⨾} v)(1,0,a) = \leftarrow$.

4.2 Continuous-space deterministic decision processes

Example 6 (Inverted pendulum)**.** A state of our time-discretized inverted pendulum on a cart consists of $[y, \dot{y}, \theta, \dot{\theta}]^\top$ in the state space $X = \mathbb{R}^4$. The linearized transition function that sends x_k to $x_{k+1} = Ax_k + Ba_k$ is a smooth map $X \to Y$. The discretized cost $J(x,a) = \sum_{k=0}^{\infty} \beta^k C(x(k), a(k))$ defines the backwards smooth function $X \times A \times \mathbb{R} \to \mathbb{R}$ which adds the cost at the kth time step $C(x(k), a(k))$ to the discounted sum:

$$\left(x(k), a(k), \sum_{j=k+1} \beta^j C(x(j), a(j))\right) \mapsto \sum_{j=k} \beta^j C(x(j), a(j))$$

These two maps form an optic $\lambda : \binom{X\times A}{\mathbb{R}} \to \binom{X}{\mathbb{R}}$ in $\mathbf{Optic}_{\mathbf{Euc}}$. Note that the cost function C is itself typically not affine, but rather convex (intuitively, since the 'good states' that should minimise the cost fall in the middle of the state space).

In conclusion, the two optics involved in this example are

$$\lambda = \begin{pmatrix} f \\ J \end{pmatrix} : \begin{pmatrix} X\times A \\ \mathbb{R} \end{pmatrix} \to \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \qquad \overline{\pi} = \begin{pmatrix} \pi \\ p_2 \end{pmatrix} : \begin{pmatrix} X \\ \mathbb{R} \end{pmatrix} \to \begin{pmatrix} X\times A \\ \mathbb{R} \end{pmatrix}$$

$$\begin{aligned} f : X\times A &\to X & \mathrm{gr}(\pi) : X &\to X\times A \\ (x,a) &\mapsto Ax+Ba & x &\mapsto (x,\pi(x)) \\ J : X\times A\times\mathbb{R} &\to \mathbb{R} & p_2 : X\times\mathbb{R} &\to \mathbb{R} \\ (x,a,r) &\mapsto C(x,a)+\beta r & (\_,r) &\mapsto r \end{aligned}$$

This formalisation of the continuous state space misses however a practical problem. Let S be a continuous state space. In numerical implementations, policy improvement over S needs to map an action to every point in the space. Two common approaches are to discretize the state space into a possibly non-uniform grid, or to restrict the space of values to a family of parametrized functions [34, Sec.4.]. The discretization approach treats the continuous state space as a distribution over a simplicial complex X obtained e.g. by triangulation, $\mathbf{Euc}(1,S) \approx \mathbf{Mark}(I,X)$, where a continuous state gets mapped to a distribution over the barycentric coordinates of the simplex. This effectively transforms the initial continuous-space deterministic decision process into a discrete-space MDP, modelling numerical approximation errors as stochastic uncertainty.

4.3 Discrete-space Markov decision processes

Consider a Markov decision process with a discrete set of states X, discrete and finite set of actions A, a transition Markov kernel $f : X\times A \to \Delta(X)$, expected payoff function $U : X\times A \to \mathbb{R}$ and discount factor $\beta \in (0,1)$. We can write the transition function as conditional probabilities $f(x' \mid x,a)$.

We can convert this data into an optic $\lambda : \binom{X\otimes A}{\mathbb{R}} \to \binom{X}{\mathbb{R}}$ in the category $\mathbf{Optic}_{\mathbf{Mark},\mathbf{Conv}}$ given by **Mark** acting on both itself and **Conv**. This optic is given concretely by $(X\otimes A, g, g')$ where $g : X\otimes A \to X\otimes A\otimes X$ in **Mark** is given by $\Delta_{X\otimes A} \mathbin{⨟} (f\otimes \mathrm{id}_{X\otimes A})$, and $g' : \Delta(X\otimes A) \to [\mathbb{R},\mathbb{R}]$ in **Conv** is defined by $g'(\alpha)(r) = \mathbb{E}U(\alpha)+\beta r$, where $\alpha \in \Delta(X\times A)$ is a joint distribution on states and actions. Alternatively, we can note that the domain of g' is free on the set $X\times A$ (although it cannot be considered free on an object of **Mark**), and define it as the linear extension of $g'(x,a)(r) = U(x,a)+\beta r$.

With this setup, value improvement $(\pi,V) \mapsto (\pi, \overline{\pi} \mathbin{⨟} \lambda \mathbin{⨟} V)$ yields the value function

$$V'(x) = \mathbb{E}_{a\sim\pi(x)}[U(x,a)+\beta V(f(x,a))]$$

Alternating steps of value and policy improvement converge to the optimal policy π^* and value function V^*, which maximises the expected value of the policy:

$$V^*_{\pi^*}(x_0) = \mathbb{E}\sum_{k=0}^{\infty} \beta^k U(x_k,\pi^*(x_k))$$

Example 7 (Gridworld, continued)**.** In a proper MDP, transition functions can be stochastic, and update steps have to take expectations over values: value improvement maps $(\pi,V) \mapsto (\pi,\overline{\pi} \mathbin{⨟} \lambda \mathbin{⨟} V)$ and policy improvement maps $(\pi,V) \mapsto (x \mapsto \arg\max_{a\in A} \mathbb{E}(\lambda \mathbin{⨟} V)(x,a), V)$. This model also accepts stochastic policy improvement steps like ε-greedy, which is an ad hoc heuristic technique of balancing exploration and exploitation in reinforcement learning [26, Sec.2], a problem which is known in control theory as the identification-control conflict.

4.4 Continuous-space Markov decision processes

For continuous-space MDPs we need a category of continuous Markov kernels. There are several possibilities for this arising as the kleisli category of a monad, such as the Giry monad on measurable spaces [23], the Radon monad on compact Hausdorff spaces [39] and the Kantorovich monad on complete metric spaces [21]. However, control theorists typically work with more specific parametrised families of distributions for computational reasons, the most common being normal distributions. We will work with the category **Gauss** of Euclidean spaces and affine functions with Gaussian noise [20, section 6]. (This is an example of a Markov category that does not arise as the kleisli category of a monad, because its multiplication map would not be affine.) This works because the pushforward measure of a Gaussian distribution along an affine function is still Gaussian, which fails for more general functions.

Example 8. We will formulate the savings problem with normally-distributed income. The inequality constraints (namely that the balance cannot be negative and that the agent cannot consume more than their current balance) introduce nonlinearities. We can deal with the latter by constraining the optimisation in the policy improvement step, but the former threatens to take us outside the category **Gauss** and we must allow the balance to possibly become negative for the purposes of this example.

Gauss is a Markov category that is not cartesian (the monoidal product is the cartesian product of Euclidean spaces, which adds the dimensions), so it acts on itself by the monoidal product and we take the category $\mathbf{Optic}_{\mathbf{Gauss}}$. We take the state and action spaces to be $X = A = \mathbb{R}$. The transition function $f : \mathbb{R} \otimes \mathbb{R} \to \mathbb{R}$ is given by $f(x,a) = (1+\gamma)x - a + \mathcal{N}(\mu,\sigma)$, and the payoff function $U : \mathbb{R} \otimes \mathbb{R} \to \mathbb{R}$ is given by $U(x,a) = a$.

We modify the policy improvement step to be

$$\textbf{Policy improvement:} \qquad (\pi, V) \mapsto (x \mapsto \arg\max_{a \in A(x)} (\lambda \fatsemi V)(x,a), V)$$

where $A(x)$ is the set $A(x) = \{a \in \mathbb{R} \mid 0 \le a \le x + i\}$. This enforces that the agent cannot consume negative amounts or consume more than their current balance - since the optimisation is done externally to the category **Gauss** we can avoid one source of nonlinearity this way.

5 Q-learning

Consider a deterministic decision process corresponding to the optic $\lambda : \binom{X \times A}{\mathbb{R}} \to \binom{X}{\mathbb{R}}$. The dynamical system with state space $A^X \times \mathbf{Optic}_{\mathbf{Set}}\left(\binom{X \times A}{\mathbb{R}}, I\right)$ has elements (π, q) consisting of a *state-action* value function $q : X \times A \to \mathbb{R}$ as in (4) rather than a state-value function $V : X \to \mathbb{R}$.

We can define similar update steps

$$\textbf{Value improvement:} \qquad (\pi, q) \mapsto (\pi, \lambda \fatsemi \overline{\pi} \fatsemi q)$$

$$\textbf{Policy improvement:} \qquad (\pi, q) \mapsto \left(x \mapsto \arg\max_{a \in A} q(x,a), q\right)$$

These can also be fused into a single step:

$$\textbf{State-action value iteration:} \qquad (\pi, q) \mapsto \left(x \mapsto \arg\max_{a \in A} q(x,a), \lambda \fatsemi \overline{\pi} \fatsemi q\right)$$

Observe that composition of the λ optic with $\overline{\pi}$ is flipped compared to the case seen in Section 4.1, as we want an element of $\mathbf{Optic}_{\mathbf{Set}}\left(\binom{X \times A}{\mathbb{R}}, \binom{X \times A}{\mathbb{R}}\right)$ to compose with q.

The advantage of learning state-action value functions $X \otimes A \to \mathbb{R}$ rather than state-value functions $X \to \mathbb{R}$ is that is gives a way to approximate $\arg\max_{a \in A} (\lambda \fatsemi \overline{\pi} \fatsemi q)(x,a)$ without making any use of λ, namely by instead using $\arg\max_{a \in A} q(x,a)$. This leads to an effective method known as Q-learning for computing optimal control policies even when the MDP is unknown, with only a single state transition and payoff being sampled at each time-step. This is the essential difference between classical control theory and *reinforcement learning*. The above method, despite learning a q-function, is *not* Q-learning because it makes use of λ during value improvement.

Q-learning [41] is a sampling algorithm that approximates the state-action value iteration, usually by a lookup table Q referred as Q-matrix. It treats the optic as a black box, having therefore no access to the transition or rewards functions used in (4), and instead updates q by interacting with the environment dynamics:

$$q'(x',a) = (1-\alpha)q(x,a) + \alpha(r + \beta \max_{a'} q(x',a'))$$

where $\alpha \in (0,1)$ is a weighting parameter. Note that both the new state x' and the reward r are obtained by interacting with the system, rather than looked ahead by $x' = f(x,a)$ and $r = U(x,a)$. It falls in the family of temporal difference algorithms.

6 Further work

At the end of the previous section, it can be seen that Q-learning is no longer essentially using the structure of the category of optics, instead treating the Q-function as a mere function. We believe this can be overcome using the framework of categorical cybernetics [11], leading to a fully optic-based approach to reinforcement learning. By combining with other instantiations of the same framework, it is hoped to encompass the zoo of modern variants of reinforcement learning that have achieved spectacular success in many applications in the last few years. For example, deep Q-learning represents the Q-function not as a matrix but as a deep neural network, trained by gradient descent, allowing much higher dimensionality to be handled in practice. Deep learning is currently one of the main applications of categorical cybernetics [14].

The proof that dynamic programming algorithms converge to the optimal policy and value function typically proceed by noting that the set of all value functions form a complete ordered metric space and that value improvement is a monotone contraction mapping. The metric structure is used to prove that iteration converges to a unique fixpoint by the contraction mapping theorem, and then the order structure is used to prove that this fixpoint is indeed optimal. Since value improvement is optic composition, these facts would be a special case of the category of optics being enriched in the category of ordered metric spaces and monotone contraction mappings. We do not currently know whether such an enrichment is possible. Unlike costates, general optics have nontrivial forwards passes, so there are two possible approaches: either ignore the forwards passes and defining a metric only in terms of the backwards passes, or defining a metric also using the forwards passes, for example using the Kantorovich metric between distributions. This would also be a reasonable place to unify our approach with the coalgebraic approach with metric coinduction [17].

Finally, continuous time MDPs pose a serious challenge to any approach for which categorical composition is sequencing in time, since composition of two morphisms in a category appears to be inherently discrete-time. (Open games are similarly unable to handle dynamic games with continuous time, such as pursuit games.) A plausible approach to this is to associate an endomorphism in a category to every real interval, by treating that interval of time as a single discrete time-step, and then requiring that all morphisms compose together correctly, similar to a sheaf condition. It is hoped that the Bellman-Jacobi-Hamilton equation, a PDE that is the continuous time analogue of the discrete-time Bellman equation, will similarly arise as a fixpoint in this way. Exploring this systematically is important future work.

References

[1] K.J Åström (1965): *Optimal control of Markov processes with incomplete state information*. Journal of Mathematical Analysis and Applications 10(1), pp. 174–205, doi:10.1016/0022-247x(65)90154-x.

[2] John Baez & Jason Erbele (2015): *Categories in control*. Theory and applications of categories 30(24), pp. 836–881. arXiv:1405.6881.

[3] Georgios Bakirtzis, Michail Savvas & Ufuk Topcu (2022): *Categorical semantics of compositional reinforcement learning*. arXiv:2208.13687.

[4] Wolfram Barfuss (2019): *Learning dynamics and decision paradigms in social-ecological dilemmas*. Ph.D. thesis, Humboldt-Universität zu Berlin, doi:10.18452/20127.

[5] Richard Bellman (1957): *Dynamic Programming*. Princeton University Press, doi:10.1515/9781400835386.

[6] Dimitri P. Bertsekas (2007): *Dynamic Programming and Optimal Control, Vol. II*, 3rd edition. Athena Scientific.

[7] Joe Bolt, Jules Hedges & Philipp Zahn (2019): *Bayesian open games*. arXiv:1910.03656. Forthcoming in *Compositionality*.

[8] Nicola Botta, Cezar Ionescu & Edwin C. Brady (2013): *Sequential decision problems, dependently-typed solutions*. In: Joint Proceedings of the MathUI, OpenMath, PLMMS and ThEdu Workshops and Work in Progress at CICM, Bath, UK, CEUR Workshop Proceedings 1010, CEUR-WS.org. Available at http://ceur-ws.org/Vol-1010/paper-06.pdf.

[9] Dylan Braithwaite, Matteo Capucci, Bruno Gavranović, Jules Hedges & Eigil Fjeldgren Rischel (2022): *Fibre optics*. arXiv:2112.11145.

[10] Matteo Capucci & Bruno Gavranović (2022): *Actegories for the working amthematician*. arXiv:2203.16351. ArXiv:2203.16351.

[11] Matteo Capucci, Bruno Gavranović, Jules Hedges & Eigil Fjeldgren Rischel (2022): *Towards foundations of categorical cybernetics*. Electronic Proceedings in Theoretical Computer Science 372, pp. 235–248, doi:10.4204/eptcs.372.17.

[12] Matteo Capucci, Neil Ghani, Jérémy Ledent & Fredrik Nordvall Forsberg (2022): *Translating extensive form games to open games with agency*. Electronic Proceedings in Theoretical Computer Science 372, pp. 221–234, doi:10.4204/eptcs.372.16.

[13] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore & Mario Román (2020): *Profunctor optics: A categorical update*. arXiv:2001.07488.

[14] G.S.H. Cruttwell, Bruno Gavranović, Neil Ghani, Paul Wilson & Fabio Zanasi (2021): *Categorical foundations of gradient-based learning*. arXiv:2103.01931.

[15] Eric V. Denardo (1967): *Contraction Mappings in the Theory Underlying Dynamic Programming*. SIAM Review 9(2), pp. 165–177, doi:10.1137/1009030. Available at https://ui.adsabs.harvard.edu/abs/1967SIAMR...9..165D.

[16] Nicolas Eschenbaum, Filip Mellgren & Philipp Zahn (2022): *Robust algorithmic collusion*. arXiv:2201.00345.

[17] Frank M. V. Feys, Helle Hvid Hansen & Lawrence S. Moss (2018): *Long-Term Values in Markov Decision Processes, (Co)Algebraically*. In Corina Cîrstea, editor: Coalgebraic Methods in Computer Science,

11202, Springer International Publishing, Cham, pp. 78–99, doi:10.1007/978-3-030-00389-0_6. Available at http://link.springer.com/10.1007/978-3-030-00389-0_6. Series Title: Lecture Notes in Computer Science.

[18] Bernard Friedland (1985): *Control Systems Design*. McGraw-Hill Companies.

[19] Tobias Fritz (2009): *Convex spaces I: Definitions and examples*. arXiv:0903.5522.

[20] Tobias Fritz (2020): *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. Advances in Mathematics 370, p. 107239, doi:10.1016/j.aim.2020.107239. arXiv:1908.07021.

[21] Tobias Fritz & Paolo Perrone (2019): *A probability monad as the colimit of spaces of finite samples*. Theory and applications of categories 34(7), pp. 170–220. arXiv:1712.05363.

[22] Neil Ghani, Jules Hedges, Viktor Winschel & Philipp Zahn (2018): *Compositional game theory*. In: Proceedings of Logic in Computer Science (LiCS) 2018, ACM, pp. 472–481, doi:10.1145/3209108.3209165.

[23] Michèlle Giry (1982): *A categorical approach to probability theory*. In: Lecture Notes in Mathematics, Springer Berlin Heidelberg, pp. 68–85, doi:10.1007/bfb0092872.

[24] Jules Hedges (2017): *Coherence for lenses and open games*. arXiv:1704.02230.

[25] Jules Hedges (2018): *Morphisms of open games*. In: Proceedings of MFPS 2018, 341, Elsevier BV, pp. 151–177, doi:10.1016/j.entcs.2018.11.008.

[26] L. P. Kaelbling, M. L. Littman & A. W. Moore (1996): *Reinforcement Learning: A Survey*. Journal of Artificial Intelligence Research 4, pp. 237–285, doi:10.1613/jair.301.

[27] Anders Kock (1971): *Closed categories generated by commutative monads*. Journal of the Australian Mathematical Society 12(4), pp. 405–424, doi:10.1017/s1446788700010272.

[28] Lars Ljungqvist & Thomas Sargent (2004): *Recursive macroeconomic theory*. MIT Press.

[29] Fosco Loregian (2021): *Coend calculus*. Cambridge University Press. arXiv:1501.02503.

[30] Sagar Mody & Thomas Steffen (2015): *Optimal charging of electric vehicles using a stochastic dynamic programming model and price prediction*. SAE International Journal of Passenger Cars - Electronic and Electrical Systems 8(2), pp. 379–393, doi:10.4271/2015-01-0302.

[31] David Jaz Myers (2021): *Double Categories of Open Dynamical Systems (Extended Abstract)*. Electronic Proceedings in Theoretical Computer Science 333, pp. 154–167, doi:10.4204/eptcs.333.11. Available at http://arxiv.org/abs/2005.05956v2.

[32] Martin L. Puterman (2005): *Markov decision processes: discrete stochastic dynamic programming*. Wiley series in probability and statistics, Wiley-Interscience, Hoboken, NJ.

[33] Mitchell Riley (2018): *Categories of Optics*. arXiv:1809.00738.

[34] John Rust (1996): *Numerical dynamic programming in economics*. In: Handbook of Computational Economics, Amsterdam: Elsevier, pp. 619–729, doi:10.1016/s1574-0021(96)01016-7.

[35] David I. Spivak (2020): *Generalized Lens Categories via functors C^op->Cat*. arXiv:1908.02202.

[36] Kirk Stirtz (2015): *Categorical probability theory*. arXiv:1406.6030.

[37] Nancy Stokey, Robert Lucas & Edward Prescott (1989): *Recursive methods in economic dynamics*. Harvard University Press, doi:10.2307/j.ctvjnrt76.

[38] Richard S. Sutton & Andrew G. Barto (2020): *Reinforcement Learning: An Introduction*. MIT Press.

[39] T. Świrszcz (1974): *Monadic functors and convexity*. Bulletin de l'Acaédemie Polonaise des Sciences 22(1).

[40] Pietro Vertechi (2022): *Dependent optics*. arXiv:2204.09547.

[41] Christopher J. C. H. Watkins & Peter Dayan (1992): *Q-learning*. Machine Learning 8(3-4), pp. 279–292, doi:10.1007/bf00992698.