EPTCS 397

Proceedings of the

Sixth International Conference on
Applied Category Theory 2023

University of Maryland, 31 July - 4 August 2023

Edited by: Sam Staton and Christina Vasilakopoulou

Published: 14th December 2023
DOI: 10.4204/EPTCS.397
ISSN: 2075-2180
Open Publishing Association

Table of Contents

Preface

Sam Staton Christina Vasilakopoulou

The Sixth International Conference on Applied Category Theory took place at the University of Maryland on 31 July – 4 August 2023, following the previous meetings at Leiden (2018), Oxford (2019), MIT (2020, fully online), Cambridge (2021) and Strathclyde (2022). It was preceded by the Adjoint School 2023 (24 – 28 July), a collaborative research event in which junior researchers worked under the mentorship of experts. The conference comprised 59 contributed talks, a poster session, an industry showcase session, four tutorial sessions 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://act2023.github.io/.

ACT 2023 was a hybrid event, with physical attendees present in Maryland and other participants taking part online. All talks were streamed to Zoom and with synchronous discussion on Zulip.

Submission to ACT2023 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 12 pages. Only papers submitted to the proceedings track were considered for publication in this volume. In total, there were 81 submissions, of which 59 were accepted for presentation and 18 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 ACT2023 ranged from pure to applied and included contributions in a wide range of disciplines in science and engineering. ACT2023 included talks in linguistics, functional programming, classical mechanics, quantum physics, probability theory, electrical engineering, epidemiology, thermodynamics, engineering, and logic. 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 9 invited talks by practitioners using category theory in private enterprise.

ACT2023 included four tutorials: David Jaz Myers on Lenses, Paolo Perrone on Markov categories, Dorette Pronk on Double categories, and Evan Patterson and Owen Lynch on AlgebraicJulia.

We are grateful to Angeline Aguinaldo, James Fairbanks, Joe Moeller, and Priyaa Varshinee Srinivasan who played various roles in the ACT organization.

ACT2023 was sponsored by AARMS (Atlantic Association for Research in the Mathematical Sciences), Conexus, DeepMind, NIST (National Institute of Standards and Technology), PIMS (Pacific Institute for the Mathematical Sciences), Quantinuum, University of Florida and 20 Squares.

Sam Staton and Christina Vasilakopoulou
Chairs of the ACT 2023 Programme Committee

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. iii–iii, doi:10.4204/EPTCS.397.0

Bicategories of Automata, Automata in Bicategories

Guido Boccali[†]
Università di Torino, Torino, Italy

Andrea Laretto[⊛]
Tallinn University of Technology, Tallinn, Estonia

Fosco Loregian[♣]
Tallinn University of Technology, Tallinn, Estonia[*]

Stefano Luneia[♡]
Università di Bologna, Bologna, Italy

We study *bicategories of* (deterministic) *automata*, drawing from prior work of Katis-Sabadini-Walters, and Di Lavore-Gianola-Román-Sabadini-Sobociński, and linking their bicategories of 'processes' to a bicategory of Mealy machines constructed in 1974 by R. Guitart. We make clear the sense in which Guitart's bicategory retains information about automata, proving that Mealy machines *à la* Guitart identify to *certain* Mealy machines *à la* K-S-W that we call *fugal automata*; there is a biadjunction between fugal automata and the bicategory of K-S-W. Then, we take seriously the motto that a monoidal category is just a one-object bicategory. We define categories of Mealy and Moore machines *inside* a bicategory $\mathbb{B}$; we specialise this to various choices of $\mathbb{B}$, like categories, relations, and profunctors. Interestingly enough, this approach gives a way to interpret the universal property of reachability as a Kan extension and leads to a new notion of 1- and 2-cell between Mealy and Moore automata, that we call *intertwiners*, related to the universal property of K-S-W bicategory.

1 Introduction

The profound connection between category theory and automata theory is easily explained: one of the founders of the first wrote extensively about the second [23, 24]. A more intrinsic reason is that category theory is a theory of *systems* and *processes*. Morphisms in a category can be considered a powerful abstraction of 'sequential operations' performed on a domain/input to obtain a codomain/output. Hence the introduction of categorical models for computational machines has been rich in results, starting from the elegant attempts by Arbib and Manes [2, 7, 5, 6, 8, 59] –cf. also [3, 20, 22] for exhaustive monographs– and Goguen [28, 29, 30], up to the ultra-formal –and sadly, under-appreciated– experimentations of [9, 10, 32, 33, 35] using hyperdoctrines, 2-dimensional monads, bicategories, lax co/limits... up to the modern coalgebraic perspective of [38, 62, 63, 67]; all this, without mentioning categorical approaches to Petri nets [54], based essentially on the same analogy, where the computation of a machine is *concurrent* –as opposed to single-threaded.

Furthermore, many constructions of computational significance often, if not always, have a mathematical counterpart in terms of categorical notions: the transition from a deterministic machine to a non-deterministic one is reflected in the passage from automata in a monoidal category (cf. [22, 55]), to automata in the Kleisli category of an opmonoidal monad (cf. [34, 40]; this approach is particularly useful to capture categorically *stochastic* automata, [19, 7, 15] as they appear as automata in the Kleisli category of a probability distribution monad); *minimisation* can be understood in terms of factorisation systems (cf. [18, 30]); behaviour as an adjunction (cf. [56, 57]).

The present work starts from the intuition, first presented in [45, 60], that the analogy between morphisms and sequential machines holds up to the point that the series and parallel composition of automata should itself be reflected in the 'series' and 'parallel' composition of morphisms in a category. As a byproduct of the 'Circ' construction in *op. cit.*, one can see how the 1-cells of a certain monoidal bicategory specialise exactly a *Mealy machines* $E \xleftarrow{d} E \otimes I \xrightarrow{s} O$ with inputs and outputs I and O.

[*]Loregian was supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001).

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 1–19, doi:10.4204/EPTCS.397.1

Outline of the paper. The first result we present in section 2 is that this category relates to *another* bicategory constructed by R. Guitart in [32]. Guitart observes that one can use certain categories $\mathsf{Mac}(\mathcal{M},\mathcal{N})$ of spans as hom categories of a bicategory Mac, and shows that Mac admits a concise description as the Kleisli bicategory of the *monad of diagrams* [32, §1] (cf. also [35], by the same author, and [58] for a more modern survey); Mealy machines shall be recognisable as the 1-cells of Mac between monoids, regarded as categories with a single object. The fundamental assumption in [32] is that a Mealy machine $E \xleftarrow{d} E \otimes M \xrightarrow{s} N$ satisfies a certain property of compatibility with the action of d on E, cf. (2.8), that we call being a *fugal* automaton:

$$s(e, m \cdot m') = s(e, m) \cdot s(d(e, m), m').$$

This notion can be motivated in the following way: if s satisfies the above equation, then it lifts to a functor $\mathcal{E}[d] \to N$ defined on the category of elements of the action d, and in fact, defines a 'relational action' in its own right, compatible with d (formally speaking, $\mathcal{E}[d]$ is a *displayed category* [4] over N). We show that there is a sub-bicategory $\mathsf{Mly}^{\flat}_{\mathsf{Set}}$ of $\mathsf{Mly}_{\mathsf{Set}}$ made of fugal automata and that $\mathsf{Mly}^{\flat}_{\mathsf{Set}}$ is biequivalent (actually, strictly) to the 1-full and 2-full sub-bicategory of Mac spanned by monoids.

The second result we propose in this paper is motivated by the motto for which a monoidal category is just a bicategory with a single object: what are automata *inside a bicategory* $\mathbb{B}$ *with more than one object*, where instead of input/output *objects* I, O we have input/output 1-cells, arranged as $e \overset{\delta}{\Leftarrow} e \circ i \overset{\sigma}{\Rightarrow} o$? Far from being merely formal speculation (a similar idea was studied in a short, cryptic note [10] to describe behaviour through Kan extensions: we take it seriously and present it as a quite straightforward observation in Remark 3.6), we show how this allows for a concise generalisation of 'monoidal' machines.

Related work. A word on related work and how we fit into it: the ideas in section 2 borrow heavily from [45, 60] where bicategories of automata (or 'processes') are studied in fine detail; in section 2 we carry on a comparison with a different approach to bicategories of automata, present in [32] but also in [33, 35]; in particular, our proof that there is an adjunction between the two bicategories is novel –to the best of our knowledge– and it hints at the fact that the two approaches are far from being independent. At the level of an informal remark, the idea of approaching automata via (spans where one leg is a) fibrations bears some resemblance to Walters' work on context-free languages through displayed categories in [69], and the requirement to have a fibration as one leg of the span should be thought as mirroring *determinism* of the involved automata: if $\langle s, d\rangle : E \times M \to N \times E$ is fugal and s defines a *fibration* over N, then E is a M-N-bimodule, not only an M-set; there is extensive work of Betti-Kasangian [12, 11, 42] and Kasangian-Rosebrugh [43] on 'profunctorial' models for automata, their behaviour, and the universal property enjoyed by their minimisation: spans of two-sided fibrations [64, 65] and profunctors are well-known to be equivalent ways to present the same bicategory of two-sided fibrations. Carrying on our study will surely determine a connection between the two approaches.

For what concerns section 3, the idea of valuing a Mealy or a Moore machine in a bicategory seems to be novel, although in light of [60] and in particular of their concrete description of $\mathcal{C} = \Omega\Sigma(\mathcal{K}, \otimes)$ it seems that both $\mathsf{Mly}_{\mathbb{B}}$ and $\mathsf{Mre}_{\mathbb{B}}$ allow defining tautological functors into $\mathcal{C}$. How these two bicategories relate is a problem we leave for future investigation: [60] proves that when $\mathcal{K}$ is Cartesian monoidal, $\mathsf{Mly}_{\mathcal{K}}$ is $\Omega\Sigma(\mathcal{K}, \times)$. The conjecture is that our $\mathsf{Mly}_{\mathbb{B}}$ is $\Omega\mathbb{B}$ under some assumptions on the bicategory $\mathbb{B}$: our notion of intertwiner seems to hint in that direction. Characterising 'behaviour as a Kan extension' is nothing but taking seriously the claim that animates applications of coalgebra theory [39, 40] to automata; the –apparently almost unknown– work of Bainbridge [10] bears some resemblance to our idea, but his note is merely sketched, no plausibility for his intuition is given. Nevertheless, we recognise the potential of his idea and took it to its natural continuation with modern tools of 2-dimensional algebra.

1.1 Mealy and Moore automata

The scope of the following subsection is to introduce the main characters studied in the paper:[1] categories of automata valued in a monoidal category $(\mathcal{K},\otimes)$ (in two flavours: 'Mealy' machines, where one considers spans $E \leftarrow E\otimes I \to O$, and 'Moore', where instead one consider pairs $E \leftarrow E\otimes I, E \to O$.

The only purpose of this short section is to fix the notation for section 2 and 3; comprehensive classical references for this material are [3, 22].

For the entire subsection, we fix a monoidal category $(\mathcal{K},\otimes,1)$.

Definition 1.1 (Mealy machine)**.** A *Mealy machine* in $\mathcal{K}$ of input object I and output object O consists of a triple (E,d,s) where E is an object of $\mathcal{K}$ and d,s are morphisms in a span $\mathfrak{e} := \Big(E \xleftarrow{d} E\otimes I \xrightarrow{s} O \Big)$.

Remark 1.2 (The category of Mealy machines)**.** Mealy machines of fixed input and output I,O form a category, if we define a *morphism of Mealy machines* $f : \mathfrak{e} = (E,d,s) \to (F,d',s') = \mathfrak{f}$ as a morphism $f : E \to F$ in $\mathcal{K}$ such that

- $d' \circ (f\otimes I) = f \circ d$;
- $s' \circ (f\otimes I) = s$.

Composition and identities are performed in $\mathcal{K}$.

The category of Mealy machines of input and output I,O is denoted as $\mathsf{Mly}_{\mathcal{K}}(I,O)$.

Definition 1.3 (Moore machine)**.** A *Moore machine* in $\mathcal{K}$ of input object I and output object O is a diagram $\mathfrak{m} := \Big(E \xleftarrow{d} E\otimes I \ ;\ E \xrightarrow{s} O \Big)$.

Remark 1.4 (The category of Moore machines)**.** Moore machines of fixed input and output I,O form a category, if we define a *morphism of Moore machines* $f : \mathfrak{e} = (E,d,s) \to (F,d',s') = \mathfrak{f}$ as a morphism $f : E \to F$ in $\mathcal{K}$ such that

- $d' \circ (f\otimes I) = f \circ d$;
- $s' \circ f = s$.

Remark 1.5 (Canonical extension of a machine)**.** If $(\mathcal{K},\otimes)$ has countable coproducts preserved by each $A\otimes \_$ then the span Definition 1.1, considering for example Mealy machines, can be 'extended' to a span

$$E \xleftarrow{d^*} E\otimes I^* \xrightarrow{s^*} O \tag{1.1}$$

where d^*, s^* can be defined inductively from components $d_n, s_n : E\otimes I^{\otimes n} \to E, O$; if $\mathcal{K}$ is closed, the map d^* corresponds, under the monoidal closed adjunction, to the monoid homomorphism $I^* \to [E,E]$ induced by the universal property of $I^* = \sum_{n\geq 0} I^{\otimes n}$.

2 Bicategories of automata

Let $(\mathcal{K},\times)$ be a Cartesian category. There is a bicategory $\mathsf{Mly}_{\mathcal{K}}$ defined as follows (cf. [60] where this is called 'Circ' and studied more generally, in case the base category has a non-Cartesian monoidal structure):

Definition 2.1 (The bicategory $\mathbf{Mly}_{\mathcal{K}}$, [60])**.** The bicategory $\mathsf{Mly}_{\mathcal{K}}$ has

- its *0-cells* $I,O,U,\dots$ are the same objects of $\mathcal{K}$;

[1] An almost identical introductory short section appears in [13], of which the present note is a parallel submission –although related, the two manuscripts are essentially independent, and the purpose of this repetition is the desire for self-containment.

- its *1-cells* $I \to O$ are the Mealy machines (E,d,s), i.e. the objects of the category $\mathsf{Mly}_{\mathcal{K}}(I,O)$ in Remark 1.2, thought as morphisms $\langle s,d\rangle : E \times I \to O \times E$ in $\mathcal{K}$;
- its *2-cells* are Mealy machine morphisms as in Remark 1.2;
- the composition of 1-cells $\_ \fatsemi \_$ is defined as follows: given 1-cells $\langle s,d\rangle : E \times I \to J \times E$ and $\langle s',d'\rangle : F \times J \to K \times F$ their composition is the 1-cell $\langle s', d' \fatsemi d\rangle : (F \times E) \times I \to K \times (F \times E)$, obtained as
$$F \times E \times I \xrightarrow{F \times \langle s,d\rangle} F \times J \times E \xrightarrow{\langle s',d'\rangle \times E} K \times F \times E; \tag{2.1}$$
- the *vertical* composition of 2-cells is the composition of Mealy machine morphisms $f : E \to F$ as in Remark 1.2;
- the *horizontal* composition of 2-cells is the operation defined thanks to bifunctoriality of $\_ \fatsemi \_ : \mathsf{Mly}_{\mathcal{K}}(B,C) \times \mathsf{Mly}_{\mathcal{K}}(A,B) \to \mathsf{Mly}_{\mathcal{K}}(A,C)$;
- the associator and the unitors are inherited from the monoidal structure of $\mathcal{K}$.

Remark 2.2. Spelled out explicitly, the composition of 1-cells in Equation 2.1 corresponds to the following morphisms (where we freely employ λ-notation available in any Cartesian closed category):

$$d_2 \fatsemi d_1 : \lambda efa.\langle d_2(f, s_1(e,a)), d_1(e,a)\rangle \qquad s_{21} : \lambda efa.s_2(f, s_1(e,a)) \tag{2.2}$$

Remark 2.3 (Kleisli extension of automata as base changes)**.** If $P : \mathcal{K} \to \mathcal{K}$ is a commutative monad [48, 49], we can lift the monoidal structure $(\mathcal{K}, \otimes)$ to a monoidal structure $(\mathsf{Kl}(P), \bar{\otimes})$ on the Kleisli category of P; this leads to the notion of *P-non-deterministic automata* or *P_λ-machines* studied in [34, §2, Définition 6]. Nondeterminism through the passage to a Kleisli category is a potent idea that developed into the line of research on automata theory through coalgebra theory [40], cf. in particular Chapter 2.3 for a comprehensive reference, or the self-contained [38].

We do not investigate the theory of P_λ-machines apart from the following two results the proof of which is completely straightforward: we content ourselves with observing that the results expounded in [44, 60], and in general the language of bicategories of processes, naturally lends itself to the generation of *base-change functors*, of which the following two are particular examples.

Proposition 2.4. The correspondence defined at the level of objects by sending $(E,d,s) \in \mathsf{Mly}_{\mathcal{K}}(I,O)$ to

$$PE \xleftarrow{\eta_E} E \xleftarrow{d} E \otimes I \xrightarrow{s} O \xrightarrow{\eta_O} PO \tag{2.3}$$

extends to a functor $L : \mathsf{Mly}_{\mathcal{K}}(I,O) \to \mathsf{Mly}_{\mathsf{Kl}(P)}(I,O)$.

Proposition 2.5. The correspondence sending $(E,d,s) \in \mathsf{Mly}_{\mathsf{Kl}(P)}(I,O)$ into

$$PE \xleftarrow{\mu_E} PPE \xleftarrow{Pd \circ D} PE \otimes PI \xrightarrow{Ps \circ D} PPO \xrightarrow{\mu_O} PO \tag{2.4}$$

extends to a functor $(-)^{\mathrm{e}} : \mathsf{Mly}_{\mathsf{Kl}(P)}(I,O) \to \mathsf{Mly}_{\mathcal{K}}(PI, PO)$.

More precisely, the proof of the following result is straightforward –only slightly convoluted in terms of notational burden– so much so that we feel content to enclose it in a remark.

Remark 2.6. Let $\mathcal{H}, \mathcal{K}$ be cartesian monoidal categories, then we can define 2-categories $\mathsf{Mly}_{\mathcal{H}}, \mathsf{Mly}_{\mathcal{K}}$ as in Definition 2.1; let $F : \mathcal{H} \to \mathcal{K}$ be a lax monoidal functor. Then, there exists a 'base change' pseudofunctor $F_* : \mathsf{Mly}_{\mathcal{H}} \to \mathsf{Mly}_{\mathcal{K}}$, which is the 1-cell part of a 2-functor $\mathsf{Cat}_\times \to \mathsf{Bicat}$ defined on objects as $\mathcal{K} \mapsto \mathsf{Mly}_{\mathcal{K}}$, from (Cartesian monoidal categories, product-preserving functors, Cartesian natural transformations), to (bicategories, pseudofunctors, oplax natural transformations).

As a corollary, we re-obtain the functors of Proposition 2.5 and Proposition 2.4 from the free and forgetful functors $F_P : \mathcal{K} \to \mathsf{Kl}(P)$ and $U_P : \mathsf{Kl}(P) \to \mathcal{K}$.

2.1 Fugal automata, Guitart machines

A conceptual construction for $\mathsf{Mly}_{\mathcal{K}}$ in Definition 2.1 is given as follows in [44]: it is the category $\Omega\Sigma(\mathcal{K},\otimes)$ of pseudofunctors $\mathbb{N} \to \Sigma(\mathcal{K},\times)$ and lax transformations, where Σ is the 'suspension' of $(\mathcal{K},\otimes)$, i.e. $\mathcal{K}$ regarded as a one-object bicategory; a universal property for $\mathsf{Mly}_{\mathcal{K}}$ is provided in [45] (actually, for any $\Omega\Sigma(\mathcal{K},\otimes)$): it is the free *category with feedbacks* (op. cit., Proposition 2.6, see also [51]) on $\mathcal{K}$. The bicategory $\mathsf{Mly}_{\mathcal{K}}$ addresses the fundamental question of whether one can fruitfully consider morphisms in a category as an abstraction of 'sequential operations' performed on a domain/input to obtain a codomain/output, and up to what point the analogy between morphisms and sequential machines holds up (composing 1-cells in $\mathsf{Mly}_{\mathcal{K}}$ accounts for the sequential composition of state machines, where the state E is an intrinsic part of the specification of a machine/1-cell $\langle s,d\rangle$).

Twenty eight years before [45], however, René Guitart [32] exhibited another bicategory Mac of 'Mealy machines', defined as a suitable category of spans, of which one leg is a fibration, and its universal property: Mac is the Kleisli bicategory of the diagram monad (*monade des diagrammes* in [32, §1], cf. [47, 58]) $\mathsf{Cat}/\!/\_$.[2]

Definition 2.7 (The bicategory $\mathbf{Mac}^{\mathsf{s}}$, adapting [32])**.** Define a bicategory $\mathsf{Mac}^{\mathsf{s}}$ as follows:

- 0-cells are categories $\mathcal{A},\mathcal{B},\mathcal{C}\dots$;
- 1-cells $(\mathcal{E};p,S) : \mathcal{A} \to \mathcal{B}$ consist of spans
$$\mathcal{A} \xleftarrow{\;p\;} \mathcal{E} \xrightarrow{\;S\;} \mathcal{B} \tag{2.5}$$
where $p : \mathcal{E} \to \mathcal{A}$ is a discrete opfibration;
- 2-cells $H : (\mathcal{E};p,S) \Rightarrow (\mathcal{F};q,T)$ are pairs where $H : \mathcal{E} \to \mathcal{F}$ is a morphism of opfibrations (cf. [37, dual of 1.7.3.(i)]): depicted graphically, a 2-cell is a diagram
$$\begin{tikzcd} \mathcal{E} \arrow[r, "S"] \arrow[d, "p"'] \arrow[dr, "H"'] & \mathcal{B} \\ \mathcal{A} & \mathcal{F} \arrow[l, two heads, "q"] \arrow[u, "T"'] \end{tikzcd} \tag{2.6}$$
where both triangles commute and H is an opCartesian functor (it preserves opCartesian morphisms);
- composition of 1-cells $\mathcal{A} \xleftarrow{p} \mathcal{E} \xrightarrow{S} \mathcal{B}$ and $\mathcal{B} \xleftarrow{q} \mathcal{F} \xrightarrow{T} \mathcal{C}$ is via pullbacks, as it happens in spans, and all the rest of the structure is defined as in spans.

Given this, a natural question that might arise is how do the two bicategories of Definition 2.1 and Definition 2.7 interact, if at all?

In the present section, we aim to prove the existence of an adjunction (cf. Theorem 2.18) between a suitable sub-bicategory of $\mathsf{Mac}^{\mathsf{s}}$ and a sub-bicategory of $\mathsf{Mly}_{\mathsf{Set}}$ spanned over what we call *fugal* Mealy

[2] Guitart's note [32] is rather obscure with respect of the fine details of his definition, as he chooses for 2-cells the H for which the upper triangle in (2.6) is only *laxly* commutative, and when it comes to composition of 1-cells he invokes a *produit fibré canonique*; apparently, this can't be interpreted as a strict pullback, or there would be no way to define horizontal composition of 2-cells; using a comma object instead of a strict pullback, the lax structure is given by the universal property –observe that the functor that must be an opfibration is indeed an opfibration, thanks to [37, Exercise 1.4.6], but this opfibration does not remember much of the opfibration q one pulled back. Our theorem involves a strict version of Guitart's Mac, because the functor Π of Theorem 2.17 factors through $\mathsf{Mac}^{\mathsf{s}} \subseteq \mathsf{Mac}$.

machines between monoids (cf. Definition 2.11).[3]

Since the construction of $\mathsf{Mac}^{\mathsf{s}}$ outlined in [32] requires some intermediate steps (and it is written in French), we deem it necessary to delve into the details of how its structure is presented. To fix ideas, we keep working in the category of sets and functions.

Notation 2.8. In order to avoid notational clutter, we will blur the distinction between a monoid M and the one-object category it represents; also, given the d part of a Mealy machine, we will denote as d^* both the extension $E \times I^* \to E$ of Remark 1.5, which is a monoid action of I^* on E, and the functor $I^* \to \mathsf{Set}$ to which the action corresponds.

Remark 2.9. In the notation above, a Mealy machine $\mathfrak{e} = (E, d, s)$ yields a discrete opfibration (cf. [1, 37]) $\mathcal{E}[d^*] \to I^*$ over the monoid I^*, and $\mathcal{E}[a]$ is the *translation category* of an M-set $a : M \times X \to X$ (cf. [14] for the case when M is a group: clearly, $\mathcal{E}[a]$ is the category of elements of the action $a : M \to \mathsf{Set}$ regarded as a functor), i.e. the category having

- objects the elements of E;
- a morphism $m : e \to e'$ whenever $e' = d^*(e, m)$.

Composition and identities are induced by the fact that d^* is an action.

Remark 2.10. The hom-categories $\mathsf{Mac}^{\mathsf{s}}(\mathcal{A}, \mathcal{B})$ of Definition 2.7 fit into strict pullbacks

$$\begin{tikzcd} \mathsf{Mac}^{\mathsf{s}}(\mathcal{A}, \mathcal{B}) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner"] & \mathsf{Cat}/\mathcal{B} \arrow[d] \\ \mathsf{opFib}/\mathcal{A} \arrow[r] & \mathsf{Cat} \end{tikzcd} \tag{2.7}$$

where $\mathsf{Cat}/\mathcal{B}$ is the usual slice category of Cat over $\mathcal{B}$.

Definition 2.11 (Fugal automaton)**.** Let M, N be monoids; a Mealy machine $\langle s, d \rangle : E \times M \to N \times E$ is *fugal* if its s part satisfies the equation

$$s(e, m \cdot m') = s(e, m) \cdot s(d(e, m), m'). \tag{2.8}$$

Remark 2.12. This definition appears in [32, §2] and it looks an ad-hoc restriction for what an output map in a Mealy machine shall be; but (2.8) can be motivated in two ways:

- A fugal Mealy machine $\langle s, d \rangle : E \times M \to N \times E$ induces in a natural way a functor $\Sigma : \mathcal{E}[d^*] \to N$ because (2.8) is exactly equivalent to the fact that Σ defined on objects in the only possible way, and on morphisms as $\Sigma(e \to d^*(e, m)) = s(e, m)$ preserves (identities and) composition;
- given a generic Mealy machine $\langle s, d \rangle : E \times A \to B \times E$ one can produce a 'universal' fugal Mealy machine $\langle s, d \rangle^\flat = \langle s^\flat, \_ \rangle : E \times A^* \to B^* \times E$, and this construction is well-behaved for 1-cell composition in $\mathsf{Mly}_{\mathsf{Set}}$, in the sense that $(s_{21})^\flat = s_{21}^{\flat\flat}$.

The remainder of this section is devoted to making these claims precise (and prove them). In particular, the 'universality' of $\langle s, d \rangle^\flat$ among fugal Mealy machines obtained from $\langle s, d \rangle$ is clarified by the following Lemma 2.13 and by Theorem 2.18, where we prove that there is a 2-adjunction between $\mathsf{Mly}_{\mathsf{Set}}$ and $\mathsf{Mly}^\flat_{\mathsf{Set}}$.

Lemma 2.13. Given sets A, B, denote with A^*, B^* their free monoids; then, there exists a 'fugal extension' functor $(\_)^\flat_{A,B} : \mathsf{Mly}_{\mathsf{Set}}(A, B) \to \mathsf{Mly}^\flat_{\mathsf{Set}}(A^*, B^*)$.

[3] A *fugue* is 'a musical composition in which one or two themes are repeated or imitated by successively entering voices and contrapuntally developed in a continuous interweaving of the voice parts', cf. [68]. In our case, the interweaving is between s, d in a Mealy machine.

Proof. The proof is deferred to the appendix, p. 16. In particular, the map $s^\flat$ is constructed inductively as

$$\begin{cases} s^\flat(e,[\,]) & = [\,] \\ s^\flat(e,a::as) & = s(e,a)::s^\flat(d(e,a),as) \end{cases} \tag{2.9}$$

and it fits in the Mealy machine $\langle s^\flat, d^*\rangle : E\times A^* \to B^*\times E$ where d^* is as in (1.1). The proof that $\langle s^\flat, d^*\rangle$ is fugal in the sense of (2.8) can be done by induction and poses no particular difficulty. □

Lemma 2.14. Given sets A,B there exists a commutative square

$$\begin{tikzcd} \mathsf{Mly}^\flat_{\mathsf{Set}}(A^*,B^*) \arrow[r] \arrow[d] & \mathsf{Cat}/B^* \arrow[d] \\ \mathsf{opFib}/A^* \arrow[r] & \mathsf{Cat}. \end{tikzcd} \tag{2.10}$$

Proof of Lemma 2.14. Given a fugal Mealy machine $\langle s,d\rangle : E\times A^* \to B^*\times E$ between free monoids, from the action d we obtain a discrete opfibration $\mathcal{E}[d] \to A^*$, and from the map $s : E\times A^* \to B^*$ we obtain a functor $\Sigma : \mathcal{E}[d^*] \to B^*$ as in Remark 2.12. So, one can obtain a span

$$A^* \xleftarrow{D} \mathcal{E}[d^*] \xrightarrow{\Sigma} B^* \tag{2.11}$$

where the leg $D : \mathcal{E}[d^*] \to A^*$ is as in Remark 2.9 and Σ is an in Remark 2.12. The functors $\mathsf{opFib}/A^* \leftarrow \mathsf{Mly}^\flat_{\mathsf{Set}}(A^*,B^*) \to \mathsf{Cat}/B^*$ project to each of the two legs. □

Corollary 2.15. The universal property of the hom-categories $\mathsf{Mac}^{\mathsf{s}}(\mathcal{A},\mathcal{B})$ exposed in Remark 2.10 yields the right-most functor in the composition

$$\Gamma_{A,B} : \mathsf{Mly}_{\mathsf{Set}}(A,B) \xrightarrow{(\_)^\flat_{A,B}} \mathsf{Mly}^\flat_{\mathsf{Set}}(A^*,B^*) \xrightarrow{\Pi_{A,B}} \mathsf{Mac}^{\mathsf{s}}(A^*,B^*) \tag{2.12}$$

Lemma 2.16 (Fugal extension preserves composition)**.** Let A,B,C be sets, $s_1 : E\times A \to B$ and $s_2 : F\times B \to C$ parts of Mealy machines $\langle s_1,\_\rangle$ and $\langle s_2,\_\rangle$; then $(s_{21})^\flat = s^{\flat\flat}_{21}$.

Proof. The proof is deferred to the appendix, p. 16.[4] □

This, together with the fact that the identity 1-cell $1\times A \to A\times 1$ is fugal (the proof is straightforward), yields that there exists a 2-subcategory $\mathsf{Mly}^\flat_{\mathsf{Set}}$ of $\mathsf{Mly}_{\mathsf{Set}}$ where 0-cells are monoids, 1-cells are the $\langle s,d\rangle$ where s is fugal in the sense of Definition 2.11, and we take all 2-cells.

Theorem 2.17. The maps $\Gamma_{A,B}$ of Corollary 2.15 constitute the action on 1-cells of a 2-functor $\Gamma : \mathsf{Mly}_{\mathsf{Set}} \to \mathsf{Mac}^{\mathsf{s}}$. More precisely, there are 2-functors $(\_)^\flat : \mathsf{Mly}_{\mathsf{Set}} \to \mathsf{Mly}^\flat_{\mathsf{Set}}$ and $\Pi : \mathsf{Mly}^\flat_{\mathsf{Set}} \to \mathsf{Mac}^{\mathsf{s}}$ whose composition is Γ.

Proof. The proof is deferred to the appendix, p. 17. □

Theorem 2.18. The 2-functor $(\_)^\flat : \mathsf{Mly}_{\mathsf{Set}} \to \mathsf{Mly}^\flat_{\mathsf{Set}}$ admits a right 2-adjoint; the 2-functor $\Pi : \mathsf{Mly}^\flat_{\mathsf{Set}} \to \mathsf{Mac}^{\mathsf{s}}$ identifies $\mathsf{Mly}^\flat_{\mathsf{Set}}$ as the 1-full and 2-full subcategory of $\mathsf{Mac}^{\mathsf{s}}$ spanned by monoids.

[4]The argument is straightforward but tedious (the difficult part is that the condition to verify on $(s_{21})^\flat$ involves $d_2 \mathfrak{z} d_1$, the expression of which we recall from (2.2), is the λ-term $\lambda efa.\langle d_2(f,s_1(e,a)),d_1(e,a)\rangle$).

Proof. The proof is deferred to the appendix, p. 17. The last statement essentially follows from (2.11): the span (D,Σ) is essentially equivalent to the fugal Mealy machine $\langle s,d\rangle$, since its left leg D determines a unique action of A^* on the set of objects $\mathcal{E}[d^*]_0$, and Σ and s are mutually defined. ◻

3 Bicategory-valued machines

A monoidal category is just a bicategory with a single 0-cell; then, do Definition 1.1 and Definition 1.3 admit a generalisation when instead of $\mathcal{K}$ we consider a bicategory $\mathbb{B}$ with more than one object? The present section answers in the positive. We also outline how, passing to automata valued in a bicategory, a seemingly undiscovered way to define morphisms between automata, different (from (1.2) and) from the categories of 'variable' automata described in [22, §11.1]: we study this notion in Definition 3.12.

In our setting, 'automata' become diagrams of *2-cells* in $\mathbb{B}$, between input, output and state *1-cells*, in contrast with previous studies where automata appeared as objects, and with [60] (and our section 2), where they appear as diagrams of 1-cells between input, output and state 0-cells. This perspective suggests that 2-dimensional diagrams of a certain shape can be thought of as state machines -so, they carry a computational meaning; but also that state machines can be fruitfully interpreted as diagrams: in Example 3.11 we explore definitions of an automaton where input and output are *relations*, or functors (in Example 3.9), or profunctors (in Example 3.10); universal objects that can be attached to the 2-dimensional diagram then admit a computational interpretation (cf. (3.9) where a certain Kan extension resembles a 'reachability' relation).

This idea is not entirely new: it resembles an approach contained in [10, 9] where the author models the state space of abstract machines as a functor, of which one can take the left/right Kan extension along an 'input scheme'. However, Bainbridge's works are rather obscure (and quite ahead of their time), so we believe we provide some advancement to state of the art by taking his idea seriously and carrying to its natural development –while at the same time, providing concrete examples of bicategories in which inputs/outputs automata can be thought of as 1-cells, and investigating the structure of the class of all such automata as a global object.

Definition 3.1. Adapting from Definition 1.1 *verbatim*, if $\mathbb{B}$ is a bicategory with 0-cells $A,B,X,Y,\dots$, 1-cells $i : A \to B, o : X \to Y,\dots$ and 2-cells $\alpha,\beta,\dots$ the kind of object we want in $\mathsf{Mly}_{\mathbb{B}}(i,o)$ is a span of the following form:

$$e \overset{\delta}{\Longleftarrow} e\circ i \overset{\sigma}{\Longrightarrow} o \tag{3.1}$$

for 1-cells $i : X \to Y$, $e : A \to B$, $o : C \to D$. Note that with $\_\circ\_$, we denote the composition of 1-cells in $\mathbb{B}$, which becomes a monoidal product in $\mathbb{B}$ has a single 0-cell.

Remark 3.2. The important observation here is that the mere existence of the span (δ,σ) 'forces the types' of i,o,e in such a way that i *must* be an endomorphism of an object $A \in \mathbb{B}$, and $e,o : A \to B$ are 1-cells. Interestingly, these minimal assumptions required even to consider an object like (3.1) make iterated compositions $i\circ\cdots\circ i$ as meaningful as iterated tensors $I\otimes\cdots\otimes I$, and in fact, the two concepts coincide when $\mathbb{B}$ has a single object $*$ and hom-category $\mathbb{B}(*,*) = \mathcal{K}$.

In the monoidal case, the fact that an input 1-cell stands on a different level from an output was completely obscured by the fact that *every* 1-cell is an endomorphism.

Let us turn this discussion into a precise definition.

Definition 3.3 (Bicategory-valued Mealy machines)**.** Let $\mathbb{B}$ be a bicategory, and fix two 1-cells $i : A \to A$ and $o : A \to B$; define a category $\mathsf{Mly}_{\mathbb{B}}(i,o)$ as follows:

BML1) the objects are diagrams of 2-cells as in (3.1);

BML2) the morphisms $(e,\delta,\sigma) \to (e',\delta',\sigma')$ are 2-cells $\varphi : e \Rightarrow e'$ subject to conditions similar to Remark 1.2:

- $\sigma' \circ (\varphi * i) = \sigma$;
- $\delta' \circ (\varphi * i) = \varphi \circ \delta$.

Definition 3.4 (Bicategory-valued Moore machines)**.** Define a category $\mathsf{Mre}_{\mathbb{B}}(i,o)$ as follows:

BMO1) the objects are pairs of 2-cells in $\mathbb{B}$, $\delta : e \circ i \Rightarrow e$ and $\sigma : e \Rightarrow o$;

BMO2) the morphisms $(e,\delta,\sigma) \to (e',\delta',\sigma')$ are 2-cells $\varphi : e \Rightarrow e'$ such that diagrams of 2-cells similar to those in Definition 3.3 are commutative.

Notation 3.5. In the following, an object of $\mathsf{Mly}_{\mathbb{B}}(i,o)$ (resp., $\mathsf{Mre}_{\mathbb{B}}(i,o)$) will be termed a *bicategorical Mealy machine* (resp., a *bicategorical Moore machine*) of input cell i and output cell o, and the objects A,B are the *base* of the bicategorical Mealy machine (e,δ,σ). To denote that a bicategorical Mealy machine is based on A,B we write $(e,\delta,\sigma)_{A,B}$.

In [10] the author models the state space of abstract machines as follows: fix categories A,X,E and a functor $\Phi : X \to A$, of which one can take the left/right Kan extension along an 'input scheme' $u : E \to X$; a *machine with input scheme u* is a diagram of 2-cells in $\mathsf{Cat}(E,A)$ of the form $\mathcal{M} = (I \Rightarrow \Phi \circ u \Rightarrow J)$, and the *behaviour $B(\mathcal{M})$* of $\mathcal{M}$ is the diagram of 2-cells $\mathrm{Lan}_u I \Rightarrow \Phi \Rightarrow \mathrm{Ran}_u J$.

All this bears some resemblance to the following remark, but at the same time looks very mysterious, and not much intuition is given in *op. cit.* for what the approach in study means; we believe our development starts from a similar point (the intuition that a category of machines is, in the end, some category of diagrams –a claim we substantiate in Proposition 3.8) but rapidly takes a different turn (cf. Definition 3.12), and ultimately gives a cleaner account of Bainbridge's perspective (see also [9] of the same author).

Remark 3.6 (Behaviour as a Kan extension)**.** A more convenient depiction of the span in BMO1 will shed light on our Definition 3.3 and 3.4, giving in passing a conceptual motivation for the convoluted shape of finite products in $\mathsf{Mre}_{\mathcal{K}}(I,O)$ and $\mathsf{Mly}_{\mathcal{K}}(I,O)$ (cf. [22, Ch. 11]): a bicategorical Moore machine in $\mathbb{B}$ of fixed input and output i,o consists of a way of filling the dotted arrows in the diagram

$$\begin{tikzcd} & A \arrow[dl, "i"'] \arrow[dr, dotted, "e"] \arrow[dr, bend left=50, "o"] \arrow[d, phantom, "\overset{\delta}{\Rightarrow}"] \arrow[dr, phantom, bend left=25, "\sigma\Uparrow"] & \\ A \arrow[rr, dotted, "e"'] & & B \end{tikzcd} \tag{3.2}$$

with $e : A \to B$ and two 2-cells δ,σ. But then the 'terminal way' of filling such a span can be characterised by the right extension of the output object along a certain 1-cell obtained from the input i. Let us investigate how.

First of all, we have to assume something on the ambient hom-categories $\mathbb{B}(A,A)$, namely that each of these admits a left adjoint to the forgetful functor

$$\mathbb{B}(A,A) \longrightarrow \mathsf{Mnd}_{/A} \tag{3.3}$$

(cf. [21, Ch. II]) so that every endo-1-cell $i : A \to A$ has an associated extension to an endo-1-cell $i^\natural : A \to A$ with a unit map $i \Rightarrow i^\natural$ that is initial among all 2-cells out of i into a monad in $\mathbb{B}$; $i^\natural$ is usually called the *free monad* on i.

Construction 3.7. Now, fix i,o as in Definition 3.4; we claim that the terminal object of $\mathsf{Mre}_{\mathbb{B}}(i,o)$ is obtained as the right extension in $\mathbb{B}$ of the output o along $i^\natural$. We can obtain

- from the unit $\boldsymbol{\eta} : \mathrm{id}_A \Rightarrow i^\natural$ of the free monad on i, a canonical modification $\mathrm{Ran}_i \Rightarrow \mathrm{Ran}_{\mathrm{id}} = \mathrm{id}_A$, with components at o given by 2-cells $\sigma : \mathrm{Ran}_i o \Rightarrow o$; this is a choice of the right leg for a diagram like BMO1;

- from the multiplication $\boldsymbol{\mu} : i^\natural \circ i^\natural \Rightarrow i^\natural$ of the free monad on i, a canonical modification $\mathrm{Ran}_{i^\natural} \Rightarrow \mathrm{Ran}_{i^\natural} \circ \mathrm{Ran}_{i^\natural}$, whose components at o mate to a 2-cell $\delta_0 : \mathrm{Ran}_{i^\natural} o \circ i^\natural \Rightarrow \mathrm{Ran}_{i^\natural} o$; the composite

$$\delta : \mathrm{Ran}_{i^\natural} o \circ i \xRightarrow{\mathrm{Ran}_{i^\natural} o * \boldsymbol{\eta}} \mathrm{Ran}_{i^\natural} o \circ i^\natural \Longrightarrow \mathrm{Ran}_{i^\natural} o \tag{3.4}$$

  The left leg is now chosen for a diagram like BMO1.

Together, $(\mathrm{Ran}_{i^\natural} o, \delta, \sigma)$ is a bicategorical Mealy machine, and the universal property of the right Kan extension says it is the terminal such. A similar line of reasoning yields the same result for $\mathsf{Mly}_{\mathbb{B}}(i,o)$, only now σ is the 2-cell obtained as mate of $\epsilon \circ (\mathrm{Ran}_{i^\natural} o * \boldsymbol{\eta}) : \mathrm{Ran}_{i^\natural} o \circ i \Rightarrow \mathrm{Ran}_{i^\natural} o \circ i^\natural \Rightarrow o$ from the counit of $\_ \circ i^\natural \dashv \mathrm{Ran}_{i^\natural}$.

Proposition 3.8 (**$\mathsf{Mre}_{\mathbb{B}}(i,o)$** and **$\mathsf{Mly}_{\mathbb{B}}(i,o)$** as categories of diagrams.)**.** There exists a 2-category $\mathcal{P}$ and a pair of strict 2-functors $W, G : \mathcal{P} \to \mathbb{B}$ such that bicategorical Moore machines with 'variable output 1-cell' i.e. the 2-dimensional diagrams like in (3.2) where o is variable, can be characterised as natural transformations $W \Rightarrow G$.

Proof. The proof is deferred to the appendix, p. 18.

As explained therein, bicategorical Moore machines with fixed output o can be characterised as particular such natural transformations that take value o on one argument.

Also, minor adjustments to the shape of G yield a similar result for bicategorical Mealy machines. □

Example 3.9 (Bicategorical machines in **Cat**)**.** Consider a span $\mathcal{C} \xleftarrow{I} \mathcal{C} \xrightarrow{O} \mathcal{D}$ in the strict 2-category $\mathbb{C}\mathrm{at}$ of categories, functors and natural transformations, where $\mathcal{D}$ is a κ-complete category. The category $\mathsf{Mre}_{\mathsf{Cat}}(I,O)$ has objects the triples (E,δ,σ) where $E : \mathcal{C} \to \mathcal{D}$ is a functor and σ, δ are natural transformations arranged as in (3.2); assuming enough limits in $\mathcal{D}$, we can compute the action of the right Kan extension of O along $I^\natural$ (the free monad on the endofunctor I, cf. [46], whose existence requires additional assumptions on $\mathcal{C}$) on an object $C \in \mathcal{C}$ as the equaliser

$$RC \longrightarrow \prod_{C \in \mathcal{C}} OC^{\mathcal{C}(A, I^\natural C)} \rightrightarrows \prod_{C \to B} OB^{\mathcal{C}(A, I^\natural C)} \tag{3.5}$$

or (better, cf. [52, 2.3.6]) as the end[5] $A \mapsto \int_C OC^{\mathcal{C}(A, I^\natural C)}$, i.e. as the 'space of fixpoints' for the conjoint action of the functor O and of the presheaf $C \mapsto \mathcal{C}(A, I^\natural C)$ on objects of $\mathcal{C}$; the free monad $I^\natural$ sends an object C to the initial algebra of the functor $A \mapsto C + IA$, so that $I^\natural C \cong C + I(I^\natural C)$.

For the sake of simplicity, let us specialise the discussion when $\mathcal{D}$ is the category of sets and functions: the input I and the output O of the state machine in Definition 1.1 are now variable objects 'indexed' over the objects of $\mathcal{C}$, and the behaviour of the terminal machine can be described as a known object: unpacking the end (3.5) we obtain the functor

$$A \longmapsto [\mathcal{C}, \mathsf{Set}](\mathcal{C}(A, I^\natural \_), O) \tag{3.6}$$

sending an object A to the set of natural transformations $\alpha : \mathcal{C}(A, I^\natural \_) \Rightarrow O$; the intuition here is that to each generalised A-element of $I^\natural C$ corresponds an element of the output space $\Upsilon_C(u) \in OC$, and that this association is natural in C.

Example 3.10 (Bicategorical machines in profunctors)**.** We can reason similarly in the bicategory of categories and profunctors of [41, 16, 17], [52, Ch. 5]; now an endo-1-cell $I : \mathcal{C} \to \mathcal{C}$ on a category $\mathcal{C}$

[5] Recall that if S is a set and C is an object of a category $\mathcal{C}$ with limits, by C^S we denote the *power* of C and S, i.e. the iterated product $\prod_{s \in S} C$ of as many copies of C as there are elements in S.

consists of an 'extension' of the underlying graph of UC to a bigger graph $(UC)^+$,[6] and the free promonad $I^\natural$ (cf. [50, §5]) corresponds to the quotient of the free category on $(UC)^+$ where 'old' arrows compose as in C, and 'new' arrows compose freely; moreover, all right extensions $\langle P/Q\rangle : \mathcal{X} \rightsquigarrow \mathcal{Y}$ of $Q : \mathcal{A} \rightsquigarrow \mathcal{Y}$ along $P : \mathcal{A} \rightsquigarrow \mathcal{X}$ exist in the bicategory $\mathbb{P}\mathrm{rof}$, as they are computed as the end in [52, 5.2.5],

$$\langle P/Q\rangle : (X,Y) \longmapsto \int_A \mathsf{Set}(P(Y,A), Q(X,A)). \tag{3.7}$$

Example 3.11 (Bicategorical machines in relations)**.** When it is instantiated in the (locally thin) bicategory of relations between sets, i.e. $\{0,1\}$-profunctors, given $I : A \rightsquigarrow A$, $O : A \rightsquigarrow B$, $I^\natural$ is the reflexive-transitive closure of I, and the above Kan extension is uniquely determined as the maximal E such that $E \subseteq O$ and $E \circ I^\natural \subseteq E$ (here $\circ$ is the relational composition). So $R = \mathrm{Ran}_{I^\natural} O$ is the relation defined as

$$(a,b) \in R \iff \forall a' \in A.((a',a) \in I^\natural \Rightarrow (a',b) \in O). \tag{3.8}$$

This relation expresses *reachability* of b from a: it characterises the sub-relation of O connecting those pairs (a,b) for which, for every other $a' \in A$, if there is a finite path (possibly of length zero, i.e. $a = a'$) connecting a', a through I-related elements, then $(a',b) \in O$. In pictures:

$$a\,R\,b \iff \Big((a' = a) \lor (a' \xrightarrow{I} a_1 \xrightarrow{I} \dots \xrightarrow{I} a_n \xrightarrow{I} a) \Rightarrow a'\,O\,b\Big) \tag{3.9}$$

When the above example is specialised to the case when $A = *$ is a singleton, there are only two possible choices for I (both reflexive and transitive), and O identifies to a subset of B; a bicategorical Moore machine is then a subset $R \subseteq O$, and thus for both choices of I, $\mathsf{Mre}_{\mathbb{R}\mathrm{el}}(I,O)_{*,B} = 2^O$. One can reason in the same fashion for Mealy machines.

3.1 Intertwiners between bicategorical machines

In passing from $\mathsf{Mly}_{\mathcal{K}}(I,O)$ to $\mathsf{Mly}_{\mathbb{B}}(i,o)$ we gain an additional degree of freedom by being able to index the category over pairs of 0-cells of $\mathbb{B}$, and this is particularly true in the sense that the definition of $\mathsf{Mly}_{\mathbb{B}}(i,o)$ and its indexing over pairs of objects A,B of $\mathcal{K}$ leads to a seemingly undiscovered way to define morphisms between automata:

Definition 3.12 (Intertwiner between bicategorical machines)**.** Consider two bicategorical Mealy machines $(e,\delta,\sigma)_{A,B}, (e',\delta',\sigma')_{A',B'}$ on different bases (so in particular $(e,\delta,\sigma)_{A,B} \in \mathsf{Mly}_{\mathbb{B}}(i,o)$ and $(e',\delta',\sigma')_{A',B'} \in \mathsf{Mly}_{\mathbb{B}}(i',o')$); an *intertwiner* $(u,v) : (e,\delta,\sigma) \leftrightarrow (e',\delta',\sigma')$ consists of a pair of 1-cells $u : A \to A', v : B \to B'$ and a triple of 2-cells ι, ϵ, ω disposed as in (A.2), to which we require to satisfy the identities in (A.1) (we provide a 'birdseye' view of the commutativities that we require, as (A.2) is unambiguous about how the 2-cells $\iota, \delta, \sigma, \epsilon, \omega$ can be composed).

Remark 3.13. Interestingly enough, when it is spelled out in the case when $\mathbb{B}$ has a single 0-cell, this notion does not reduce to Remark 1.2, as an intertwiner between a Mealy machine $(E,d,s)_{I,O}$ and another $(E',d',s')_{I',O'}$ consists of a pair of objects $U,V \in \mathcal{K}$, such that

- IC1) there exist morphisms $\iota : I' \otimes U \to V \otimes I, \epsilon : E' \otimes U \to V \otimes E, \omega : O' \otimes U \to V \otimes O$;
- IC2) the following two identities hold:

$$\begin{aligned}\epsilon \circ (d' \otimes U) &= (V \otimes d) \circ (\epsilon \otimes I) \circ (E' \otimes \iota)\\ \omega \circ (s' \otimes U) &= (V \otimes s) \circ (\epsilon \otimes I) \circ (E' \otimes \iota)\end{aligned}$$

[6] More precisely, to the underlying graph of C, made of 'old' arrows, we adjoin a directed edge $e_x : C \to C'$ for each $x \in I(C,C')$.

In the single-object case, this notion does not trivialise in any obvious way, and –in stark contrast with the notion of morphism of automata given in (1.2)– intertwiners between machines support a notion of higher morphisms *even in the monoidal case*.

Definition 3.14 (2-cell between machines)**.** In the same notation of Definition 3.12, let $(u,v),(u',v') : (e,\delta,\sigma) \looparrowright (e',\delta',\sigma')$ be two parallel intertwiners between bicategorical Mealy machines; a 2-cell $(\varphi,\psi) : (u,v) \Rightarrow (u',v')$ consists of a pair of 2-cells $\varphi : u \Rightarrow u'$, $\psi : v \Rightarrow v'$ such that the identities in (A.3) hold true.

Remark 3.15. When it is specialised to the monoidal case, Definition 3.14 yields the following notion: a 2-cell $(f,g) : (U,V) \Rightarrow (U',V')$ as in Remark 3.13 consists of a pair of morphisms $f : U \to U'$ and $g : V \to V'$ subject to the conditions that the two squares in (A.4) commute: intuitively speaking, in this particular case, the machine 2-cells correspond to pairs (f,g) of $\mathcal{K}$-morphisms such that both pairs $(E' \otimes I' \otimes f, E' \otimes f)$ and $(g \otimes E \otimes I, g \otimes E)$ form morphisms in the arrow category of $\mathcal{K}$.

Remark 3.16. Let $\mathbb{B}$ be a bicategory; in [44] the authors exploit the universal property of a bicategory $\Omega\mathbb{B} = \mathsf{Psd}(\mathbf{N},\mathbb{B})$ as the category of pseudofunctors, lax natural transformations and modifications with domain the monoid of natural numbers, regarded as a single object category. The typical object of $\Omega\mathbb{B}$ is an endomorphism $i : A \to A$ of an object $A \in \mathbb{B}$, and the typical 1-cell consists of a lax commutative square

$$\begin{tikzcd} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\swarrow"] & A \arrow[d] \\ B \arrow[r] & B. \end{tikzcd} \tag{3.10}$$

This presentation begs the natural question of whether there is a tautological functor $\mathsf{Mly}_{\mathbb{B}} \to \Omega\mathbb{B}$ given by ‘projection’, sending $(i,o;(e,\delta,\sigma))$ into i; the answer is clearly affirmative, and in fact such functor mates to a unique 2-functor $\mathbf{N} \boxtimes \mathsf{Mly}_{\mathbb{B}} \to \mathbb{B}$ under the isomorphism given by Gray tensor product [31]; this somehow preserves the intuition (cf. [66, §1]) of $\Omega\mathbb{B}$ as a category of ‘lax dynamical systems’.

4 Conclusions

We sketch some directions for future research.

Conjecture 4.1. Given a monad T on Set and a quantale $\mathcal{V}$ [25, Ch. 2] we can define the locally thin bicategory $(T,\mathcal{V})$-$\mathbb{Prof}$ as in [36, Ch. III]; as $(T,\mathcal{V})$ vary we generate a plethora of bicategories, yielding the categories of topological spaces, approach spaces [53], metric and ultrametric, and closure spaces as the $(T,\mathcal{V})$*-categories* of [36, §III.1.6]. We conjecture that when instantiated in $(T,\mathcal{V})$-$\mathbb{Prof}$, Equation 3.9 yields a 2-categorical way to look at topological, metric and loosely speaking ‘fuzzy’ approaches to automata theory.

Conjecture 4.2. From Example 3.9 and 3.10 we argue that the ‘non-determinism via Kleisli category’ approach of [34] can be carried over for the presheaf construction on Cat and its Kleisli bicategory $\mathbb{Prof}$: if automata (classically intended) in the Kleisli category of the powerset monad are nondeterministic automata in Set, *bicategorical* automata in the Kleisli *bicategory* of the *presheaf construction* (cf. [26]) are nondeterministic bicategorical automata: passing from Example 3.9 to Example 3.10 accounts for a form of non-determinism. But then one might be able to address *nondeterministic* bicategorical automata in $\mathbb{B}$ as *deterministic* bicategorical automata in a generic proarrow equipment [61, 70, 71] for $\mathbb{B}$!

References

[1] J. Adámek, H. Herrlich, and G.E. Strecker. *Abstract and concrete categories: the Joy of cats*, volume 17 of *Reprints in Theory and Applications of Categories*. Theory and Applications of Categories, 2006. Available online at `http://www.tac.mta.ca/tac/reprints/articles/17/tr17.pdf`.

[2] J. Adámek. Realization theory for automata in categories. *Journal of Pure and Applied Algebra*, 9:281–296, 1977. `doi:10.1016/0022-4049(77)90071-8`.

[3] J. Adámek and V. Trnková. *Automata and Algebras in Categories*. Kluwer, 1990.

[4] B. Ahrens and P. L. Lumsdaine. Displayed Categories. *Logical Methods in Computer Science*, Volume 15, Issue 1, March 2019. URL: `https://lmcs.episciences.org/5252`, `doi:10.23638/LMCS-15(1:20)2019`.

[5] M.A. Arbib and E.G. Manes. Adjoint machines, state-behavior machines, and duality. *Journal of Pure and Applied Algebra*, 6(3):313–344, 1975. `doi:10.1016/0022-4049(75)90028-6`.

[6] M.A. Arbib and E.G. Manes. A categorist's view of automata and systems. In *Lecture Notes in Computer Science*, pages 51–64. Springer Berlin Heidelberg, 1975. `doi:10.1007/3-540-07142-3_61`.

[7] M.A. Arbib and E.G. Manes. Fuzzy machines in a category. *Bulletin of the Australian Mathematical Society*, 13(2):169–210, 1975. `doi:10.1017/S0004972700024412`.

[8] M.A. Arbib and E.G. Manes. Machines in a category. *Journal of Pure and Applied Algebra*, 19:9–20, December 1980. `doi:10.1016/0022-4049(80)90090-0`.

[9] E.S. Bainbridge. A unified minimal realization theory, with duality, for machines in a hyperdoctrine. In *Technical Report*. Computer and Communication Sciences Department, University of Michigan, 1972.

[10] E.S. Bainbridge. Addressed machines and duality. In Ernest Gene Manes, editor, *Category Theory Applied to Computation and Control*, pages 93–98, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.

[11] R. Betti and S. Kasangian. A quasi-universal realization of automata. *Università degli Studi di Trieste. Dipartimento di Scienze Matematiche*, 1981.

[12] R. Betti and S. Kasangian. Una proprietà del comportamento degli automi completi. *Università degli Studi di Trieste. Dipartimento di Scienze Matematiche*, 1982.

[13] G. Boccali, A. Laretto, F. Loregian, and S. Luneia. Completeness for categories of generalized automata. mar 2023. `arXiv:2303.03867`.

[14] R Brown. *Topology and groupoids*. www.groupoids.org, rev., updated, and expanded version edition, 2006.

[15] E. Burroni. Lois distributives. applications aux automates stochastiques. *Theory and Applications of Categories*, 22:199–221, 2009.

[16] J. Bénabou and T. Streicher. Distributors at work. Lecture notes written by Thomas Streicher, 2000.

[17] G.L. Cattani and G. Winskel. Profunctors, open maps and bisimulation. *Mathematical Structures in Computer Science*, 15(03):553–614, 2005.

[18] T. Colcombet and D. Petrisan. Automata Minimization: a Functorial Approach. In *7th Conference on Algebra and Coalgebra in Computer Science (CALCO 2017)*, volume 72 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:16, Dagstuhl, Germany, 2017. `doi:10.4230/LIPIcs.CALCO.2017.8`.

[19] P.R. D'Argenio and J.-P. Katoen. A theory of stochastic systems part i: Stochastic automata. *Information and Computation*, 203(1):1–38, November 2005. `doi:10.1016/j.ic.2005.07.001`.

[20] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. Springer Berlin Heidelberg, 2009. `doi:10.1007/978-3-642-01492-5`.

[21] E.J. Dubuc. *Kan extensions in enriched category theory*. Lecture Notes in Mathematics, Vol. 145. Springer-Verlag, Berlin-New York, 1970. `doi:10.1007/BFb0060485`.

[22] H. Ehrig, K.-D. Kiermeier, H.-J. Kreowski, and W. Kühnel. *Universal theory of automata. A categorical approach*. `doi:10.1007/978-3-322-96644-5`.

[23] S. Eilenberg. *Automata, Languages and Machines*, volume A. Academic Press, New York, 1974.

[24] S. Eilenberg and J.B. Wright. Automata in general algebras. *Information and Control*, 11(4):452–470, October 1967. doi:10.1016/s0019-9958(67)90670-5.

[25] P. Eklund, J. Gutiérrez Garciá, U. Höhle, and J. Kortelainen. *Semigroups in Complete Lattices*. Springer International Publishing, 2018. doi:10.1007/978-3-319-78948-4.

[26] M. Fiore, N. Gambino, M. Hyland, and G. Winskel. Relative pseudomonads, kleisli bicategories, and substitution monoidal structures. *Selecta Mathematica*, 24(3):2791–2830, November 2017. doi:10.1007/s00029-017-0361-3.

[27] T.M. Fiore. *Pseudo Limits, Biadjoints, And Pseudo Algebras: Categorical Foundations of Conformal Field Theory*. Memoirs AMS 860. American Mathematical Society, 2006.

[28] J.A. Goguen. Realisation is universal. *Mathematical System Theory*, 6(4), 1973.

[29] J.A. Goguen. Discrete-time machines in closed monoidal categories. I. *Journal of Computer and System Sciences*, 10(1):1–43, February 1975. doi:10.1016/s0022-0000(75)80012-2.

[30] J.A. Goguen, J. W. Thatcher, E. G. Wagner, and J. B. Wright. Factorizations, congruences, and the decomposition of automata and systems. In *Lecture Notes in Computer Science*, pages 33–45. Springer Berlin Heidelberg, 1975. doi:10.1007/3-540-07162-8_665.

[31] J.W. Gray. *Formal Category Theory: Adjointness for 2-Categories*. Springer Berlin Heidelberg, 1974. doi:10.1007/bfb0061280.

[32] R. Guitart. Remarques sur les machines et les structures. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 15:113–144, 1974.

[33] R. Guitart. Des machines aux bimodules. Univ. Paris 7, apr 1978. URL: http://rene.guitart.pagesperso-orange.fr/textespublications/rg30.pdf.

[34] R. Guitart. Tenseurs et machines. *Cahiers de topologie et géométrie différentielle*, 21(1):5–62, 1980. URL: http://www.numdam.org/item/CTGDC_1980__21_1_5_0/.

[35] R. Guitart and L. Van den Bril. Décompositions et Lax-complétions. *Cahiers de topologie et géométrie différentielle*, 18(4):333–407, 1977. URL: http://www.numdam.org/item/CTGDC_1977__18_4_333_0/.

[36] D. Hofmann, G.J. Seal, and W. Tholen. *Monoidal Topology: A Categorical Approach to Order, Metric, and Topology*, volume 153. Cambridge University Press, 2014. doi:10.1017/CBO9781107517288.

[37] B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in SLFM. Elsevier, 1999.

[38] B. Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In *Algebra, Meaning, and Computation*, pages 375–404. Springer Berlin Heidelberg, 2006. doi:10.1007/11780274_20.

[39] B. Jacobs. New directions in categorical logic, for classical, probabilistic and quantum logic. *Logical Methods in Computer Science*, 11(3):76, 2015. Id/No 24. doi:10.2168/LMCS-11(3:24)2015.

[40] B. Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016. doi:10.1017/CBO9781316823187.

[41] M. B. Justesen. Bikategorien af profunktorer. Naturvidenskabelig embedseksamen, Aarhus University, 1968.

[42] S. Kasangian, G.M. Kelly, and F. Rossi. Cofibrations and the realization of non-deterministic automata. *Cahiers de topologie et géométrie différentielle catégoriques*, 24(1):23–46, 1983.

[43] S. Kasangian and R. Rosebrugh. Glueing enriched modules and composition of automata. *Cahiers de topologie et géométrie différentielle catégoriques*, 31(4):283–290, 1990.

[44] P. Katis, N. Sabadini, and R.F.C. Walters. Bicategories of processes. *Journal of Pure and Applied Algebra*, 115(2):141–178, February 1997. doi:10.1016/s0022-4049(96)00012-6.

[45] P. Katis, N. Sabadini, and R.F.C. Walters. Feedback, trace and fixed-point semantics. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 36(2):181–194, 2002. URL: http://www.numdam.org/articles/10.1051/ita:2002009/, doi:10.1051/ita:2002009.

[46] G.M. Kelly. A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on. *Bulletin of the Australian Mathematical Society*, 22(01):1–83, 1980. doi:10.1017/S0004972700006353.

[47] A. Kock. *Limit monads in categories*. The University of Chicago, 1967.

[48] A. Kock. Closed categories generated by commutative monads. *J. Austral. Math. Soc.*, 12:405–424, 1971.

[49] A. Kock. Strong functors and monoidal monads. *Archiv der Mathematik*, 23(1):113–120, 1972.

[50] S. Lack. Note on the construction of free monoids. *Applied Categorical Structures*, 18(1):17–29, October 2008. doi:10.1007/s10485-008-9167-y.

[51] E. Di Lavore, A. Gianola, M. Román, N. Sabadini, and P. Sobociński. A canonical algebra of open transition systems. In Gwen Salaün and Anton Wijs, editors, *Formal Aspects of Component Software - 17th International Conference, FACS 2021, Virtual Event, October 28-29, 2021, Proceedings*, volume 13077 of *Lecture Notes in Computer Science*, pages 63–81. Springer, 2021. doi:10.1007/978-3-030-90636-8_4.

[52] F. Loregian. *Coend Calculus*, volume 468 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, first edition, jul 2021. ISBN 9781108746120.

[53] R. Lowen. *Approach Spaces: The Missing Link in the Topology-uniformity-metric Triad*. Oxford mathematical monographs. Clarendon Press, 1997.

[54] J. Meseguer and U. Montanari. Petri nets are monoids. *Information and Computation*, 88(2):105–155, October 1990. doi:10.1016/0890-5401(90)90013-8.

[55] J. Meseguer and I. Sols. Automata in semimodule categories. In *Lecture Notes in Computer Science*, pages 193–198. Springer Berlin Heidelberg, 1975. doi:10.1007/3-540-07142-3_81.

[56] G. Naudé. On the adjoint situations between behaviour and realization. *Quaestiones Mathematicae*, 2:245–267, 1977. doi:10.1080/16073606.1977.9632546.

[57] G. Naudé. Universal realization. *Journal of Computer and System Sciences*, 19(3):277–289, 1979. doi:10.1016/0022-0000(79)90005-9.

[58] P. Perrone and W. Tholen. Kan extensions are partial colimits. *Applied Categorical Structures*, 30(4):685–753, January 2022. doi:10.1007/s10485-021-09671-9.

[59] I. Pohl and M.A. Arbib. Theories of abstract automata. *Mathematics of Computation*, 24(111):760, jul 1970. doi:10.2307/2004866.

[60] R. Rosebrugh, N. Sabadini, and R.F.C. Walters. Minimal realization in bicategories of automata. *Mathematical Structures in Computer Science*, 8(2):93–116, 1998. doi:10.1017/S0960129597002454.

[61] R. Rosebrugh and R.J. Wood. Proarrows and cofibrations. *Journal of Pure and Applied Algebra*, 53(3):271–296, 1988. doi:10.1016/0022-4049(88)90128-4.

[62] J.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, October 2000. doi:10.1016/s0304-3975(00)00056-6.

[63] S. Singh and S.P. Tiwari. On the category of L-fuzzy automata, coalgebras and dialgebras. *Fuzzy Sets and Systems*, 420:1–28, September 2021. doi:10.1016/j.fss.2020.07.013.

[64] R. Street. Fibrations and Yoneda lemma in a 2-category. In G.M. Kelly, editor, *Proceedings Sydney Category Theory Seminar 1972/1973*, volume 420 of *Lecture Notes in Mathematics*, pages 104–133. Springer, 1974. doi:10.1007/BFb0063096.

[65] R. Street. Fibrations in bicategories. *Cahiers de topologie et géométrie différentielle catégoriques*, 21(2):111–160, 1980.

[66] M. Tierney. *Categorical Constructions in Stable Homotopy Theory*. Springer Berlin Heidelberg, 1969. doi:10.1007/bfb0101425.

[67] Y. Venema. Automata and fixed point logic: A coalgebraic perspective. *Information and Computation*, 204(4):637–678, April 2006. doi:10.1016/j.ic.2005.06.003.

[68] VV.AA. Fugal: definition & meaning. URL: https://www.merriam-webster.com/dictionary/fugal.

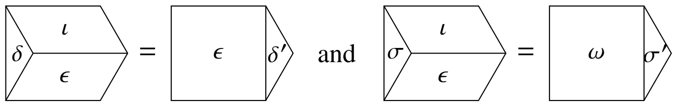
[69] R.F.C. Walters. A note on context-free languages. *Journal of Pure and Applied Algebra*, 62(2):199–203, December 1989. doi:10.1016/0022-4049(89)90151-5.

[70] R.J. Wood. Abstract pro arrows I. *Cahiers de topologie et géométrie différentielle*, 23(3):279–290, 1982. URL: http://www.numdam.org/item/CTGDC_1982__23_3_279_0/.

[71] R.J. Wood. Proarrows II. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 26(2):135–168, 1985. URL: http://www.numdam.org/item/CTGDC_1985__26_2_135_0/.

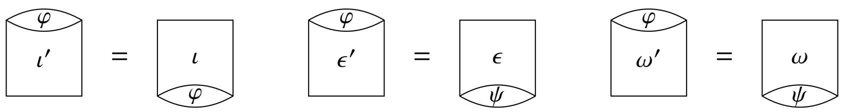
A Appendix A: Proofs

A.1 Diagrams



(A.1)

$$\begin{tikzcd} A \arrow[r, "u"] \arrow[d, "i"'] \arrow[dr, phantom, "\Swarrow\iota"] & A' \arrow[d, "i'"] \\ A \arrow[r, "u"'] & A' \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "u"] \arrow[d, "e"'] \arrow[dr, phantom, "\Swarrow\epsilon"] & A' \arrow[d, "e'"] \\ B \arrow[r, "v"'] & B' \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "u"] \arrow[d, "o"'] \arrow[dr, phantom, "\Swarrow\omega"] & A' \arrow[d, "o'"] \\ B \arrow[r, "v"'] & B' \end{tikzcd} \tag{A.2}$$



(A.3)

$$\begin{tikzcd} E'\otimes I'\otimes U \arrow[r, "d'\otimes U"] \arrow[d, "E'\otimes I'\otimes f"'] & E'\otimes U \arrow[d, "E'\otimes f"] \\ E'\otimes I'\otimes U' \arrow[r, "d'\otimes U'"'] & E'\otimes U' \end{tikzcd} \qquad \begin{tikzcd} V\otimes E\otimes I \arrow[r, "V\otimes d"] \arrow[d, "g\otimes E\otimes I"'] & V\otimes E \arrow[d, "g\otimes E"] \\ V'\otimes E\otimes I \arrow[r, "V'\otimes d"'] & V'\otimes E \end{tikzcd} \tag{A.4}$$

A.2 Proofs

Proof of Lemma 2.13. In order to prove that the assignment $s \mapsto s^\flat$ is well defined in the set of fugal automata, we proceed by induction on the length of a string ℓ. We have to prove that

$$s^\flat(e, \ell +\!\!+ as) = s^\flat(e,\ell) +\!\!+ s^\flat(d^*(e,\ell), as) \tag{A.5}$$

The base case $\ell = [\,]$ is evidently true, so suppose that $\ell = x :: xs$ is not empty and the claim is true for every choice of a shorter xs: then,

$$\begin{aligned} s^\flat(e,(x::xs) +\!\!+ as) &= s^\flat(e,(x::xs) +\!\!+ as) \\ &= s^\flat(e, x :: (xs +\!\!+ as)) \\ &= s(e,x) :: s^\flat(d(e,x), xs +\!\!+ as) \\ &= s(e,x) :: \big(s^\flat(d(e,x),xs) +\!\!+ s^\flat(d^*(e,xs),as)\big) \\ &= \big(s(e,x) :: s^\flat(d(e,x),xs)\big) +\!\!+ s^\flat(d(x,d^*(e,xs)),as) \\ &= s^\flat(e,x::xs) +\!\!+ s^\flat(d^*(e,x::xs),as) \\ &= s^\flat(e,\ell) +\!\!+ s^\flat(d^*(e,\ell),as). \end{aligned}$$

We now have to show that any 2-cell $f : (E,d,s) \to (F,c,t)$ is in fact a 2-cell $(E,d^*,s^\flat) \to (F,c^*,t^\flat)$. This can be done by induction as well, with completely similar reasoning. ◻

Proof of Lemma 2.16. We have to prove that

$$(s_{21})^\flat = s_{21}^{\flat\flat}. \tag{A.6}$$

The two functions coincide on the empty list by definition; hence, let $\ell = a :: as$ be a nonempty list and $(e, f) \in E \times F$ a generic element. The right-hand side of the equation is

$$\begin{aligned}
(s_{21}^{\flat\flat})((e,f), a :: as) &= s_2^\flat(f, s_1^\flat(e, a :: as)) \\
&= s_2^\flat\big(f, s_1(e,a) :: s_1^\flat(d_1(e,a), as)\big) \\
&= s_2(f, s_1(e,a)) :: s_2^\flat(d_2(f, s_1(e,a)), s_1^\flat(d_1(e,a), as)) \\
&= (s_{21})((e,f),a) :: (s_{21})^\flat((d_2 \fatsemi d_1)((e,f),a), as)
\end{aligned}$$

which concludes the proof. □

Proof of Theorem 2.17. Similarly to Lemma 2.16, we have to prove that $d_2^* \fatsemi d_1^* = (d_2 \fatsemi d_1)^*$ whenever d_2, d_1 are two dynamic maps of composable Mealy machines $\langle s_1, d_1 \rangle : E \times M \to N \times E$ and $\langle s_2, d_2 \rangle : F \times N \to P \times F$. This, together with Lemma 2.16, will establish functoriality on 1-cells of $(\_)^\flat$. Functoriality on 2-cells is very easy to establish. For what concerns Π, the proof amounts to showing that the composition of (fugal) Mealy machines gets mapped into the composition of spans in $\mathsf{Mac}^{\mathsf{s}}$; this can be checked with ease and follows from the fact that the translation category of the action $d_2 \fatsemi d_1$ as defined in (2.2) has the universal property of the pullback $\mathcal{Z}$ in

$$\begin{tikzcd}
& & \mathcal{Z} \arrow[dl] \arrow[dr] & & \\
& \mathcal{E}[d_1^*] \arrow[dl, "D_1"'] \arrow[dr, "\Sigma_1"] & & \mathcal{E}[d_2^*] \arrow[dl, "D_2"'] \arrow[dr, "\Sigma_2"] & \\
M & & N & & R.
\end{tikzcd} \tag{A.7}$$

This is a straightforward check, and it is also straightforward to see that the composition of Σ_2 with the right projection from $\mathcal{Z}$ coincides with the 'Sigma' functor induced by s_{21}, which concludes the proof. □

Proof of Theorem 2.18. It is worthwhile to recall what a biadjunction is

$$F : \mathsf{C} \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} \mathsf{D} : G \tag{A.8}$$

if C, D are bicategories (cf. [27, Ch. 9]): for each two objects C, D we are given an equivalence between hom-categories $\mathsf{D}(FC, D) \simeq \mathsf{C}(C, GD)$, i.e. a pair of functors $H : \mathsf{D}(FC, D) \leftrightarrows \mathsf{C}(C, GD) : K$ whose composition in both directions is isomorphic to the identity functor of the respective hom-category –and all this depends naturally on C, D.

In order to prove this, let's fix a set A and a monoid M, let's build functors

$$\mathsf{Mly}^\flat_{\mathsf{Set}}(A^*, M) \xrightarrow{H} \mathsf{Mly}_{\mathsf{Set}}(A, UM) \qquad \mathsf{Mly}_{\mathsf{Set}}(A, UM) \xrightarrow{K} \mathsf{Mly}^\flat_{\mathsf{Set}}(A^*, M) \tag{A.9}$$

and prove that they form an equivalence of categories by explicitly showing that HK and KH are isomorphic to the respective identities. We'll often adopt the convenient notation $\langle s, d \rangle : E \times X \to Y \times E$ for a Mealy machine of input X and output Y.

- Let $\langle s,d\rangle : E\times A^* \to M\times E$ be a fugal Mealy machine; $H\langle s,d\rangle$ is defined as the composition

$$E\times A \xrightarrow{E\times\eta_A} E\times A^* \xrightarrow{\langle s,d\rangle} M\times E \tag{A.10}$$

  where $\eta_A : A \to A^*$ is the unit of the free-forgetful adjunction between Set and monoids. In simple terms, H acts 'restricting' a fugal Mealy machine to the set of generators of its input.

- Let $\langle s_0,d_0\rangle : F\times A \to UM\times F$ be any Mealy machine on Set, where UM means that M is regarded as a mere set; $K\langle s_0,d_0\rangle$ is defined as the composition

$$F\times A^* \xrightarrow{\langle s_0,d_0\rangle^\flat} (UM)^*\times F \xrightarrow{\varepsilon\times F} M\times F \tag{A.11}$$

  where $\varepsilon : (UM)^* \to M$ is the counit of the free-forgetful adjunction between Set and monoids, and $\langle s_0,d_0\rangle^\flat$ is the fugal extension of Lemma 2.13.

The claim is now that the fugal Mealy machine $KH\langle s,d\rangle$ coincides with $\langle s,d\rangle$, and that the generic Mealy machine $HK\langle s_0,d_0\rangle$ coincides with $\langle s_0,d_0\rangle$.

Both statements depend crucially on the following fact: if $s : E\times M \to N$ satisfies Equation (2.8), then for all $e\in E$ the element $s(1_M,e)$ is idempotent in N. In particular, if N is free on a set B, $s(1_M,e) = [\,]$ is the empty list, and more in particular, for a generic Mealy machine $\langle s,\_\rangle$ the fugal extension $s^\flat$ is such that for all $e\in E$, $s^\flat([\,],e) = [\,]$.

Given this, observe that the Mealy machine $HK\langle s_0,d_0\rangle$ coincides with $\langle s_0^\flat\circ(F\times\eta_A), d_0^*\circ(F\times\eta_A)\rangle$; now clearly the composition $d_0^*\circ(F\times\eta_A)$ coincides with $d_0 : F\times A\to F$ and the two maps determine each other. As for $s_0^\flat\circ(F\times\eta_A)$, we have that for all $(f,a)\in F\times A$

$$\begin{aligned} s_0^\flat\circ(F\times\eta_A)(f,a) &= s_0^\flat(f,a::[\,]) \\ &= s_0(f,a) :: s_0^\flat(f,[\,]) \\ &= s_0(f,a) :: [\,] \end{aligned}$$

Reasoning similarly, one proves that the fugal Mealy machine $KH\langle s,d\rangle$ has components $\langle (s\circ(E\times\eta_A))^\flat, (d\circ(E\times\eta_A))^*\rangle$: again, since functions $E\times A\to E$ correspond bijectively to monoid actions $E\times A^*\to E$, the map $(d\circ(E\times\eta_A))^*$ coincides with d; as for $(s\circ(E\times\eta_A))^\flat$, we can argue by induction that

$$\begin{aligned} (s\circ(E\times\eta_A))^\flat(e,[\,]) &= [\,] = s(e,[\,]) \\ (s\circ(E\times\eta_A))^\flat(e,a::as) &= s(e,a) :: (s\circ(E\times\eta_A))^\flat(d(a,e),as) \\ &= s(e,a) :: s(d(a,e),as) \\ &= s(e,a::as) \end{aligned}$$

where the last equality uses that s was fugal to start with. This concludes the proof. □

Proof of Proposition 3.8. The category $\mathcal{P}$ is in fact 2-discrete (it has no 2-cells) and its objects and morphisms are arranged as follows:

$$1 \underset{y}{\overset{x}{\leftleftarrows}} 0 \underset{t}{\overset{z}{\rightrightarrows}} 2 \tag{A.12}$$

For lack of a better name, $\mathcal{P}$ is the *generic double span*.

The functors W,G are then constructed as follows:

- $G : \mathcal{P} \to \mathsf{Cat}$ is constant on objects at the category $\mathcal{K}(A,B)$, and chooses the double span

$$\mathcal{K}(A,B) \underset{\mathrm{id}}{\overset{\mathrm{id}}{\leftleftarrows}} \mathcal{K}(A,B) \underset{\mathrm{id}}{\overset{\_\circ i}{\rightrightarrows}} \mathcal{K}(A,B); \tag{A.13}$$

- $W : \mathcal{P} \to \mathsf{Cat}$ chooses the double span

$$\{0 \to 1\} \underset{j}{\overset{j}{\leftleftarrows}} \{\heartsuit, \spadesuit\} \underset{c_1}{\overset{c_0}{\rightrightarrows}} \{0 \to 1\} \tag{A.14}$$

where $\{\heartsuit, \spadesuit\}$ is a discrete category with two objects, $j = \binom{\heartsuit \mapsto 0}{\spadesuit \mapsto 1}$, and c_k is constant at $k \in \{0,1\}$.

Now, it is a matter of unwinding the definition of a natural transformation $\alpha : W \Rightarrow G$ to find that we are provided with maps

$$\begin{aligned} \{e, \#\} = \alpha_0 &: W0 \to \mathcal{K}(A,B) \\ \sigma = \alpha_1 &: W1 \to \mathcal{K}(A,B) \\ \delta = \alpha_2 &: W2 \to \mathcal{K}(A,B) \end{aligned} \tag{A.15}$$

and with commutative diagrams arising from naturality as follows, if we agree to label $\alpha_0(\heartsuit) = e$ and $\alpha_0(\spadesuit) = o$, and we blur the distinction between α_0 and the embedding of its image $\{e, o\}$ in $\mathcal{K}(A,B)$:

$$\begin{tikzcd} \{e,o\} \arrow[r, "j"] \arrow[d, "\alpha_0"'] & \{0 \to 1\} \arrow[d, "\alpha_1"] \\ \mathcal{K}(A,B) \arrow[r, equal] & \mathcal{K}(A,B) \end{tikzcd} \qquad \begin{tikzcd} \{e,o\} \arrow[r, "c_0"] \arrow[d, "\alpha_0"'] & \{0 \to 1\} \arrow[d, "\alpha_2"] \\ \mathcal{K}(A,B) \arrow[r, "\_\circ i"'] & \mathcal{K}(A,B) \end{tikzcd} \qquad \begin{tikzcd} \{e,o\} \arrow[r, "c_1"] \arrow[d, "\alpha_0"'] & \{0 \to 1\} \arrow[d, "\alpha_2"] \\ \mathcal{K}(A,B) \arrow[r, equal] & \mathcal{K}(A,B) \end{tikzcd} \tag{A.16}$$

Altogether, we have that these data yield a diagram of 2-cells

$$\begin{tikzcd} & A \arrow[dl, "i"'] \arrow[dr, "e"'] \arrow[dr, bend left=60, "o"] \arrow[d, phantom, "\overset{\delta}{\Rightarrow}"] \arrow[dr, phantom, bend left=30, "\sigma\Uparrow"] & \\ A \arrow[rr, "e"'] & {} & B \end{tikzcd} \tag{A.17}$$

as in (3.2). Modifications between these natural transformations correspond to suitable arrangements of 2-cells, in such a way that we recover the notion of morphism of bicategorical Moore machine given in BMO2.

In case the output o is fixed, we just constrain $\alpha_0(\spadesuit)$ to be mapped in o and modifications to be the identity at $\spadesuit$.

For bicategorical Mealy machines, redefine $Gx = Gz = \_ \circ i$ and the rest of the argument is unchanged. $\square$

Discussion A.1. In a world of war and crippling inflation bytes are expensive, so page limits shorten by the month. *This forces authors to shrink their papers, and the only way to do that is remove text.*

A simple interpolation suggests that *one day, the average submission will consist of just the picture of a cat surrounded by a circle and a square*; already today, we feel constrained to push in the appendix the email addresses of the authors: [†]`guidoboccali@gmail.com`, [⊛]`anlare@ttu.ee`, [♣]`folore@ttu.ee`, and [♡]`stefano.luneia@gmail.com`.

Subsumptions of Algebraic Rewrite Rules

Thierry Boy de la Tour
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
thierry dot boy-de-la-tour at imag.fr

What does it mean for an algebraic rewrite rule to subsume another rule (that may then be called a subrule)? We view subsumptions as rule morphisms such that the simultaneous application of a rule and a subrule (i.e. the application of a subsumption morphism) yields the same result as a single application of the subsuming rule. Simultaneous applications of categories of rules are obtained by Global Coherent Transformations and illustrated on graphs in the DPO approach. Other approaches are possible since these transformations are formulated in an abstract Rewriting Environment, and such environments exist for various approaches to Algebraic Rewriting, including DPO, SqPO and PBPO.

1 Introduction

In Global Transformations [16] rules may be seen as pairs (L,R) of graphs (or objects in a category $\mathscr{C}$) that are applied simultaneously to an input graph (as in L-systems [10] and cellular automata [9]). Such rules are related by pairs of $\mathscr{C}$-morphisms. These morphisms come from representing possible overlaps of rules as subrules whose applications are induced by the overlapping applications of rules, therefore establishing a link between these. By computing a colimit of a diagram involving the morphisms between occurrences of right-hand sides, Global Transformations offer the possibility to merge items (vertices or edges) in these occurrences of right-hand sides.

This form of rules has the advantage of simplicity, first because rule morphisms are those of the product category $\mathscr{C} \times \mathscr{C}$, and second because the input object is completely removed. Indeed, when all occurrences of L have been found in the input graph G, the output graph H is produced solely from the corresponding occurrences of R, thus effectively removing G. In particular, if no L has any match in G then H is the empty graph. If G is, say, a relational database, this may be inconvenient.

More standard approaches to algebraic rewriting use rules for *replacing* matched parts of the input object by new parts. These substitutions are performed by first removing the matched part and then adding the new part, this last operation being performed by a pushout. But since there is no general algebraic way of removing parts of a $\mathscr{C}$-object, several approaches have been devised, from DPO [7] to PBPO [4] rules, for defining the *context* (a $\mathscr{C}$-object) in which R can be "pushed". These rules always have an interface K with a pair of $\mathscr{C}$-morphisms from K to L and R (a span), but can be more complicated. Hence the necessity of a general notion of morphism between rules that does not depend on a specific shape of rules.

In Section 3 an intuitive analysis of rule subsumptions on a simple example with DPO-rules leads to a natural definition of subsumption morphisms between DPO-rules, and of corresponding subsumption morphisms between direct DPO-transformations. This leads in Section 4 to a general notion of *Rewriting Environment* that provides the relevant categories of rules and of direct transformations, and functors between them and to a category of *partial transformations*.

Section 5 is devoted to the Global Coherent Transformation. It derives from the Parallel Coherent Transformations defined in [2] (only for a variant of DPO-rules), where sets or rules can be applied

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 20–38, doi:10.4204/EPTCS.397.2

simultaneously on an input object. The first step defines the *global context* as a limit of a diagram that involves the subsumption morphisms.

One important problem is that overlapping applications of rules (i.e., overlapping direct transformations) may conflict as one transformation deletes an item of G that another transformation preserves. Note that conflicts cannot happen with Global Transformations since they preserve nothing. Only non conflicting, so called *coherent* transformations can be applied simultaneously, hence the notion of Parallel Coherence from [2] must be adapted in order to embrace subsumption morphisms. The adapted definition ensures that the right-hand sides of the rules can be pushed in the global context by means of a colimit.

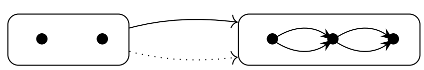
Section 6 is devoted to the analysis of Rewriting Environments, and yields natural definitions of environments for the SqPO and PBPO approaches. Future work and open questions are found in Section 7.

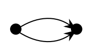
2 Notations

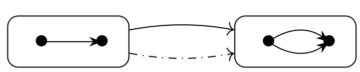
Embeddings are injective functors, all other notions are compatible with [15]. We also use *meets* and *sums* of functors, see [12].

For any category $\mathscr{C}$, we write $G \in \mathscr{C}$ to indicate that G is a $\mathscr{C}$-object, and $|\mathscr{C}|$ is the discrete category on $\mathscr{C}$-objects. Then G also denotes the functor from the terminal category $\mathbf{1}$ to $|\mathscr{C}|$ that maps the object of $\mathbf{1}$ to G. $\varnothing$ denotes the initial object of $\mathscr{C}$, if any. The *slice* category $\mathscr{C} \setminus G$ has as objects $\mathscr{C}$-morphisms of codomain G, and as morphisms $h : f \to g$ $\mathscr{C}$-morphisms such that $g \circ h = f$. The *coslice* category $G \setminus \mathscr{C}$ has as objects $\mathscr{C}$-morphisms of domain G, and as morphisms $h : f \to g$ $\mathscr{C}$-morphisms such that $h \circ f = g$.

We will use the standard notion of graphs with multiple directed edges. In the running example we will use graphs with 2 to 3 vertices and 0 to 4 edges denoted directly by their drawings, as in  and . In order to avoid naming vertices, they will always be depicted from left to right, and we will use at most two monomorphisms from one graph to another: one (depicted as a plain arrow) that maps the leftmost (resp. rightmost) vertex of the domain graph to the leftmost (resp. rightmost) vertex of the codomain graph, and one (dotted arrow) that swaps these vertices. For example we consider only two possible morphisms:



The two morphisms from to  will be distinguished similarly:



3 Subrules in DPO Graph Transformations

The notion of a rule ρ being a subrule of a rule ρ', or more generally of a subsumption morphism $\sigma : \rho \to \rho'$, covers the idea that ρ represents a part (specified by σ) of what ρ' achieves, and therefore that any application of ρ' entails and subsumes a particular application (obtained through σ) of ρ. We first try to make this idea more precise with DPO-rules.

Definition 3.1 (DPO rules and direct transformations, gluing condition)**.** A *DPO-rule* ρ in a category $\mathscr{C}$ is a span diagram

$$L \xleftarrow{\;l\;} K \xrightarrow{\;r\;} R$$

in $\mathscr{C}$, where l is monic. Diagrams in $\mathscr{C}$ are functors from an index category to $\mathscr{C}$, and it will sometime be convenient to refer to the objects and morphisms of this index category; they will be denoted by the corresponding roman letters (here $\rho\mathrm{L} = L$, $\rho\mathrm{l} = l$, etc.)

We say that an item (edge or vertex) of a graph G is *marked for removal* by a *matching* $m : L \to G$ for a rule ρ if it has a preimage by m that has none by l (see [3]). The *gluing condition* for m, ρ states that

$$\begin{cases} \text{all items marked for removal have only one preimage by } m, & \text{(GC1)} \\ \text{if a vertex adjacent to an edge is marked for removal, then so is this edge.} & \text{(GC2)} \end{cases}$$

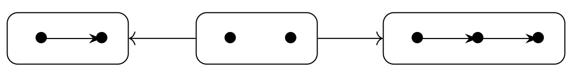
A *direct DPO-transformation* δ in $\mathscr{C}$ is a diagram

$$\begin{tikzcd} L \arrow[d, "m"'] & K \arrow[l, tail, "l"'] \arrow[r, "r"] \arrow[d, "k"] & R \arrow[d, "n"] \\ G & D \arrow[l, "f"] \arrow[r, "g"'] & H \end{tikzcd}$$

in $\mathscr{C}$ such that l is monic and the two squares are pushouts.

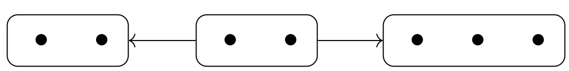
It is well known (see [8, 6]) that in the category of graphs, given ρ and $m : L \to G$, there exists a direct DPO-transformation δ with ρ and m iff the gluing condition holds. The *pushout complement* D is then a subgraph of G (f is monic) and contains all the items of G that are not marked for removal.

Example 3.2. In the running example we transform every directed edge in a graph into a pair of consecutive edges. This can be expressed as the following rule



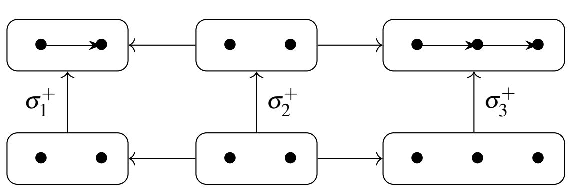
(ρ')

We do not wish to transform loops in this way, hence we adopt the DPO approach restricted to monic matchings. We also wish to create only one middle vertex for parallel edges, so that the input graph $G =$  in our running example shall be transformed into $H =$  . In order to merge the two vertices created by the two simultaneous applications of ρ' on G we need to link them through the application of a common subrule on their overlap. Consider the rule



(ρ)

The right hand side expresses the fact that the middle vertex is created depending on the overlap and not on the edges of G. Thus we need to link the middle vertices from ρ and ρ' right-hand sides through a morphism $\sigma^+ : \rho \to \rho'$, given as three $\mathscr{C}$-morphisms:



(σ^+)

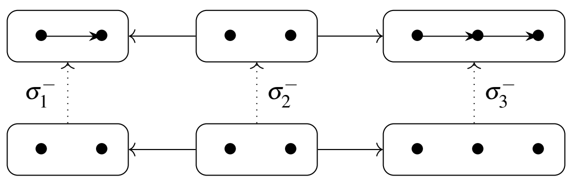
The two square diagrams commute, and we easily understand that this is necessary for ρ to be a subrule of ρ'. But commutation would also hold if the interface graph of ρ were $\varnothing$, and then ρ would remove the overlap $\bullet \quad \bullet$. This would conflict with ρ' that preserves this part of G. We need the two rules to behave similarly on the overlap, which means that the interface of the subrule ρ is determined by the way the interface of ρ' intersects the overlap. This can be expressed by stating that the left square should be a pullback.

Definition 3.3 (categories $\mathscr{R}_{\text{DPO}}$, $\mathscr{R}_{\text{DPOm}}$)**.** For any category $\mathscr{C}$, let $\mathscr{R}_{\text{DPO}}$ be the category whose objects are the DPO-rules and morphisms (or *subsumptions*) $\sigma : \rho \to \rho'$ are triples $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ of $\mathscr{C}$-morphisms such that

$$\begin{tikzcd} L \arrow[d, "\sigma_1"'] & K \arrow[l, tail, "l"'] \arrow[r, "r"] \arrow[d, "\sigma_2"] \arrow[dl, phantom, "\lrcorner"] & R \arrow[d, "\sigma_3"] \\ L' & K' \arrow[l, tail, "l'"] \arrow[r, "r'"'] & R' \end{tikzcd}$$

(where $L' = \rho'$L etc.) commutes in $\mathscr{C}$ and the left square is a pullback. Composition is componentwise and the obvious identities are $1_\rho = (1_L,\, 1_K,\, 1_R)$ (this is a subcategory of $\mathscr{C}^{\cdot \leftarrow \cdot \rightarrow \cdot}$). Let $\mathscr{R}_{\text{DPOm}}$ be the subcategory of $\mathscr{R}_{\text{DPO}}$ with all rules and all morphisms σ such that σ_1 and σ_2 are monics.

Example 3.4. We consider two morphisms of rules, σ^+ above and $\sigma^- : \rho \to \rho'$ that swaps the left and right vertices:

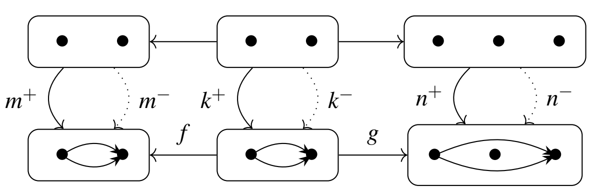


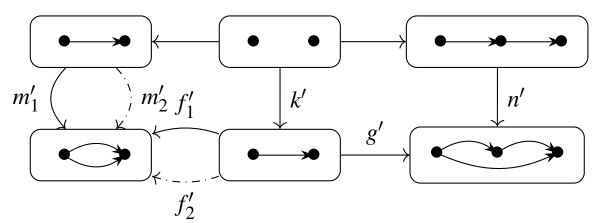
(σ^-)

We now see that the gluing condition is inherited (backward) along the morphisms of $\mathscr{R}_{\text{DPOm}}$.

Proposition 3.5. *If $\mathscr{C}$ is the category of graphs, $\sigma : \rho \to \rho'$ is a morphism in $\mathscr{R}_{\text{DPO}}$ such that σ_1 is monic and $m' : L' \to G$ satisfies the gluing condition for ρ' then so does $m' \circ \sigma_1 : L \to G$ for ρ.*

Example 3.6. There are two obvious matchings m'_1 and m'_2 of ρ' in G, and they induce two matchings of ρ in G, say $m^+ = m'_1 \circ \sigma_1^+ = m'_2 \circ \sigma_1^+$ and $m^- = m'_1 \circ \sigma_1^- = m'_2 \circ \sigma_1^-$. We see that m'_1 and m'_2 satisfy the gluing condition, hence they have a pushout complement by l' and so do m^+ and m^- by l. We therefore get two DPO-transformations of G by ρ (below left), one with (m^+, k^+, n^+, f, g) and the other with (m^-, k^-, n^-, f, g), and two DPO-transformations of G by ρ' (below right), one with (m'_1, k', n', f'_1, g') and the other with (m'_2, k', n', f'_2, g').





The following result reveals the relationship induced by morphisms $\sigma : \rho \to \rho'$ on the corresponding direct DPO-transformations.

Proposition 3.7. *If $\mathscr{C}$ is the category of graphs, $\sigma : \rho \to \rho'$ is a morphism in $\mathscr{R}_{\text{DPO}}$, $m' : L' \to G$ and $m' \circ \sigma_1 : L \to G$ have pushout complements as below, then there is a unique graph morphism d such that*

$$\begin{tikzcd}
 & L \arrow[dl, "\sigma_1"'] \arrow[dd] & & K \arrow[ll, tail, "l"'] \arrow[dl, "\sigma_2"] \arrow[dd, "k"] \arrow[dll, phantom, "\lrcorner"] \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, tail, "l'"] \arrow[dd, "k'"'] & \\
 & G \arrow[dl, phantom, "="] & & D \arrow[ll, tail, "f"'] \\
G & & D' \arrow[ll, tail, "f'"] \arrow[ur, dashed, tail, "d"'] &
\end{tikzcd}$$

commutes.

The existence of d means that all items marked for removal by $m' \circ \sigma_1$, i.e., removed by the subrule ρ, are also removed by ρ'. In Example 3.6 we have $f = 1_G$, hence with $m' = m'_i$ we get $d = f'_i$. We also see that there are no morphisms between the results of the transformations of G by ρ and ρ', in either direction. This is due to the fact that subrules remove less, but also add less. Subsumptions of rules cannot be deduced from the properties of the transformation functions (from $|\mathscr{C}|$ to $|\mathscr{C}|$) they induce.

Definition 3.8 (categories $\mathscr{D}_{\text{DPO}}$, $\mathscr{D}_{\text{DPOm}}$, functors R_{DPO}, R_{DPOm})**.** Let $\mathscr{D}_{\text{DPO}}$ be the category whose objects are direct DPO-transformations in a category $\mathscr{C}$ and whose morphisms (or *subsumptions*) $\mu : \delta \to \delta'$ are 4-tuples $(\mu_1, \mu_2, \mu_3, \mu_4)$ of $\mathscr{C}$-morphisms such that the following diagram

$$\begin{tikzcd}
 & L \arrow[dl, "\mu_1"'] \arrow[dd, "m"] & & K \arrow[ll, tail, "l"'] \arrow[rr, "r"] \arrow[dl, "\mu_2"] \arrow[dd, "k"] \arrow[dll, phantom, "\lrcorner"] & & R \arrow[dl, "\mu_3"] \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, tail, "l'"] \arrow[rr, "r'"] \arrow[dd, "k'"'] & & R' & \\
 & G \arrow[dl, phantom, "="] & & D \arrow[ll, tail, "f"'] & & \\
G' & & D' \arrow[ll, tail, "f'"] \arrow[ur, "\mu_4"'] & & &
\end{tikzcd}$$

commutes and the top left square is a pullback, with componentwise composition (but due to the contravariance of μ_4 we are not in a functor category anymore). Let R_{DPO} be the obvious functor from $\mathscr{D}_{\text{DPO}}$ to $\mathscr{R}_{\text{DPO}}$, i.e. such that $(\mathsf{R}_{\text{DPO}}\delta)\mathsf{L} = \delta\mathsf{L}$ etc. and $\mathsf{R}_{\text{DPO}}\mu = (\mu_1, \mu_2, \mu_3)$. Let $\mathscr{D}_{\text{DPOm}}$ be the full subcategory of $\mathscr{D}_{\text{DPO}}$ whose objects are the direct transformations δ such that $\delta\mathsf{m}$ is monic, and let $\mathsf{R}_{\text{DPOm}} : \mathscr{D}_{\text{DPOm}} \to \mathscr{R}_{\text{DPOm}}$ be the corresponding restriction of R_{DPO}.

4 Rewriting Environments

Given an input object G and a category of rules, we are left with the problem of finding all relevant transformations of G by these rules. We cannot simply rely on the matchings of their left-hand sides in G (as in [16]) since they may not have pushout complements, or they may have several non isomorphic ones. We will therefore use the relevant direct transformations, albeit in an abbreviated version that do not contain L, since we don't use matchings, nor H since they are not relevant to subsumption.

Definition 4.1 (category $\mathscr{C}_{\text{pt}}$, functors In, P_{DPOm})**.** A *partial transformation* τ in $\mathscr{C}$ is a diagram

$$G \xleftarrow{\ f\ } D \xleftarrow{\ k\ } K \xrightarrow{\ r\ } R$$

For any category $\mathscr{C}$, let $\mathscr{C}_{\text{pt}}$ be the category whose objects are partial transformations and morphisms $\nu : \tau \to \tau'$ are triples (ν_1, ν_2, ν_3) such that

$$\begin{tikzcd} G & D \arrow[l, "f"'] & K \arrow[l, "k"'] \arrow[r, "r"] \arrow[d, "v_2"] & R \arrow[d, "v_3"] \\ G' \arrow[u, dash, "="] & D' \arrow[l, "f'"] \arrow[u, "v_1"] & K' \arrow[l, "k'"] \arrow[r, "r'"'] & R' \end{tikzcd}$$

commutes in $\mathscr{C}$, with obvious composition and identities.

Let $\mathsf{In} : \mathscr{C}_{\mathrm{pt}} \to |\mathscr{C}|$ be the *input functor* defined as $\mathsf{In}\,\tau = G$. Let $\mathsf{P}_{\mathrm{DPO}} : \mathscr{D}_{\mathrm{DPO}} \to \mathscr{C}_{\mathrm{pt}}$ and $\mathsf{P}_{\mathrm{DPOm}} : \mathscr{D}_{\mathrm{DPOm}} \to \mathscr{C}_{\mathrm{pt}}$ be the obvious functors (such that $(\mathsf{P}_{\mathrm{DPO}}\delta)\mathrm{G} = \delta\mathrm{G}$ etc. and $\mathsf{P}_{\mathrm{DPO}}\mu = (\mu_4, \mu_2, \mu_3)$).

Using inverse images along $\mathsf{P}_{\mathrm{DPOm}}$ and $\mathsf{R}_{\mathrm{DPOm}}$ we can easily focus on the direct transformations of concern (and the morphisms between them), i.e., the transformations *of* a graph *by* a rule.

Definition 4.2 (Rewriting Environments, rule systems, notations D_δ, $\pi_1\mu$ ...)**.** For any category $\mathscr{C}$, a *Rewriting Environment* for $\mathscr{C}$ consists of a category $\mathscr{D}$ of *direct transformations*, a category $\mathscr{R}$ of *rules* and two functors

$$\mathscr{R} \xleftarrow{\mathsf{R}} \mathscr{D} \xrightarrow{\mathsf{P}} \mathscr{C}_{\mathrm{pt}}$$

A *rule system* in a Rewriting Environment is a category $\mathscr{S}$ with an embedding $\mathsf{I} : \mathscr{S} \to \mathscr{R}$ (alternately, $\mathscr{S}$ is a subcategory of $\mathscr{R}$ and I is the inclusion functor).

Given a rule system and an *input* $\mathscr{C}$-object G, we build the categories $\mathscr{D}|_G$, $\mathscr{D}|_G^{\mathscr{S}}$ and functors I_G, $\mathsf{I}_{\mathscr{S}}$, $\mathsf{R}|_G^{\mathscr{S}}$ as meets of previous functors:

$$\begin{tikzcd} \mathscr{S} \arrow[r, tail, "\mathsf{I}"] & \mathscr{R} & & \\ & \mathscr{D} \arrow[u, "\mathsf{R}"] \arrow[r, "\mathsf{P}"] & \mathscr{C}_{\mathrm{pt}} \arrow[r, "\mathsf{In}"] & |\mathscr{C}| \\ \mathscr{D}|_G^{\mathscr{S}} \arrow[uu, "\mathsf{R}|_G^{\mathscr{S}}"] \arrow[r, tail, "\mathsf{I}_{\mathscr{S}}"'] \arrow[ur, phantom, "\urcorner"] & \mathscr{D}|_G \arrow[u, tail, "\mathsf{I}_G"] \arrow[rr] \arrow[urr, phantom, "\urcorner"] & & \mathbf{1} \arrow[u, tail, "G"'] \end{tikzcd}$$

For any $\delta \in \mathscr{D}|_G^{\mathscr{S}}$ we write D_δ for $(\mathsf{P}\mathsf{I}_G\mathsf{I}_{\mathscr{S}}\delta)\mathrm{D}$ and similarly f_δ etc. For any $\mu : \delta \to \delta'$ in $\mathscr{D}|_G^{\mathscr{S}}$ we write $\pi_1\mu$ for the first coordinate of $\mathsf{P}\mathsf{I}_G\mathsf{I}_{\mathscr{S}}\mu$ and similarly $\pi_2\mu$, $\pi_3\mu$.

Example 4.3. For $\mathscr{S}$ we take the subcategory $\begin{tikzcd} \rho \arrow[r, bend left, "\sigma^+"] \arrow[r, bend right, "\sigma^-"'] & \rho' \end{tikzcd}$ of $\mathscr{R}_{\mathrm{DPO}}$. To the matchings m'_1 and m'_2 of ρ' in G correspond two[1] transformations in $\mathscr{D}_{\mathrm{DPOm}}$ that will be denoted δ'_1 and δ'_2 (depicted on the right in Example 3.6). To the matchings m^+ and m^- of ρ in G correspond another two transformations denoted δ^+ and δ^- (on the left in Example 3.6). To each $i = 1, 2$ correspond one morphism $\mu_i^+ : \delta^+ \to \delta'_i$ such that $\mathsf{R}_{\mathrm{DPOm}}\mu_i^+ = \sigma^+$ and one morphism $\mu_i^- : \delta^- \to \delta'_i$ such that $\mathsf{R}_{\mathrm{DPOm}}\mu_i^- = \sigma^-$. Thus $\mathscr{D}_{\mathrm{DPOm}}|_G^{\mathscr{S}}$ is the following subcategory of $\mathscr{D}_{\mathrm{DPOm}}$.

$$\begin{tikzcd} & \delta'_1 & \\ \delta^+ \arrow[ur, "\mu_1^+"] \arrow[dr, "\mu_2^+"'] & & \delta^- \arrow[ul, "\mu_1^-"'] \arrow[dl, "\mu_2^-"] \\ & \delta'_2 & \end{tikzcd}$$

[1] We consider transformations only up to isomorphisms, see Footnote 2.

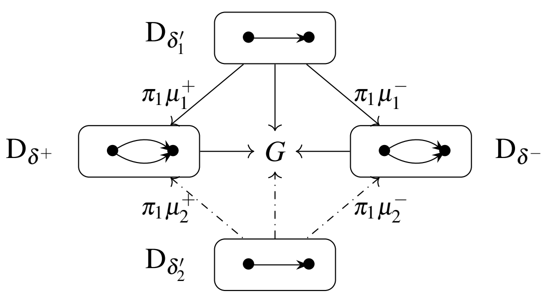
5 Global Coherent Transformations

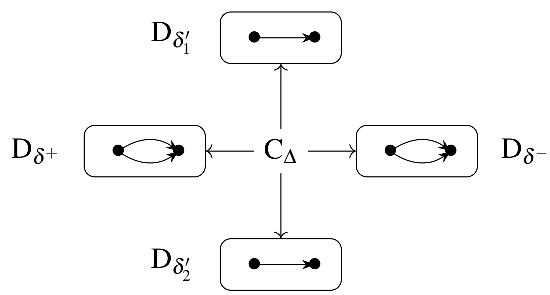
As stated above we will use the partial transformations that are accessible from $\mathscr{D}|_G^{\mathscr{S}}$ through $\mathsf{P} \circ \mathsf{I}_G \circ \mathsf{I}_{\mathscr{S}}$ (a restriction of P). We first need to build a context between the input G and the expected output H. In Parallel Coherent Transformation [2] the context is obtained as a limit of the morphisms $\mathrm{f}_\delta : \mathrm{D}_\delta \to G$ (that need not be monics) for all δ in a set Δ of direct transformations, hence of a diagram that is a sink to G and thus corresponds to a discrete diagram in $\mathscr{C} \setminus G$. In Global Coherent Transformations the *global context* (denoted C_Δ below) is obtained similarly, but now Δ is a category and the diagram contains the morphisms $\pi_1\mu : \mathrm{f}_{\delta'} \to \mathrm{f}_\delta$ for all $\mu : \delta \to \delta'$ in Δ (since $\mathrm{f}_\delta \circ \pi_1\mu = \mathrm{f}_{\delta'}$).

Definition 5.1 (functor $\mathsf{P}_\Delta^{\leftarrow}$, limit $f_\Delta : \mathrm{C}_\Delta \to G$, limit cone γ_Δ)**.** For any subcategory Δ of $\mathscr{D}|_G^{\mathscr{S}}$ let $\mathsf{P}_\Delta^{\leftarrow} : \Delta^{\mathrm{op}} \to \mathscr{C} \setminus G$ be the contravariant functor that maps every $\delta \in \Delta$ to $\mathrm{f}_\delta : \mathrm{D}_\delta \to G$ and every morphism μ of Δ to $\pi_1\mu : \mathrm{f}_{\delta'} \to \mathrm{f}_\delta$. Let $f_\Delta : \mathrm{C}_\Delta \to G$ be the limit of $\mathsf{P}_\Delta^{\leftarrow}$ and γ_Δ be the limit cone from f_Δ to $\mathsf{P}_\Delta^{\leftarrow}$.

Note that if Δ is empty then the limit f_Δ of the empty diagram is the terminal object of $\mathscr{C} \setminus G$, that is 1_G, hence $\mathrm{C}_\Delta = G$.

Example 5.2. Let $\Delta = \mathscr{D}_{\mathrm{DPOm}}|_G^{\mathscr{S}}$. The diagram on the left below corresponds to the functor $\mathsf{P}_\Delta^{\leftarrow}$ together with the morphisms $\mathrm{f}_{\delta_i^\pm} : \mathrm{D}_{\delta_i^\pm} \to G$ (objects in $\mathscr{C} \setminus G$). The limit of this diagram yields $\mathrm{C}_\Delta = \bullet \quad \bullet$ and the limit cone is represented on the right.



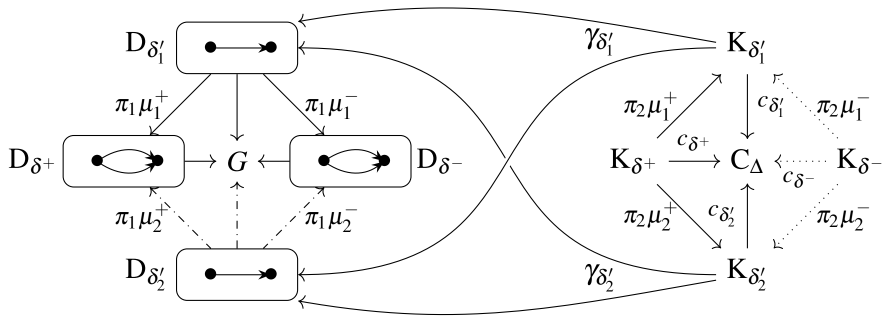


We next need to check that the transformations in Δ do not conflict with each other, i.e., that for all $\delta \in \Delta$ the image of K_δ in G is not only preserved by δ (in D_δ) but also by all other transformations $\delta' \in \Delta$. This is ensured by finding (natural) cones from these K_δ to the $\mathrm{D}_{\delta'}$, which we shall formulate with $\mathsf{P}_\Delta^{\leftarrow}$, hence in $\mathscr{C} \setminus G$.

Definition 5.3 (coherent system of cones, morphisms c_δ, global coherence)**.** A *coherent system of cones for* Δ is a set of cones γ_δ from $\mathrm{f}_\delta \circ \mathrm{k}_\delta$ to $\mathsf{P}_\Delta^{\leftarrow}$ such that $\gamma_\delta\delta = \mathrm{k}_\delta$ for all $\delta \in \Delta$, and $\gamma_\delta = \gamma_{\delta'} \circ \pi_2\mu$ for all $\mu : \delta \to \delta'$ in Δ. Δ is *globally coherent* if there exists a coherent system of cones for Δ. We then let $c_\delta : \mathrm{f}_\delta \circ \mathrm{k}_\delta \to f_\Delta$ be the unique morphism in $\mathscr{C} \setminus G$ such that $\gamma_\delta = \gamma_\Delta \circ c_\delta$.

Note that if $\gamma_{\delta'}$ is a cone from $\mathrm{f}_{\delta'} \circ \mathrm{k}_{\delta'}$ to $\mathsf{P}_\Delta^{\leftarrow}$ then $\gamma_{\delta'} \circ \pi_2\mu$ is a cone from $\mathrm{f}_\delta \circ \mathrm{k}_\delta$ to $\mathsf{P}_\Delta^{\leftarrow}$, hence global coherence means that we should find cones for overlapping direct transformations (say δ_1' and δ_2'), with the constraint that they should be compatible on their common subtransformations $\delta_1' \leftarrow \delta \to \delta_2'$. If $\mathscr{S}$ and therefore Δ are discrete, this amounts to parallel coherence (that generalizes parallel independence in DPO, see [2]).

Example 5.4. On our example the four graphs $\mathrm{K}_{\delta_i^\pm}$ are equal to $\bullet \quad \bullet$. It is easy to build the four cones from the four morphisms from $\mathrm{K}_{\delta_i'}$ to $\mathrm{D}_{\delta_i'}$ depicted below, by composing them with the $\pi_1\mu_i^\pm$ on the left and the $\pi_2\mu_i^\pm$ on the right. On the right are also depicted the morphisms $c_{\delta_i^\pm}$.



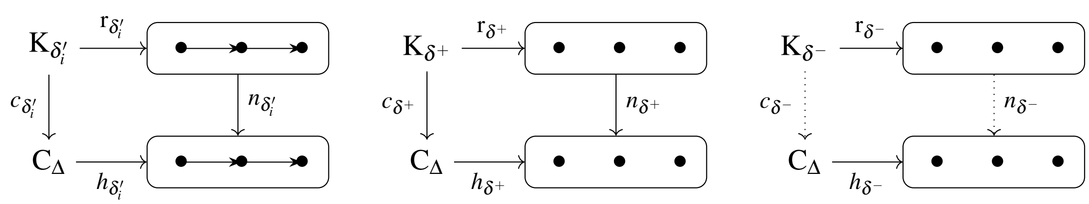
The reader may check that $\gamma_{\delta'_1} \circ \pi_2\mu_1^+ = \gamma_{\delta'_2} \circ \pi_2\mu_2^+$ (this is γ_{δ^+}) and $\gamma_{\delta'_1} \circ \pi_2\mu_1^- = \gamma_{\delta'_2} \circ \pi_2\mu_2^-$ ($= \gamma_{\delta^-}$).

The morphisms c_δ specify where the right-hand sides R_δ should be pushed in the global context.

Definition 5.5 (morphisms $h_\delta : C_\Delta \to H_\delta$)**.** If Δ is globally coherent for all $\delta \in \Delta$ then c_δ can be viewed as a $\mathscr{C}$-morphism $c_\delta : K_\delta \to C_\Delta$, and we consider the following pushout in $\mathscr{C}$.

$$\begin{tikzcd} K_\delta \arrow[r, "r_\delta"] \arrow[d, "c_\delta"'] & R_\delta \arrow[d, "n_\delta"] \\ C_\Delta \arrow[r, "h_\delta"'] & H_\delta \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

Example 5.6. On our example we get:



Thanks to the coherent system of cones we can turn h into a functor.

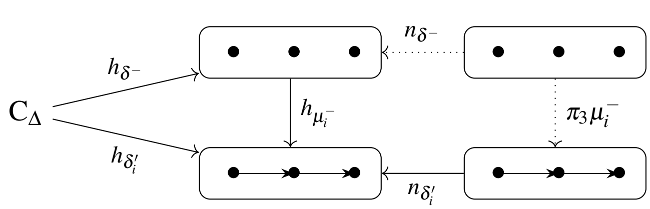
Proposition 5.7. *For every $\mu : \delta \to \delta'$ in Δ there exists a unique h_μ such that*

$$\begin{tikzcd} & H_\delta \arrow[d, dashed, "h_\mu"] & R_\delta \arrow[l, "n_\delta"'] \arrow[d, "\pi_3\mu"] \\ C_\Delta \arrow[ur, "h_\delta"] \arrow[dr, "h_{\delta'}"'] & H_{\delta'} & R_{\delta'} \arrow[l, "n_{\delta'}"] \\ & & \end{tikzcd}$$

commutes.

Corollary 5.8. *By unicity we get $h_{\mu'\circ\mu} = h_{\mu'} \circ h_\mu$.*

Example 5.9. For instance the morphisms $\mu_i^- : \delta^- \to \delta'_i$ yield the morphisms $h_{\mu_i^-}$ depicted below.

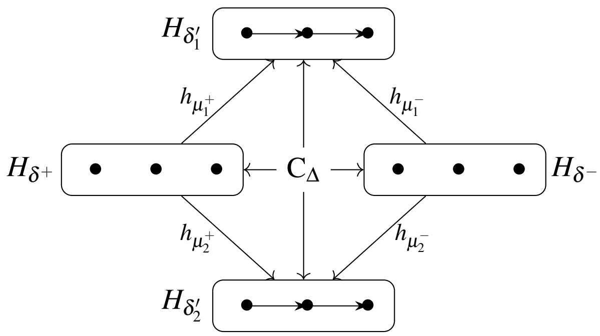


The final step of the Global Coherent Transformation, symmetric to the first step, consists in taking the colimit in the coslice category $\mathrm{C}_\Delta \backslash \mathscr{C}$ of the covariant diagram of index Δ with objects h_δ and morphisms $h_\mu : h_\delta \to h_{\delta'}$ for all $\mu : \delta \to \delta'$ in Δ.

Definition 5.10 (functor $\mathsf{P}^{\rightarrow}_{\Delta}$, colimit $h_\Delta : \mathrm{C}_\Delta \to H_\Delta$)**.** If Δ is globally coherent let $\mathsf{P}^{\rightarrow}_{\Delta} : \Delta \to \mathrm{C}_\Delta \backslash \mathscr{C}$ be the functor defined by $\mathsf{P}^{\rightarrow}_{\Delta}\delta = h_\delta$ (interpreted as an object of $\mathrm{C}_\Delta \backslash \mathscr{C}$) and $\mathsf{P}^{\rightarrow}_{\Delta}\mu = h_\mu$ for all $\mu : \delta \to \delta'$ in Δ. Let $h_\Delta : \mathrm{C}_\Delta \to H_\Delta$ be the colimit[2] of $\mathsf{P}^{\rightarrow}_{\Delta}$, then the $\mathscr{C}$-span $G \xleftarrow{f_\Delta} \mathrm{C}_\Delta \xrightarrow{h_\Delta} H_\Delta$ is a *Global Coherent Transformation by* Δ.

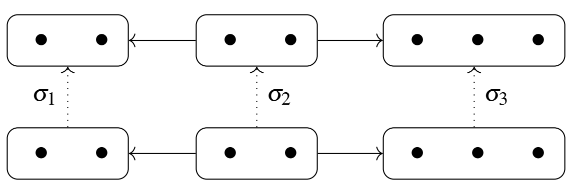
If Δ is empty then the colimit h_Δ of the empty diagram is the initial object of $\mathrm{C}_\Delta \backslash \mathscr{C}$, that is 1_{C_Δ}, hence $H_\Delta = \mathrm{C}_\Delta = G$. Generally, the functor $\mathsf{P}^{\rightarrow}_{\Delta}$ depends on the choice of cones γ_δ for $\delta \in \Delta$, hence h_Δ is not determined by Δ.

Example 5.11. The functor $\mathsf{P}^{\rightarrow}_{\Delta}$ applied to Δ yields the following diagram



The leftmost vertices of these five graphs are connected as images or preimages of each other, and similarly for the five right vertices, and the four middle vertices. The four edges are not likewise connected, hence the colimit of this diagram is the expected result $H =$  . We therefore see that the two middle vertices created in δ'_1 and δ'_2 are merged by their common subtransformation δ^+ (or δ^-), but also that the two middle vertices created in δ^+ and δ^- are merged by their common subsuming transformation δ'_1 (or δ'_2).

If we apply $\mathscr{S}$ to the graph $G' = \bullet \quad \bullet$ then rule ρ' does not apply to G' and hence the two matchings of ρ in G' apply independently, thus adding two vertices to G'. We can merge them by adding to $\mathscr{S}$ the following rule morphism $\sigma : \rho \to \rho$ that swaps the left and right vertices:



We have $\sigma^2 = 1_\rho$ hence σ is an automorphism of ρ. Adding σ to $\mathscr{S}$ means that the symmetric applications of ρ, i.e., direct transformations with matchings m and $m \circ \sigma$, shall be merged (this seems to generalize to the algebraic context the notion of Parallel Rewriting Modulo Automorphism devised in an algorithmic approach in [1]). Since $\sigma^+ \circ \sigma = \sigma^-$ and $\sigma^- \circ \sigma = \sigma^+$, the new rule system is

$$\mathscr{S}' = \begin{tikzcd} \rho \arrow[loop left, "\sigma"] \arrow[r, bend left, "\sigma^+"] \arrow[r, bend right, "\sigma^-"'] & \rho' \end{tikzcd}$$

[2] Global Coherent Transformations are obtained as limits and colimits of diagrams whose index category is Δ, hence are not affected by isomorphisms in Δ, which can therefore be replaced by its skeleton.

If we apply $\mathscr{S}'$ to G, we add two new morphisms in $\mathscr{D}_{\mathrm{DPOm}}|^{\mathscr{S}}_{G}$, i.e,

$$\Delta' = \mathscr{D}_{\mathrm{DPOm}}|^{\mathscr{S}'}_{G} = \begin{tikzcd} & \delta'_1 & \\ \delta^+ \arrow[ur, "\mu_1^+"] \arrow[dr, "\mu_2^+"'] \arrow[rr, bend left] & & \delta^- \arrow[ul, "\mu_1^-"'] \arrow[dl, "\mu_2^-"] \arrow[ll, bend left] \\ & \delta'_2 & \end{tikzcd}$$

It is easy to see that the Global Coherent Transformation by Δ' is the same as above with Δ. This is due to the fact that δ^+ and δ^- are already related in Δ through δ'_1 (or δ'_2).

We finally prove that, apart from this mechanism of sharing common subtransformations, isolated transformations always subsume their subtransformations, so that morphisms in $\mathscr{R}$ are rule subsumptions as intended.

Proposition 5.12. *If Δ' is restricted to δ' and Δ to $\mu : \delta \to \delta'$ (or more generally if δ' is terminal in Δ) then Δ and Δ' are globally coherent and $H_\Delta \simeq H_{\Delta'}$.*

6 Some Rewriting Environments and Their Properties

An obvious property of Rewriting Environments is that they can be combined: if $\mathscr{R}_1 \xleftarrow{\mathsf{R}_1} \mathscr{D}_1 \xrightarrow{\mathsf{P}_1} \mathscr{C}_{\mathrm{pt}}$ and $\mathscr{R}_2 \xleftarrow{\mathsf{R}_2} \mathscr{D}_2 \xrightarrow{\mathsf{P}_2} \mathscr{C}_{\mathrm{pt}}$ are Rewriting Environments for $\mathscr{C}$ then so is $\mathscr{R}_1 + \mathscr{R}_2 \xleftarrow{\mathsf{R}_1+\mathsf{R}_2} \mathscr{D}_1 + \mathscr{D}_2 \xrightarrow{[\mathsf{P}_1,\mathsf{P}_2]} \mathscr{C}_{\mathrm{pt}}$. It is therefore possible to mix rules of different approaches to transform a graph, though of course rules of distinct approaches cannot subsume each other.

A property that one might reasonably expect is that when a rule applies and yields a direct transformation then its subrules also apply and yield subtransformations. We express this by means of the following notion.

Definition 6.1 (right-full)**.** A functor $\mathsf{F} : \mathscr{A} \to \mathscr{B}$ is *right-full*[3] if for all $a' \in \mathscr{A}$, all $b \in \mathscr{B}$ and all $g : b \to Fa'$, there exist $a \in \mathscr{A}$ and $f : a \to a'$ such that $Ff = g$.

It is obvious that right-fullness is closed by composition.

Lemma 6.2. *I_G is a full and right-full embedding.*

Proposition 6.3. *If R is right-full (resp. faithful) then so is $\mathsf{R}|^{\mathscr{S}}_{G}$ for every rule system $\mathscr{S}$ and $G \in \mathscr{C}$.*

Hence when R is right-full and faithful every morphism $\sigma : \rho \to \rho'$ in $\mathscr{S}$ is reflected by a morphism in $\mathscr{D}|^{\mathscr{S}}_{G}$ whenever ρ' is reflected by a direct transformation δ' (i.e., whenever ρ' applies to G), and this morphism is uniquely determined by σ and δ'.

6.1 Double-Pushouts

Definitions 3.3, 3.8 and 4.1 provide two Rewriting Environments that we may call DPO and DPOm. By Proposition 3.7 it is obvious that $\mathsf{R}_{\mathrm{DPO}}$ and $\mathsf{R}_{\mathrm{DPOm}}$ are faithful when $\mathscr{C}$ is the category of graphs. This is easily seen to generalize to all adhesive categories [11]. Proposition 3.7 generalizes as follows:

Proposition 6.4. *If $\mathscr{C}$ is adhesive, $\delta, \delta' \in \mathscr{D}_{\mathrm{DPO}}$ and $\sigma : \mathsf{R}_{\mathrm{DPO}}\delta \to \mathsf{R}_{\mathrm{DPO}}\delta'$ such that $m = m' \circ \sigma_1$ then there exists a unique $\mu : \delta \to \delta'$ such that $\mathsf{R}_{\mathrm{DPO}}\mu = \sigma$.*

[3] This is named after the symmetric definition of *left-full* functors in [17, p. 63].

According to Proposition 3.5 it is obvious that $\mathsf{R}_{\mathrm{DPOm}}$ is right-full (when $\mathscr{C}$ is the category of graphs). It is easy to see that $\mathsf{R}_{\mathrm{DPO}}$ is not right-full (with σ_1 not monic, see Proposition 3.5).

One drawback with span rules is that every item matched by m that is not removed must be preserved in the result, hence cannot be removed by an overlapping rule, by the requirement of global coherence. In [2] we have defined *weak* DPO-rules by inserting a second interface I between K and L. A weak DPO transformation is a diagram

$$\begin{tikzcd}
L \arrow[d, "m"'] & I \arrow[l, tail, "l"'] \arrow[d, "k"'] & K \arrow[l, tail, "i"'] \arrow[r, "r"] \arrow[d, "k\circ i"'] & R \arrow[d, "n"] \\
G & D \arrow[l, "f"] \arrow[r, equal, "="] & D \arrow[r, "g"'] & H
\end{tikzcd}$$

so that the images of items in I are not removed by this transformation, but only images of items in K may not be removed by any simultaneous transformation. In cellular automata we need items in I that match the states of the neighbour cells, but there should be none in K since these states may be modified by overlapping rules (see [2, Example 3], note that K and I are swapped).

It is easy to define subsumption morphisms between weak DPO-rules (as 4-tuples of $\mathscr{C}$-morphisms with commuting properties and a pullback as in Definition 3.3), and corresponding morphisms between direct transformations of weak DPO-rules (as 5-tuples of $\mathscr{C}$-morphisms with commuting properties and a pullback as in Definition 3.8). This yields a Rewriting Environment for weak double-pushouts.

6.2 Sesqui-Pushouts

We now consider the case of Sesqui-Pushouts [5]. It is based on the notion of final pullback complement that allows not only to remove parts of the input G but also to make copies of parts of G.

Definition 6.5 (category $\mathscr{R}_{\mathrm{SqPO}}$, direct SqPO-transformations)**.** A *SqPO-rule* ρ in $\mathscr{C}$ is a span diagram $L \xleftarrow{l} K \xrightarrow{r} R$. Let $\mathscr{R}_{\mathrm{SqPO}}$ be the category whose objects are the SqPO-rules and morphisms $\sigma : \rho \to \rho'$ are triples $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ such that

$$\begin{tikzcd}
L \arrow[d, "\sigma_1"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "\sigma_2"] & R \arrow[d, "\sigma_3"] \\
L' & K' \arrow[l, "l'"] \arrow[r, "r'"'] & R'
\end{tikzcd}$$

commutes in $\mathscr{C}$ and the left square is a pullback, with obvious composition and identities. Let $\mathscr{R}_{\mathrm{SqPOm}}$ be the subcategory with morphisms σ such that σ_1 and σ_2 are monics.

A *final pullback complement* of (m, l) is a pair (f, k) such that (k, l) is a pullback of (f, m) and for every pullback $(k', l \circ c)$ of any (f', m) there exists a unique d such that

$$\begin{tikzcd}
 & L \arrow[dd, "m"] & & K \arrow[ll, "l"'] \arrow[dd, "k"] \\
L \arrow[ur, equal, "="] \arrow[dd, "m"'] & & K' \arrow[ll, "l\circ c"'] \arrow[ur, "c"'] \arrow[dd, "k'"'] & \\
 & G & & D \arrow[ll, "f"'] \\
G \arrow[ur, equal, "="] & & D' \arrow[ll, "f'"] \arrow[ur, dashed, "d"'] &
\end{tikzcd}$$

commutes.

A *direct SqPO-transformation* in $\mathscr{C}$ is a diagram

$$\begin{tikzcd}
L \arrow[d, "m"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "k"] \arrow[dl, phantom, "\lrcorner", very near start] & R \arrow[d, "n"] \\
G & D \arrow[l, "f"] \arrow[r, "g"'] & H \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

such that (f,k) is a final pullback complement of (m,l) and the right square is a pushout.

Proposition 6.6. *For every direct SqPO-transformations δ, δ' with corresponding SqPO-rules ρ, ρ', every $\sigma : \rho \to \rho'$ in $\mathscr{R}_{\mathrm{SqPO}}$ such that $m = m' \circ \sigma_1$, there exists a unique $\mathscr{C}$-morphism d such that*

$$\begin{tikzcd}
& L \arrow[dl, "\sigma_1"'] \arrow[dd, "m" near start] & & K \arrow[ll, "l"'] \arrow[dl, "\sigma_2"'] \arrow[dd, "k"] \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, "l'" near start, crossing over] & \\
& G \arrow[dl, dash, "="'] & & D \arrow[ll, "f"' near start] \\
G' & & D' \arrow[ll, "f'"] \arrow[ur, dashed, "d"'] \arrow[from=uu, "k'"' near end, crossing over] &
\end{tikzcd}$$

commutes.

Here the existence of d means not only that ρ' removes at least as much as its subrule ρ, but also that it makes at least as many copies of the items of G. Note that when, among two simultaneous transformations, one makes p copies of an item and the other makes q copies of the same item, the global context must contain pq copies of this item, *unless* there is a subsumption morphism between them. In such a case all the copies made by the subsumed transformation are simply merged with those made by the subsuming one (as witnessed by Proposition 5.12). Hence the necessary symmetry between the first and last steps of the Global Coherent Transformation.

It is then easy to define the category $\mathscr{D}_{\mathrm{SqPO}}$ of direct SqPO-transformations, the category $\mathscr{D}_{\mathrm{SqPOm}}$ of direct SqPO-transformations with monic matches and faithful functors $\mathsf{R}_{\mathrm{SqPO}} : \mathscr{D}_{\mathrm{SqPO}} \to \mathscr{R}_{\mathrm{SqPO}}$ and $\mathsf{R}_{\mathrm{SqPOm}} : \mathscr{D}_{\mathrm{SqPOm}} \to \mathscr{R}_{\mathrm{SqPOm}}$, as in Definition 3.8. We leave this to the reader.

Proposition 6.7. *In the category of graphs $\mathsf{R}_{\mathrm{SqPOm}}$ is right-full.*

Another notion of subrule in the Sesqui-Pushout approach can be found in [14, Definition 8], where a rule ρ' is defined as a (σ_1, σ_3)-extension of ρ if two conditions are met. The first is that $\sigma_3 \circ \rho = \rho' \circ \sigma_1$, where σ_1 stands for the span $L \xleftarrow{1_L} L \xrightarrow{\sigma_1} L'$ (and similarly for σ_3) and $\circ$ is the standard composition of spans (using pullbacks, see [14, Definition 3]). The products $\sigma_3 \circ \rho$, $\rho' \circ \sigma_1$ yield

$$\begin{tikzcd}
& & R & & \\
& K \arrow[ur, "r"] \arrow[dl, "l"'] & & R \arrow[ul, "1_R"'] \arrow[dr, "\sigma_3"] & \\
L & & K \arrow[ll] \arrow[rr] \arrow[ul, "1_K"] \arrow[ur, "r"'] \arrow[u, phantom, "\wedge"] & & R'
\end{tikzcd}
\qquad
\begin{tikzcd}
& & L' & & \\
& L \arrow[ur, "\sigma_1"] \arrow[dl, "1_L"'] & & K' \arrow[ul, "l'"'] \arrow[dr, "r'"] & \\
L & & K \arrow[ll] \arrow[rr] \arrow[ul] \arrow[ur, "\sigma_2"'] \arrow[u, phantom, "\wedge"] & & R'
\end{tikzcd}$$

hence the equality between these two bottom spans is equivalent to the existence of $(\sigma_1,\sigma_2,\sigma_3):\rho\to\rho'$, i.e. that the left square in Definition 6.5 is a pullback and the right square commutes. This means that any extension of a rule according to [14, Definition 8] subsumes this rule according to Definition 6.5. The converse is false since the extension requires a second condition, namely that (σ_1,l) has a final pullback complement. This ensures that the extension can be decomposed as a product of two spans [14, Proposition 9], but this is relevant to sequential rewriting and not to the present notion of subsumption.

6.3 Pullback-Pushouts

We next consider the case of PBPO-rules [4], that also enables copies of parts of G but with better control of the way they are linked together and to the rest of G. The drawback is that matchings of the left-hand side of a rule into G should be completed with a co-match from G to a given "type" of the left-hand side.

Definition 6.8 (category $\mathscr{D}_{\mathrm{PBPO}}$, direct PBPO-transformations)**.** A *PBPO-rule* ρ in $\mathscr{C}$ is a commuting diagram

$$\begin{tikzcd}
L \arrow[d, "t_L"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "t_K"] & \mathrm{R} \arrow[d, "t_R"] \\
T_L & T_K \arrow[l, "u"] \arrow[r, "v"'] & T_R
\end{tikzcd}$$

A morphism $\sigma:\rho\to\rho'$ is a 5-tuple $(\sigma_1,\sigma_2,\sigma_3,\sigma_4,\sigma_5)$ of $\mathscr{C}$-morphisms such that

$$\begin{tikzcd}
& L \arrow[ld, "\sigma_1"'] \arrow[dd, "t_L"] & & K \arrow[ll, "l"'] \arrow[rr, "r"] \arrow[ld, "\sigma_2"'] \arrow[dd, "t_K"] & & R \arrow[ld, "\sigma_3"] \\
L' \arrow[dd, "t_{L'}"'] & & K' \arrow[ll, "l'"'] \arrow[rr, "r'"] \arrow[dd, "t_{K'}"] & & R' & \\
& T_L & & T_K \arrow[ll, "u"] & & \\
T_{L'} \arrow[ru, "\sigma_4"] & & T_{K'} \arrow[ll, "u'"] \arrow[ru, "\sigma_5"'] & & &
\end{tikzcd}$$

commutes. Let $\mathscr{D}_{\mathrm{PBPO}}$ be the category of PBPO-rules on $\mathscr{C}$ and their morphisms, with obvious composition and identities.

A *direct PBPO-transformation* in $\mathscr{C}$ is a commuting diagram

$$\begin{tikzcd}
L \arrow[d, "m"] \arrow[dd, bend right, "t_L"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "k"] \arrow[dd, bend right, "t_K"'] & R \arrow[d, "n"] \arrow[dd, bend right, "t_R"'] \\
G \arrow[d, "t_G"] & D \arrow[l, "f"] \arrow[r, "g"'] \arrow[d, "t_D"] \arrow[dl, phantom, "\llcorner", very near start] & H \arrow[d, "t_H"] \arrow[ul, phantom, "\ulcorner", very near start] \\
T_L & T_K \arrow[l, "u"] \arrow[r, "v"'] & T_R
\end{tikzcd}$$

with lower left pullback and upper right pushout.

To every direct PBPO-transformation obviously corresponds a PBPO-rule and a partial transformation.

Proposition 6.9. *For every direct PBPO-transformations δ, δ' with corresponding PBPO-rules ρ, ρ', every $\sigma : \rho \to \rho'$ in $\mathscr{D}_{\mathrm{PBPO}}$ such that $m = m' \circ \sigma_1$ and $t_G = \sigma_4 \circ t_{G'}$, there exists a unique $\mathscr{C}$-morphism d such that*

$$\begin{tikzcd}
 & L \arrow[dl, "\sigma_1"'] \arrow[dd, "m"] & & K \arrow[ll, "l"'] \arrow[dl, "\sigma_2"'] \arrow[dd, "k"] \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, "l'"'] \arrow[dd, "k'"] & \\
 & G \arrow[dl, dash, "="'] \arrow[dd, "t_G"] & & D \arrow[ll, "f"'] \arrow[dd, "t_D"] \\
G' \arrow[dd, "t_{G'}"'] & & D' \arrow[ll, "f'"] \arrow[ur, dashed, "d"'] \arrow[dd, "t_{D'}"] & \\
 & T_L & & T_K \arrow[ll, "u"] \\
T_{L'} \arrow[ur, "\sigma_4"] & & T_{K'} \arrow[ll, "u'"] \arrow[ur, "\sigma_5"'] &
\end{tikzcd}$$

commutes.

We leave it to the reader to define a Rewriting Environment for PBPO-rules and transformations, with a right-full faithful functor $\mathsf{R}_{\mathrm{PBPO}} : \mathscr{D}_{\mathrm{PBPO}} \to \mathscr{R}_{\mathrm{PBPO}}$ (provided $\mathscr{C}$ has pushouts and pullbacks).

7 Conclusion and Future Work

Global Coherent Transformations are built from partial transformations in a way pertaining both to Parallel Coherent Transformations [2], by the use of limits on local contexts, and to Global Transformations [16] by applying categories of rules. The partial transformations involved in a Global Coherent Transformation are extracted from a Rewriting Environment that provide a category of rules and a corresponding category of direct transformations. Their morphisms can be understood as subsumptions due to Property 5.12, i.e., that any subsumed transformation as defined by a morphism removes or adds nothing more than the subsuming transformation. This is valid even when rules are able to make multiple copies of parts of the input.

We have provided Rewriting Environments for the most common approaches to algebraic rewriting, except the Single Pushout [13], which will be done in a future paper (where we will see that the interface and right-hand side provided in a partial transformation are not necessarily those of the applied rule). We also intend to show that Global Transformations can be obtained as Global Coherent Transformations in a suitable environment (except when Δ is empty). Expressiveness of Global Coherent Transformations should be investigated further, and possibly enhanced.

The notion of Rewriting Environment is as simple as required to define Global Coherent Transformations, but does not guarantee some properties that the user might reasonably expect. In particular it does not prevent the categories $\mathscr{R}$ and $\mathscr{D}$ from being discrete. Of course this is correct if no subsumption is possible, but is there a way to characterize such properties? It may also seem strange that, through $\mathscr{C}_{\mathrm{pt}}$, rules are not assumed to have left-hand sides and direct transformations are not assumed to use matchings. Thus we may need to enhance Rewriting Environments with a notion of matching in order to better understand their structure. We also need to further analyze the properties of the Rewriting Environments in Section 6: when $\mathscr{C}$ is an adhesive category it is an open question whether $\mathsf{R}_{\mathrm{DPOm}}$ is right-full.

Acknowledgements We thank Rachid Echahed for helpful discussions and an anonymous reviewer in particular for suggesting the generalization of Proposition 5.12.

References

[1] T. Boy de la Tour & R. Echahed (2020): *Parallel Rewriting of Attributed Graphs*. *Theoretical Computer Science* 848, pp. 106–132, doi:10.1016/j.tcs.2020.09.025.

[2] T. Boy de la Tour & R. Echahed (2021): *Parallel Coherent Graph Transformations*. In: *Recent Trends in Algebraic Development Techniques, 25th International Workshop, WADT 2020, Revised Selected Papers*, *LNCS* 12669, Springer, pp. 75–97, doi:10.1007/978-3-030-73785-6_5.

[3] Thierry Boy de la Tour (2023): *Algebraic Monograph Transformations*, doi:10.48550/ARXIV.2303.01137.

[4] Andrea Corradini, Dominique Duval, Rachid Echahed, Frédéric Prost & Leila Ribeiro (2019): *The PBPO graph transformation approach*. *J. Log. Algebr. Meth. Program.* 103, pp. 213–231, doi:10.1016/j.jlamp.2018.12.003.

[5] Andrea Corradini, Tobias Heindel, Frank Hermann & Barbara König (2006): *Sesqui-Pushout Rewriting*. In: *ICGT 2006*, *LNCS* 4178, Springer, pp. 30–45, doi:10.1007/11841883_4.

[6] Andrea Corradini, Ugo Montanari, Francesca Rossi, Hartmut Ehrig, Reiko Heckel & Michael Löwe (1997): *Algebraic Approaches to Graph Transformation - Part I: Basic Concepts and Double Pushout Approach*. In Grzegorz Rozenberg, editor: *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*, World Scientific, pp. 163–246, doi:10.1142/9789812384720_0003.

[7] Hartmut Ehrig (1979): *Introduction to the algebraic theory of graph grammars (a survey)*. In Volker Claus, Hartmut Ehrig & Grzegorz Rozenberg, editors: *Graph-Grammars and Their Application to Computer Science and Biology*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–69, doi:10.1007/BFb0025714.

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

[9] Alexandre Fernandez, Luidnel Maignan & Antoine Spicher (2021): *Cellular Automata and Kan Extensions*. In Alonso Castillo-Ramirez, Pierre Guillon & Kévin Perrot, editors: *27th IFIP WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems, AUTOMATA 2021*, *OASIcs* 90, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 7:1–7:12, doi:10.4230/OASIcs.AUTOMATA.2021.7.

[10] Alexandre Fernandez, Luidnel Maignan & Antoine Spicher (2022): *Non-Determinism in Lindenmayer Systems and Global Transformations*. In Stefan Szeider, Robert Ganian & Alexandra Silva, editors: *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022*, *LIPIcs* 241, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 49:1–49:13, doi:10.4230/LIPIcs.MFCS.2022.49.

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

[12] F. W. Lawvere (1963): *Functorial Semantics of Algebraic Theories*. Ph.D. thesis, Columbia University, doi:10.1073/pnas.50.5.869. Available at `http://www.tac.mta.ca/tac/reprints/articles/5/tr5abs.html`.

[13] Michael Löwe (1993): *Algebraic Approach to Single-Pushout Graph Transformation*. *Theoretical Computer Science* 109, pp. 181–224, doi:10.1016/0304-3975(93)90068-5.

[14] Michael Löwe (2015): *Polymorphic Sesqui-Pushout Graph Rewriting*. In Francesco Parisi-Presicce & Bernhard Westfechtel, editors: *Graph Transformation - 8th International Conference, ICGT 2015*, *Lecture Notes in Computer Science* 9151, Springer, pp. 3–18, doi:10.1007/978-3-319-21145-9_1.

[15] Saunders Mac Lane (1997): *Categories for the Working Mathematician*, 2nd edition. *Graduate Texts in Mathematics* 5, Springer-Verlag, New York, doi:10.1007/978-1-4757-4721-8. (1st ed., 1971).

[16] Luidnel Maignan & Antoine Spicher (2015): *Global Graph Transformations*. In Detlef Plump, editor: *Proceedings of the 6th International Workshop on Graph Computation Models*, *CEUR Workshop Proceedings* 1403, CEUR-WS.org, pp. 34–49. Available at `http://ceur-ws.org/Vol-1403/paper4.pdf`.

[17] Peter Selinger (1997): *Functionality, Polymorphism, and Concurrency: a Mathematical Investigation of Programming Paradigms*. Ph.D. thesis, University of Pennsylvania. Technical Report No. IRCS-97-17.

Appendix: Proofs

Proof of Proposition 3.5. If $\sigma : \rho \to \rho'$ in $\mathscr{R}_{\mathrm{DPO}}$ such that σ_1 is monic and $m' : L' \to G$ satisfies the gluing condition for ρ' then so does $m' \circ \sigma_1 : L \to G$ for ρ.

We use the fact that the pullback K of l', σ_1 is isomorphic to en equalizer in $L \times K'$.

(GC1) Let x be an item in L such that $m' \circ \sigma_1(x)$ is marked for removal for ρ, hence such that x has no preimage by l, and let x' in L such that $m' \circ \sigma_1(x) = m' \circ \sigma_1(x')$. If $\sigma_1(x)$ had a preimage y by l' then x and y would have a common preimage in the pullback K, a contradiction. Hence $\sigma_1(x)$ has no preimage by l' so that $m'(\sigma_1(x))$ is marked for removal by m', hence $\sigma_1(x) = \sigma_1(x')$ by the (GC1) for m', ρ', hence $x = x'$.

(GC2) Let v be a vertex of L that has no preimage by l and is adjacent to an edge e in L, then as above $\sigma_1(v)$ has no preimage by l'. If e had a preimage e' by l then $l' \circ \sigma_2(e') = \sigma_1 \circ l(e') = \sigma_1(e)$, i.e., $\sigma_1(e)$ would have a preimage by l' in contradiction with (GC2) for m', ρ'. Hence $m' \circ \sigma_1(e)$ is marked for removal by $m' \circ \sigma_1$ for ρ'.

□

Proof of Proposition 3.7. If $\sigma : \rho \to \rho'$ in $\mathscr{R}_{\mathrm{DPO}}$, $m' : L' \to G$ and $m' \circ \sigma_1 : L \to G$ have pushout complements as below, then there is a unique d such that

$$
\begin{tikzcd}
 & L \arrow[dl, "\sigma_1"'] \arrow[dd] & & K \arrow[ll, tail, "l"'] \arrow[dl, "\sigma_2"] \arrow[dd, "k"] \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, tail, "l'"'] \arrow[dd, "k'"'] & \\
 & G \arrow[dl, equal, "="'] & & D \arrow[ll, tail, "f"'] \arrow[dl, dashed, tail, "d"] \\
G & & D' \arrow[ll, tail, "f'"] &
\end{tikzcd}
$$

commutes.

The front and back faces are pushouts. For all item x in D', $f'(x)$ is not marked for removal by m' and we show that is also the case by $m' \circ \sigma_1$. Suppose otherwise, then $f'(x)$ has a preimage y by $m' \circ \sigma_1$ that has no preimage by l. However, $\sigma_1(y)$ has a preimage y' by l', and since the top face is a pullback there should be a common preimage of y and y' in K, a contradiction. Thus we let $d(x)$ be the unique preimage of $f'(x)$ by f, so that d is unique such that $f \circ d = f'$. We easily see that $f \circ k = f \circ d \circ k' \circ \sigma_2$ hence the right face of the cube commutes. □

Proof of Proposition 5.7. For every $\mu : \delta \to \delta'$ in Δ there exists a unique h_μ such that

$$
\begin{tikzcd}
 & H_\delta \arrow[dd, dashed, "h_\mu"] & \mathrm{R}_\delta \arrow[l, "n_\delta"'] \arrow[dd, "\pi_3\mu"] \\
\mathrm{C}_\Delta \arrow[ur, "h_\delta"] \arrow[dr, "h_{\delta'}"'] & & \\
 & H_{\delta'} & \mathrm{R}_{\delta'} \arrow[l, "n_{\delta'}"]
\end{tikzcd}
$$

commutes.

Since $\gamma_\Delta \circ c_\delta = \gamma_\delta = \gamma_{\delta'} \circ \pi_2\mu = \gamma_\Delta \circ c_{\delta'} \circ \pi_2\mu$ then by the unicity of c_δ the left face of the following cube commutes.

$$\begin{tikzcd}
& \mathrm{K}_\delta \arrow[rr, "\mathrm{r}_\delta"] \arrow[dl, "\pi_2\mu"'] \arrow[dd, "c_\delta"] & & \mathrm{R}_\delta \arrow[dl, "\pi_3\mu"'] \arrow[dd, "n_\delta"] \\
\mathrm{K}_{\delta'} \arrow[rr, "\mathrm{r}_{\delta'}"] \arrow[dd, "c_{\delta'}"'] & & \mathrm{R}_{\delta'} \arrow[dd, "n_{\delta'}"] & \\
& \mathrm{C}_\Delta \arrow[rr, "h_\delta"] \arrow[dl, dash, "="'] & & H_\delta \arrow[dl, dashed, "h_\mu"] \\
\mathrm{C}_\Delta \arrow[rr, "h_{\delta'}"'] & & H_{\delta'} &
\end{tikzcd}$$

Since the top and front faces also commute then $n_{\delta'} \circ \pi_3\mu \circ \mathrm{r}_\delta = h_{\delta'} \circ c_\delta$, and since the back face is a pushout we get the result. □

Proof of Proposition 5.12. If Δ' is restricted to δ' and δ' is terminal in Δ then Δ and Δ' are globally coherent and $H_\Delta \simeq H_{\Delta'}$.

For any $\delta \in \Delta$ let $\delta!$ be the unique morphism $\delta! : \delta \to \delta'$. Since $(\pi_1\delta!, \pi_2\delta!, \pi_3\delta!) : \mathsf{PI}_G\mathsf{I}_{\mathscr{S}}\delta \to \mathsf{PI}_G\mathsf{I}_{\mathscr{S}}\delta'$ is a morphism in $\mathscr{C}_{\mathrm{pt}}$, then $\mathrm{f}_\delta \circ \pi_1\delta! = \mathrm{f}_{\delta'}$ and hence $\pi_1\delta! : \mathrm{f}_{\delta'} \to \mathrm{f}_\delta$ is a morphism in $\mathscr{C} \setminus G$.

Since δ' is initial in Δ^{op} there is a unique cone γ_Δ from $\mathsf{P}^{\leftarrow}_\Delta\delta' = \mathrm{f}_{\delta'}$ to $\mathsf{P}^{\leftarrow}_\Delta$ (defined by $\gamma_\Delta\delta = \pi_1\delta!$ for all $\delta \in \Delta$) and any cone γ from any $f \in \mathscr{C} \setminus G$ to $\mathsf{P}^{\leftarrow}_\Delta$ can be written $\gamma = \gamma_\Delta \circ \gamma\delta'$, hence γ_Δ is a limit cone of $\mathsf{P}^{\leftarrow}_\Delta$ (see [15, Exercise III.4.3]), so that $f_\Delta \simeq \mathrm{f}_{\delta'}$ and $\mathrm{C}_\Delta \simeq \mathrm{D}_{\delta'}$.

Let $\gamma_\delta = \gamma_\Delta \circ \mathrm{k}_{\delta'} \circ \pi_2\delta!$ (where $\pi_2\delta! : \mathrm{f}_\delta \circ \mathrm{k}_\delta \to \mathrm{f}_{\delta'} \circ \mathrm{k}_{\delta'}$ and $\mathrm{k}_{\delta'} : \mathrm{f}_{\delta'} \circ \mathrm{k}_{\delta'} \to \mathrm{f}_{\delta'}$ are morphisms in $\mathscr{C} \setminus G$ as above), this is a cone from $\mathrm{f}_\delta \circ \mathrm{k}_\delta$ to $\mathsf{P}^{\leftarrow}_\Delta$ such that $\gamma_\delta\delta = \pi_1\delta! \circ \mathrm{k}_{\delta'} \circ \pi_2\delta! = \mathrm{k}_\delta$. Besides, for every $\mu : \delta_1 \to \delta_2$ we have $\gamma_{\delta_1} = \gamma_{\delta_2} \circ \pi_2\mu$ since $\delta_2! \circ \mu = \delta_1!$. Hence $(\gamma_\delta)_{\delta\in\Delta}$ is a coherent system of cones for Δ, which is therefore globally coherent.

Since δ' is terminal in Δ there is as above a colimit cone from $\mathsf{P}^{\rightarrow}_\Delta$ to $\mathsf{P}^{\rightarrow}_\Delta\delta' = h_{\delta'} : \mathrm{C}_\Delta \to H_{\delta'}$, hence $H_\Delta \simeq H_{\delta'}$ (the pushout of $\mathrm{r}_{\delta'}$ and $c_{\delta'} = \mathrm{k}_{\delta'} \circ \pi_2\delta'! = \mathrm{k}_{\delta'}$). We finally note that δ' is terminal in Δ'. □

Proof of Lemma 6.2. I_G is a full and right-full embedding.

The functor $G : \mathbf{1} \to |\mathscr{C}|$ is a full embedding hence so is I_G. For all $\delta' \in \mathscr{D}|_G$, $\delta \in \mathscr{D}$ and $\mu : \delta \to \mathsf{I}_G\delta'$ we have $\mathsf{InP}\delta = \mathsf{I}_G\mathsf{InP}\delta' = G$ hence $\mathsf{InP}\mu = 1_G$. Since G and 1_G also have preimages by functor G there must be preimages $\delta'_1 \in \mathscr{D}|_G$ and $\mu_1 : \delta'_1 \to \delta'$ in $\mathscr{D}|_G$ such that $\mathsf{I}_G\mu_1 = \mu$, hence I_G is right-full. □

Proof of Proposition 6.3. If R is right-full (resp. faithful) then so is $\mathsf{R}|^{\mathscr{S}}_G$.

For all $\delta' \in \mathscr{D}|^{\mathscr{S}}_G$, $\rho \in \mathscr{S}$ and $\sigma : \rho \to \rho'$, where $\rho' = \mathsf{R}|^{\mathscr{S}}_G\delta'$, we have $\mathsf{I}\rho' = \mathsf{RI}_G\mathsf{I}_{\mathscr{S}}\delta'$ and $\mathsf{I}\sigma : \mathsf{I}\rho \to \mathsf{I}\rho'$ in $\mathscr{R}$, and since by Lemma 6.2 $\mathsf{R} \circ \mathsf{I}_G$ is right-full then there exists $\delta'_1 \in \mathscr{D}|_G$ and $\mu_1 : \delta'_1 \to \mathsf{I}_{\mathscr{S}}\delta'$ such that $\mathsf{RI}_G\mu_1 = \mathsf{I}\sigma$. Thus $\mathsf{I}\rho$ and $\mathsf{I}\sigma$ have preimages by I and $\mathsf{R} \circ \mathsf{I}_G$, hence they must have preimages $\delta \in \mathscr{D}|^{\mathscr{S}}_G$ and $\mu : \delta \to \delta'$ such that $\mathsf{I}_G\mu = \mu_1$ and $\mathsf{R}|^{\mathscr{S}}_G\mu = \sigma$.

If R is faithful, since I_G is faithful then so is $\mathsf{R} \circ \mathsf{I}_G$, and hence so is $\mathsf{R}|^{\mathscr{S}}_G$. □

Proof of Proposition 6.4. If $\mathscr{C}$ is adhesive, $\delta, \delta' \in \mathscr{D}_{\mathrm{DPO}}$ and $\sigma : \mathsf{R}_{\mathrm{DPO}}\delta \to \mathsf{R}_{\mathrm{DPO}}\delta'$ such that $m = m' \circ \sigma_1$ then there exists a unique $\mu : \delta \to \delta'$ such that $\mathsf{R}_{\mathrm{DPO}}\mu = \sigma$.

Let $G = \mathsf{InP}\delta = \mathsf{InP}\delta'$, we consider the following diagram

$$\begin{tikzcd}
 & L \arrow[dl, "\sigma_1"'] \arrow[dd, "m"] & & K \arrow[ll, tail, "l"'] \arrow[dl, "\sigma_2"] \arrow[dd, "k"] \arrow[ddr, dashed, bend left, "z"] & \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, tail, "l'"'] \arrow[dd, "k'"] & & \\
 & G \arrow[dl, dash, "="'] & & D \arrow[ll, tail, "f"'] & \\
G & & D' \arrow[ll, tail, "f'"] & & P \arrow[ll, tail, "x"] \arrow[ul, tail, "y"']
\end{tikzcd}$$

where the bottom face is a pullback. By [11, Lemma 4.2] monics are stable under pushouts hence f and f' are monics and therefore also x and y. By the commuting properties we have $f \circ k = f' \circ k' \circ \sigma_2$, hence there exists a unique z such that $y \circ z = k$ and $x \circ z = k' \circ \sigma_2$.

The front face is a pushout along the monic l, hence it is a pullback [11, Lemma 4.3], as is the top face, hence by composition the square formed by l, m, f', $k' \circ \sigma_2$ is also a pullback.

The back face is a pushout along the monic l, hence it is a VK-square and bottom face of the commuting cube below

$$\begin{tikzcd}
 & K \arrow[rr, "z"] \arrow[dl, "1"'] \arrow[dd, "1"] & & P \arrow[dl, tail, "x"] \arrow[dd, tail, "y"] \\
K \arrow[rr, "k' \circ \sigma_2"] \arrow[dd, tail, "l"'] & & D' \arrow[dd, "f'"] & \\
 & K \arrow[rr, "k"] \arrow[dl, tail, "l"'] & & D \arrow[dl, tail, "f"] \\
L \arrow[rr, "m"'] & & G &
\end{tikzcd}$$

Its front and right faces are pullbacks. Since l is monic then its left face is a pullback, and since y is monic its back face is also a pullback. Hence its top face is a pushout, and since isomorphisms are preserved by pushouts, x is an isomorphism.

Let $d = y \circ x^{-1}$, we see that $f \circ d = f'$ and $d \circ k' \circ \sigma_2 = y \circ z = k$, so that $\mu = (\sigma_1, \sigma_2, \sigma_3, d)$ is a morphism from δ to δ' in $\mathscr{D}_{\mathrm{DPO}}$ such that $\mathsf{R}_{\mathrm{DPO}}\mu = \sigma$. Its unicity is obvious. ☐

Proof of Proposition 6.6. For every direct SqPO-transformations δ, δ' with corresponding SqPO-rules ρ, ρ', every $\sigma : \rho \to \rho'$ in $\mathscr{R}_{\mathrm{SqPO}}$ such that $m = m' \circ \sigma_1$, there exists a unique d such that

$$\begin{tikzcd}
 & L \arrow[dl, "\sigma_1"'] \arrow[dd, "m"] & & K \arrow[ll, "l"'] \arrow[dl, "\sigma_2"] \arrow[dd, "k"] \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, "l'"'] \arrow[dd, "k'"] & \\
 & G \arrow[dl, dash, "="'] & & D \arrow[ll, "f"'] \\
G' & & D' \arrow[ll, "f'"] \arrow[ur, dashed, "d"'] &
\end{tikzcd}$$

commutes.

By composition of pullbacks $(k' \circ \sigma_2, l)$ is a pullback of (f', m), and since (f, k) is a final pullback complement of (m, l) then there is a unique $d : D' \to D$ such that $f' = f \circ d$ and $k = d \circ k' \circ \sigma_2$. ☐

Proof of Proposition 6.7. In the category of graphs $\mathsf{R}_{\mathrm{SqPOm}}$ *is right-full.*

For all $\delta' \in \mathscr{D}_{\mathrm{SqPOm}}$ and $\sigma : \rho \to \mathsf{R}_{\mathrm{SqPOm}}\delta'$ in $\mathscr{R}_{\mathrm{SqPOm}}$, the matching $m' \circ \sigma_1 : L \to G$ is monic hence by [5, Construction 6] $(m' \circ \sigma_1, l)$ has a final pullback complement, hence there is a $\delta \in \mathscr{D}_{\mathrm{SqPOm}}$ with $m = m' \circ \sigma_1$ and $\mathsf{R}_{\mathrm{SqPOm}}\delta = \rho$, and by Proposition 6.6 there is a (unique) $\mu : \delta \to \delta'$ in $\mathscr{D}_{\mathrm{SqPOm}}$ such that $\mathsf{R}_{\mathrm{SqPOm}}\mu = \sigma$. □

Proof of Proposition 6.9. For every direct PBPO-transformations δ, δ' *with corresponding PBPO-rules* ρ, ρ', *every* $\sigma : \rho \to \rho'$ *in* $\mathscr{D}_{\mathrm{PBPO}}$ *such that* $m = m' \circ \sigma_1$ *and* $t_G = \sigma_4 \circ t_{G'}$, *there exists a unique* d *such that*

$$
\begin{tikzcd}
 & L \arrow[dl, "\sigma_1"'] \arrow[dd, "m"] & & K \arrow[ll, "l"'] \arrow[dl, "\sigma_2"] \arrow[dd, "k"] \\
L' \arrow[dd, "m'"'] & & K' \arrow[ll, "l'"] \arrow[dd, "k'"] & \\
 & G \arrow[dl, equal] \arrow[dd, "t_G"] & & D \arrow[ll, "f"'] \arrow[dd, "t_D"] \\
G' \arrow[dd, "t_{G'}"'] & & D' \arrow[ll, "f'"] \arrow[ur, dashed, "d"'] \arrow[dd, "t_{D'}"] & \\
 & T_L & & T_K \arrow[ll, "u"] \\
T_{L'} \arrow[ur, "\sigma_4"] & & T_{K'} \arrow[ll, "u'"] \arrow[ur, "\sigma_5"'] &
\end{tikzcd}
$$

commutes.

By hypothesis the two front, back, left faces commute, as well as the top and bottom faces. Thus

$$u \circ \sigma_5 \circ t_{D'} = \sigma_4 \circ u' \circ t_{D'} = \sigma_4 \circ t_{G'} \circ f' = t_G \circ f',$$

and since D is a pullback then there exists a unique d such that the right and top face of the bottom cube commute. This also means that (D, f, t_D) is a mono-source, and since

$$\begin{cases} f \circ d \circ k' \circ \sigma_2 = f' \circ k' \circ \sigma_2 = m' \circ l' \circ \sigma_2 = m' \circ \sigma_1 \circ l = m \circ l = f \circ k \\ t_D \circ d \circ k' \circ \sigma_2 = \sigma_5 \circ t_{D'} \circ k' \circ \sigma_2 = \sigma_5 \circ t_{K'} \circ \sigma_2 = t_K = t_D \circ k \end{cases}$$

then $d \circ k' \circ \sigma_2 = k$. □

Collages of String Diagrams

Dylan Braithwaite
University of Strathclyde
dylan.braithwaite@strath.ac.uk

Mario Román
Tallinn University of Technology
mroman@ttu.ee

We introduce collages of string diagrams as a diagrammatic syntax for gluing multiple monoidal categories. Collages of string diagrams are interpreted as pointed bimodular profunctors. As the main examples of this technique, we introduce string diagrams for bimodular categories, string diagrams for functor boxes, and string diagrams for internal diagrams.

1 Introduction

String diagrams are a convenient and intuitive, sound and complete syntax for monoidal categories [26]. Monoidal categories are algebras of processes composing in parallel and sequentially [30]; string diagrams formalize the process diagrams of engineering [5, 7]. Formalization is not only of conceptual interest: it means we can sharpen our reasoning, scale our diagrams, or explain them to a computer [37].

However, the formal syntax of monoidal categories is not enough for all applications and, sometimes, we need to extend it. Functor boxes allow us to reason about translations between theories of processes [13, 32], ownership [34], higher-order processes [1], or programming effects [38]. Quantum combs not only model some classes of supermaps [10, 14, 21], but they coincide with the monoidal lenses of functional programming [4, 11, 45] and compositional game theory [20, 6]. Premonoidal categories, which appear in Moggi's semantics of programming effects [33, 27, 46], are now within the realm of string diagrammatic reasoning [42]. Internal diagrams extend the syntax of monoidal categories allowing us to draw diagrams inside tubular cobordisms and reason about topological quantum field theories [3], but also coends [41] and traces [24].

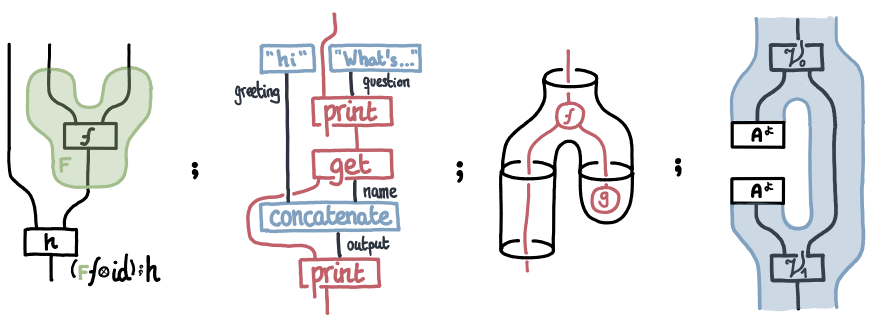


Figure 1: Examples from the literature. From left to right: functor boxes [32], premonoidal categories [42], internal diagrams [3], and combs or optics [10, 11, 21].

The extensions showcase the expressive power of string diagrams on surprisingly diverse application domains. At the same time, these different ideas could be regarded as separate ad-hoc extensions: they belong to different fields; they use different categorical formalisms. The overhead of learning and combining each one of them prevents the exchange of ideas between the different domains of application: e.g. an idea about topological quantum field diagrams does not transfer to premonoidal diagrams.

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 39–53, doi:10.4204/EPTCS.397.3

Collages. This manuscript claims that this division is only apparent and that all these extensions are particular instances of the same encompassing idea: that of glueing multiple string diagrams into what we call a *collage of string diagrams*. We introduce a formal notion of collage (Section 4.4) and employ string diagrammatic syntaxes for them, based on the calculus of bicategories (Sections 2.1, 3.1 and 5).

Even though collages of string diagrams are our novel contribution, collages are not yet another new concept to category theory. "Collage" was Bob Walters' term for a lax colimit in a module-like category [47]. This can be considered as a glueing of objects together along the action of a scalar. For example, given two sets A and B, with an action of a monoid M, we can construct their tensor product $A \otimes_M B$, where $(a \cdot m) \otimes b = a \otimes (m \cdot b)$ for any scalar $m \in M$. Categorifying this idea in a possible direction we obtain monoidal categories acting on *bimodular categories*. The following is the takeaway of this work.

> Collages of string diagrams consist of multiple string diagrams of different monoidal categories glued together. Collages can be interpreted as *pointed bimodular profunctors* between *bimodular categories*.

A bimodular category, sometimes referred to as a biactegory [9], is to a bimodule what a monoidal category is to a monoid. This is, a plain category $\mathbb{A}$ endowed with a left action of a monoidal category $(\triangleright) \colon \mathbb{M} \times \mathbb{A} \to \mathbb{A}$ and a right action of another, possibly different, monoidal category $(\triangleleft) \colon \mathbb{A} \times \mathbb{N} \to \mathbb{A}$. We can *collage* two bimodular categories along a common monoidal category that acts on both. Later on the paper, exploiting a second axis of categorification, we pass from bimodular categories to *bimodular profunctors*, which are a kind of 2-dimensional bimodule, and we define their collage. This structure facilitates glueing categories together in 2-dimensions: we can represent complexes of morphisms from different categories and glue them together. Collages of string diagrams are the syntactic representations of this glueing, in the same sense that ordinary string diagrams represent tensors in monoidal categories.

We observe that collages of bimodular categories embed into a tricategory of pointed bimodules. This provides a versatile setting where we can interpret many syntaxes already present in the literature.

Contributions. We introduce string diagrams of bimodular categories and we prove they construct the free bimodular category on a signature (Theorem 2.6). We introduce novel string diagrammatic syntax for *functor boxes* and we prove it constructs the free lax monoidal functor on a suitable signature (Theorem 3.4). We describe the tricategory of pointed bimodular profunctors (Definition 4.6) and, in terms of it, we explain the semantics of functor boxes (Proposition 4.9) and internal diagrams (Theorem 5.3), for which we also provide a novel explicit formal syntax (Definition 5.2).

2 String Diagrams of Bimodular Categories

We introduce string diagrams for bimodular categories in terms of the better-known string diagrams of bicategories. In algebra, a *bimodule* is a structure with a compatible left and right action. *Bimodular categories are to bimodules what monoidal categories are to monoids*. Explicitly this means a category, $\mathbb{C}$, acted on by two monoidal categories, $\mathbb{M}$ and $\mathbb{N}$ [48]. Bimodular categories have also been known as "biactegories" [9, 31], while the name "bimodule category" typically refers to actions of certain vector enriched categories [17]. For our purposes, we consider a bimodular category $\mathbb{C}$, as gluing together the two acting categories, $\mathbb{M}$ and $\mathbb{N}$.

To simplify the presentation, we limit ourselves to considering only strict structure, but we expect that all of the results considered hold analogously in the weaker setting. In the following, we assume all monoidal, bimodular, and 2-categories to be strict, along with the associated functors between them.

Definition 2.1. A *bimodular category* $(\mathbb{C}, \mathbb{M}, \mathbb{N})$ is a category $\mathbb{C}$ endowed with a left monoidal action $(\triangleright) \colon \mathbb{M} \times \mathbb{C} \to \mathbb{C}$, and a right monoidal action $(\triangleleft) \colon \mathbb{C} \times \mathbb{N} \to \mathbb{C}$, which are compatible in that $M \triangleright (X \triangleleft N) = (M \triangleright X) \triangleleft N$.

Bimodular categories over arbitrary monoidal categories form a category, **Bimod**. The morphisms $(F, H, K) \colon (\mathbb{C}, \mathbb{M}, \mathbb{N}) \to (\mathbb{D}, \mathbb{P}, \mathbb{Q})$ consist of two monoidal functors $H \colon \mathbb{M} \to \mathbb{P}$ and $K \colon \mathbb{N} \to \mathbb{Q}$ and a functor $F \colon \mathbb{C} \to \mathbb{D}$ that strictly preserves monoidal actions according to H and K.

Every monoidal category $(\mathbb{C}, \otimes, I)$ is a $(\mathbb{C}, \mathbb{C})$-bimodular category with its own tensor product defining the two actions.

2.1 Signature of a Bimodular Category

The next sections exhibit a sound and complete string diagram syntax for bimodular categories. Bimodular string diagrams consist of two monoidal regions glued by a bimodular wire. We begin by defining a notion of bimodular signature and then construct an adjunction (Theorem 2.7) using the notion of *collages*.

Definition 2.2. A *bimodular graph* $(\mathscr{A}, \mathscr{M}, \mathscr{N})$ (the bimodular analogue of a multigraph [43]) is given by three sets of objects $(\mathscr{A}_{obj}, \mathscr{M}_{obj}, \mathscr{N}_{obj})$ and three different types of edges:

- the left-acting edges, a set $\mathscr{M}(\vec{M}; \vec{P})$ for each pair of lists of objects $\vec{M}, \vec{P} \in \mathscr{M}^*_{obj}$,
- the right-acting edges, a set $\mathscr{N}(\vec{N}; \vec{Q})$ for $\vec{N}, \vec{Q} \in \mathscr{N}^*_{obj}$;
- the *central edges*, a set of edges $\mathscr{A}(\vec{M}, A, \vec{N}\ ;\ \vec{P}, B, \vec{Q})$, for each $\vec{M}$, $\vec{P} \in \mathscr{M}^*_{obj}$; each $\vec{N}$, $\vec{Q} \in \mathscr{N}^*_{obj}$ and each A, $B \in \mathscr{A}_{obj}$.

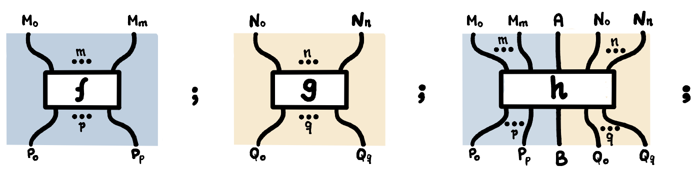


Figure 2: Left, right, and central edges of a bimodular graph.

Proposition 2.3. *Bimodular graphs form a category* **bmGraph**. *We define a morphism of bimodular graphs* $(l, f, g) \colon (\mathscr{A}, \mathscr{M}, \mathscr{N}) \to (\mathscr{A}', \mathscr{M}', \mathscr{N}')$ *to be a triple of functions on objects,* $(l_{obj}, f_{obj}, g_{obj})$*, that extend to the morphism sets. There exists a forgetful functor* $U \colon$ **Bimod** $\to$ **bmGraph**.

This provides a syntactic presentation of bimodular categories. We would like to additionally, construct a free model from a syntactic presentation. We make use of the well-known similar result for 2-categories, exhibiting bimodular categories as certain bicategories: explicitly, those which are the *collage* of a bimodular category.

2.2 The Collage of a Bimodular Category

Each profunctor induces a *collage category*; analogously a bimodular category induces a *collage 2-category*. This section proves that constructing the collage of a bimodular category is left adjoint to considering the bimodular hom-category between any two cells of a 2-category.

Definition 2.4. The *collage* of an $(\mathbb{M},\mathbb{N})$-bimodular category $\mathbb{C}$ is a 2-category, $\mathsf{Coll}_{\mathbb{C}}$. This 2-category has two 0-cells, M and N. The hom-categories are given by $\mathsf{Coll}_{\mathbb{C}}(M,M) = \mathbb{M}$, $\mathsf{Coll}_{\mathbb{C}}(N,N) = \mathbb{N}$, and $\mathsf{Coll}_{\mathbb{C}}(M,N) = \mathbb{C}$; and finally $\mathsf{Coll}_{\mathbb{C}}(N,M)$ is the empty category. The composition of 1-cells is given by the monoidal products and actions.

Definition 2.5. The category of bipointed 2-categories, $\mathbf{2Cat}_2$, has as objects $(\mathbb{A},M,N)$, 2-categories $\mathbb{A}$ with two chosen 0-cells on it, $M \in \mathbb{A}$ and $N \in \mathbb{A}$. A morphism of $\mathbf{2Cat}_2$ is a 2-functor preserving the chosen 0-cells.

Theorem 2.6. *There exists an adjunction* $\mathsf{Coll}_{\mathbb{C}} : \mathbf{Bimod} \rightleftarrows \mathbf{2Cat}_2 : \mathsf{Chosen}$ *given by the collage, and by picking the hom-category between the chosen 0-cells. Moreover, the unit of this adjunction is a natural isomorphism.*

2.3 String Diagrams of Bimodular Categories, via Collages

We have the two ingredients for bimodular string diagrams: sound complete string diagrams for 2-categories, and an embedding of bimodular categories into 2-categories by taking the collage. We combine results to provide an adjunction from bimodular graphs to bimodular categories.

$$\begin{tikzcd}
\mathbf{2Graph} \arrow[d, bend right, "\mathsf{Str}"'] \arrow[d, phantom, "\dashv"] & \mathbf{2Graph}_2 \arrow[r, bend left, "u"] \arrow[r, phantom, "\top"] \arrow[d, bend right, "\mathsf{Str}_2"'] \arrow[d, phantom, "\dashv"] & \mathbf{bmGraph} \arrow[l, bend left, "i"] \arrow[d, dashed, bend right, "\mathsf{bmStr}"'] \arrow[d, phantom, "\dashv"] \\
\mathbf{2Cat} \arrow[u, bend right, "\mathsf{U}"'] & \mathbf{2Cat}_2 \arrow[r, bend left, "\mathsf{Chosen}"] \arrow[r, phantom, "\top"] \arrow[u, bend right, "\mathsf{U}_2"'] & \mathbf{Bimod} \arrow[l, bend left, "\mathsf{Coll}"] \arrow[u, bend right, "\mathsf{U}"']
\end{tikzcd}$$

Figure 3: Summary of adjunctions for the string diagrams of bimodular categories.

Theorem 2.7. *There exists an adjunction between bimodular graphs and bimodular categories. The left adjoint finds the bimodular category whose collage is the free 2-category on the bimodular graph,* $\mathsf{bmStr}\colon \mathbf{bmGraph} \to \mathbf{Bimod}$*. The right adjoint is the forgetful functor* $\mathsf{U}\colon \mathbf{Bimod} \to \mathbf{bmGraph}$.

This result provides a basis for a graphical syntax for bimodular categories. We now sketch an example of how these string diagrams can be of interest, but a larger class of examples come from premonoidal and effectful categories [42].

2.4 Example: Shared State

In the same way that premonoidal categories are particularly well-suited to describe stateful computations, bimodular categories are particularly well-suited to describe shared state between two processes. These processes can be different and even live in different categories. As an example, consider the generators in Figure 4. They represent two different process theories that access a common state with get and put operations.

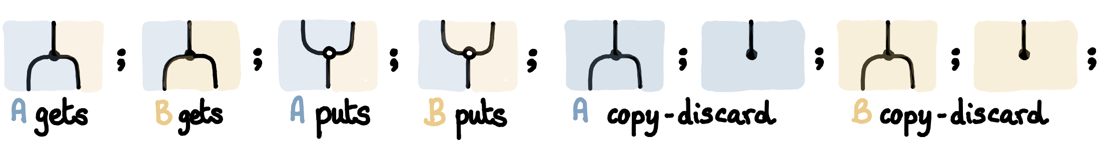


Figure 4: Signature generators for the bimodular theory of shared state.

In the same way that monoidal categories are a good setting for defining monoids and comonoids, bimodular categories are a good setting for defining bimodules. To capture interacting shared state, the generators of Figure 4 are quotiented by the equations of a pair of semifrobenius modules with compatible comonoid actions and semimonoid actions.

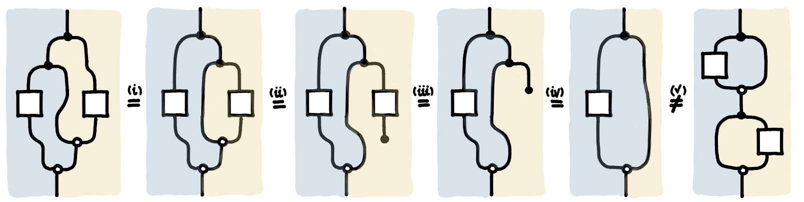


Figure 5: Race condition in bimodular string diagrams.

This setup is enough to exhibit one of the most salient features of shared state: *race conditions*. Race conditions were first studied by Huffman in 1954, who used diagrams to show how the behavior of a shared state is dependent on the relative timing of the actions of the parties [25]. We employ string diagrams of bimodular categories to show how two different timings of the actions – the leftmost and rightmost sides of the equation in Figure 5 – result in two different executions: even when the two get statements are compatible *(i)*, the two put statements interact causing the earlier of the two to be discarded *(ii, iii, iv)*; this causes the discrepancy with the intended protocol *(v)*.

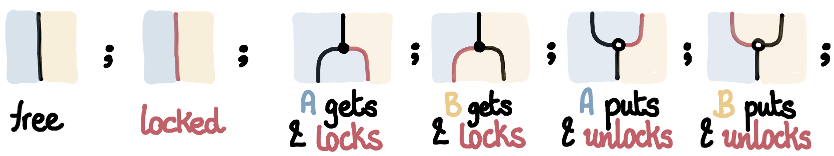


Figure 6: Binary semaphore in bimodular string diagrams.

Race conditions have a commonly accepted workaround: the *binary semaphore* [44]. Dijkstra described general semaphores with the aid of flow diagrams [16]; we instead use bimodular categories to model a binary semaphore (Figure 6). We consider a signature with two object generators, free and locked, for our bimodular category. Each operation must suitably lock or unlock the semaphore, rendering race conditions ill-typed, and leaving most of the interaction equations of the theory of shared state unnecessary.

String diagrams of bimodular categories model a pair of interacting monoidal categories. We can also model an arbitrary number of interacting monoidal categories via a general collage construction.These collages can be subsumed into a 'universe of collages' that we describe in Section 4: the tricategory of pointed bimodular profunctors. To motivate this, we study a second example: the syntax of functor boxes.

3 String Diagrams of Functor Boxes

Functor boxes are an extension of the string diagrammatic notation that represents plain functors, lax, oplax and strong monoidal functors. Functor boxes were introduced by Cockett and Seely [13] and later studied by Melliès [32]. We introduce here a syntactic presentation of (op)lax functor boxes that has the advantage of treating each piece of the box as a separate entity in a 2-category and applying the string diagrammatic calculus of 2-categories.

3.1 Functor box signatures

Definition 3.1. A *functor box signature* $\mathscr{F} = (\mathscr{A}, \mathscr{X}, \mathscr{F}_\bullet, \mathscr{F}^\bullet)$ consists of a pair of sets, $\mathscr{A}_{obj}$ and $\mathscr{X}_{obj}$, and four different types of edges:

- the plain edges, $\mathscr{A}(A_0, \dots, A_n; B_0, \dots, B_m)$ for any objects $A_0, \dots, A_n, B_0, \dots, B_m \in \mathscr{A}_{obj}$;
- the functor box edges, $\mathscr{X}(X_0, ..., X_n; Y_0, ..., Y_m)$ for any objects $X_0, \dots, X_n, Y_0, \dots, Y_m \in \mathscr{X}_{obj}$;
- the in-box edges, $\mathscr{F}_\bullet(A_0, ..., A_n; Y_0, ..., Y_m)$ for any $A_0, ..., A_n \in \mathscr{A}_{obj}$ and $Y_0, ..., Y_m \in \mathscr{X}_{obj}$
- the out-box edges, $\mathscr{F}^\bullet(X_0, ..., X_n; B_0, ..., B_m)$ for any $B_0, ..., B_m \in \mathscr{A}_{obj}$ and $X_0, ..., X_n \in \mathscr{X}_{obj}$.

A *functor box signature morphism* $(h,k,l) \colon (\mathscr{A}, \mathscr{X}, \mathscr{F}) \to (\mathscr{B}, \mathscr{Y}, \mathscr{G})$ is a pair of functions between the object sets, $h_{obj} \colon \mathscr{A}_{obj} \to \mathscr{B}_{obj}$ and $k_{obj} \colon \mathscr{X}_{obj} \to \mathscr{Y}_{obj}$, that extend to a function between the edge sets;

- $h \colon \mathscr{A}(A_0, ..., A_n; B_0, ..., B_m) \to \mathscr{B}(h(A_0), ..., h(A_n); h(B_0), ..., h(B_m))$;
- $k \colon \mathscr{X}(X_0, ..., X_n; Y_0, ..., Y_m) \to \mathscr{Y}(k(X_0), ..., k(X_n); k(Y_0), ..., k(Y_m))$;
- $l_\bullet \colon \mathscr{F}_\bullet(A_0, ..., A_n; Y_0, ..., Y_m) \to \mathscr{G}_\bullet(h(A_0), ..., h(A_n); k(Y_0), ..., k(Y_m))$;
- $l^\bullet \colon \mathscr{F}^\bullet(X_0, ..., X_n; B_0, ..., B_m) \to \mathscr{G}^\bullet(k(X_0), ..., k(X_n); h(B_0), ..., h(B_m))$.

Functor box signatures and homomorphisms form a category, **Fbox**.

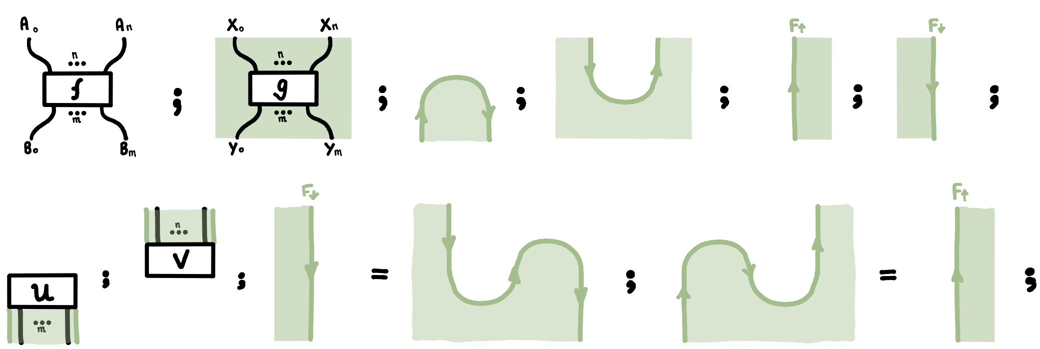


Figure 7: Syntactic 2-category of a lax monoidal functor box signature.

Definition 3.2. The syntactic 2-category of a functor box signature $\mathscr{F} = (\mathscr{A}, \mathscr{X}, \mathscr{F}_\bullet, \mathscr{F}^\bullet)$ is the 2-category freely presented by Figure 7, which we call $\mathbb{S}_{\mathscr{A},\mathscr{X},\mathscr{F}}$.

In other words, the 2-category $\mathbb{S}_{\mathscr{A},\mathscr{X},\mathscr{F}}$ contains exactly two 0-cells, labelled $\mathscr{A}$ and $\mathscr{X}$; it contains a 1-cell $A \colon \mathscr{A} \to \mathscr{A}$ for each $A \in \mathscr{A}_{obj}$, a 1-cell $X \colon \mathscr{X} \to \mathscr{X}$ for each $X \in \mathscr{X}_{obj}$ and, moreover, a pair of adjoint 1-cells $F^\uparrow \colon \mathscr{A} \to \mathscr{X}$ and $F^\downarrow \colon \mathscr{X} \to \mathscr{A}$. Finally, it contains a pair of 2-cells witnessing the adjunction $F^\uparrow \dashv F^\downarrow$, given by $n \colon \mathrm{id} \to F^\uparrow ⨾ F^\downarrow$ and $e \colon F^\downarrow ⨾ F^\uparrow \to \mathrm{id}$ which additionally satisfy the snake equations; and it also contains

- a 2-cell, $f \in \mathbb{S}(\mathscr{A}, \mathscr{A})(A_0 ⨾ \dots ⨾ A_n;\ B_0 ⨾ \dots ⨾ B_m)$, for each *plain edge*;
- a 2-cell, $g \in \mathbb{S}(\mathscr{X}, \mathscr{X})(X_0 ⨾ ... ⨾ X_n;\ Y_0 ⨾ ... ⨾ Y_m)$, for each *functor box edge*;
- a 2-cell, $u \in \mathbb{S}(\mathscr{A}, \mathscr{A})(A_0 ⨾ ... ⨾ A_n;\ F^\uparrow ⨾ Y_0 ⨾ ... ⨾ Y_m ⨾ F^\downarrow)$ for each *in-box edge*; and
- a 2-cell, $v \in \mathbb{S}(\mathscr{A}, \mathscr{A})(F^\uparrow ⨾ X_0 ⨾ ... ⨾ X_n ⨾ F^\downarrow;\ B_0 ⨾ \dots ⨾ B_m)$ for each *out-box edge*.

3.2 Lax Monoidal Functor Semantics

Definition 3.3 (Lax functors category)**.** An object of the *lax functors category*, **Lax**, is a pair of monoidal categories $(\mathbb{A},\mathbb{X})$ together with a lax monoidal functor between them, (F,ε,μ); that is, a functor $F\colon \mathbb{X}\to\mathbb{A}$ endowed with two natural transformations $\varepsilon\colon I\to FI$, and $\mu\colon FX\otimes FY\to F(X\otimes Y)$, satisfying associativity $(\mu\otimes id)\,⨟\,\mu=(id\otimes\mu)\,⨟\,\mu$, left unitality $(\varepsilon\otimes id)\,⨟\,\mu=id$ and right unitality $(id\otimes\varepsilon)\,⨟\,\mu=id$.

A morphism of the *lax functors category*, from $(\mathbb{A},\mathbb{X},F,\varepsilon_F,\mu_F)$ to $(\mathbb{B},\mathbb{Y},G,\varepsilon_G,\mu_G)$ is a pair of monoidal functors $H\colon \mathbb{X}\to\mathbb{A}$ and $K\colon \mathbb{A}\to\mathbb{B}$ such that $F\,⨟\,K=H\,⨟\,G$ and such that $K(\varepsilon_F)=\varepsilon_G$ and $K(\mu_F)=\mu_G$.

Theorem 3.4. *There exists an adjunction between the category of functor box signatures,* **Fbox***, and the category of pairs of monoidal categories with a lax monoidal functor between them,* **Lax***. The free side of this adjunction is given by the syntax of Figure 7.*

Collages, by themselves, explained the 2-region diagrams of bimodular categories; collages will also explain the two-region diagrams of functor boxes in Section 4.5. However, as currently defined, collages are only sufficient to encode the vertical boundaries. To additionally represent boundaries along the horizontal axis we can make use of profunctors between bimodular categories and extend our notion of collage to these structures. Following this thread we find that collages embed into a tricategory of pointed bimodular profunctors, described in the next section, which we consider a universe of interpretation for all of the graphical theories described.

4 Bimodular Profunctors

Where can we interpret all these string diagrams and provide compositional semantics for them? In this section, we introduce a single structure where all the previous calculi take semantics.

We will need two different ingredients: *coends* and *bimodularity*. Coends and profunctors [29, 30], far from being obscure concepts from category theory, can be seen as the right tool to glue together morphisms from different categories [15, 41]; we follow an explicitly *pointed* version of coend calculus, which keeps track of the transformation between profunctors we are constructing (Section 4.3). In a similar sense, bimodular categories tensor together objects from different monoidal categories. Both ideas combine into the calculus of pointed bimodular profunctors.

4.1 Bimodular Profunctors

Consider $\mathbb{C}$ and $\mathbb{D}$, both $(\mathbb{M},\mathbb{N})$-bimodular categories. A natural notion of morphism between them is a functor $\mathbb{C}\to\mathbb{D}$ which preserves both actions. However, there is another notion of morphism between them, which is a generalization of a profunctor between categories to this bimodular setting. Bimodular profunctors are a generalized reformulation of the Tambara modules of Pastro and Street [36].

Definition 4.1. Let $\mathbb{M}$ and $\mathbb{N}$ be two monoidal categories and let $\mathbb{C}$ and $\mathbb{D}$ be two $(\mathbb{M},\mathbb{N})$-bimodular categories. A *bimodular profunctor* from $\mathbb{C}$ to $\mathbb{D}$ is a profunctor $T\colon \mathbb{C}^{op}\times\mathbb{D}\to\mathrm{Set}$ with a natural family of strengths,

$$t_M : T(X,Y)\to T(M\triangleright X, M\triangleright Y),\qquad\text{and}\qquad t^N : T(X,Y)\to T(X\triangleleft N, Y\triangleleft N),$$

such that the actions are associative, $t_M\,⨟\,t_{M'}=t_{M\otimes M'}$ and $t_N\,⨟\,t_{N'}=t_{N\otimes N'}$, unital $t_I=id$ and $t^I=id$, and compatible, $t_M\,⨟\,t^N=t^N\,⨟\,t_M$, up to the coherence isomorphisms of the monoidal category.

Proposition 4.2. *For any pair of monoidal categories, $\mathbb{M}$ and $\mathbb{N}$, there is a 2-category ${}_{\mathbb{M}}\mathbf{Mod}_{\mathbb{N}}$ of $(\mathbb{M},\mathbb{N})$-bimodular categories, bimodular profunctors, and natural transformations between them.*

These will form the hom-bicategories of the tricategory we later define. The other significant piece of data we require is a family of tensors $\otimes : {}_{\mathbb{M}}\mathbf{Mod}_{\mathbb{N}} \times {}_{\mathbb{N}}\mathbf{Mod}_{\mathbb{O}} \to {}_{\mathbb{M}}\mathbf{Mod}_{\mathbb{O}}$, which we now study.

4.2 Tensor of Bimodular Profunctors

The tensor of bimodular categories is similar to the tensor of modules over a monoid in classical algebra: we consider pairs of elements and we quotient out the action of a common scalar [35]. In this case, the quotienting is substituted by an appropriate structural isomorphism: the *equilibrator*.

Definition 4.3 (Tensor of bimodular categories, [35])**.** Let $\mathbb{C}$ be a $(\mathbb{M},\mathbb{N})$-bimodular category and let $\mathbb{D}$ be a $(\mathbb{N},\mathbb{O})$-bimodular category. Their tensor product, $\mathbb{C} \otimes_{\mathbb{N}} \mathbb{D}$, is a category with the same objects as $\mathbb{C} \times \mathbb{D}$: we write them as $X \otimes_{\mathbb{N}} Y$. The category is presented by the morphisms of $\mathbb{C} \times \mathbb{D}$ and a free family of natural isomorphisms, called the *equilibrators*,

$$\tau_{X,N,Y} \colon (X \triangleleft N) \otimes_N Y \to X \otimes_N (N \triangleright Y), \text{ for each } N \in \mathbb{N}, X \in \mathbb{C}, Y \in \mathbb{D},$$

which are additionally quotiented by the following equations up to the structure isomorphisms of the monoidal actions, $\tau_{X,M \otimes N,Y} = \tau_{X \triangleleft M,N,Y} \mathbin{⨟} \tau_{X,M,N \triangleright Y}$, and $\tau_{X,I,Y} = \mathrm{id}$.

Definition 4.4. Let $\mathbb{C}$ and $\mathbb{C}'$ be two $(\mathbb{M},\mathbb{N})$-bimodular categories and let $\mathbb{D}$ and $\mathbb{D}'$ be a $(\mathbb{N},\mathbb{O})$-bimodular categories. Given two bimodular profunctors, $T \colon \mathbb{C} \to \mathbb{C}'$ and $R \colon \mathbb{D} \to \mathbb{D}'$, their tensor is a bimodular profunctor, $T \otimes_{\mathbb{N}} R \colon \mathbb{C} \otimes_{\mathbb{N}} \mathbb{D} \to \mathbb{C}' \otimes_{\mathbb{N}} \mathbb{D}'$, defined by

$$(T \otimes_{\mathbb{N}} R)(X \otimes_N Y; X' \otimes_N Y') = T(X;X') \times R(Y,Y')/(\sim),$$

where $(\sim)$ is the equivalence relation generated by $(t_N(x), y) \sim (x, t_N(y))$.

4.3 Pointed Profunctors

Profunctors deal with families of morphisms, and their natural isomorphisms determine correspondences between these families. However, when we use profunctors for the semantics of string diagrams, we most often want to single out a particular morphism between a particular pair of objects. A simple technique to achieve this is to use *pointed profunctors* instead of simply profunctors: this technique was explicitly described by this second author [41] although it has implicit appearances in the literature [3, 24].

Definition 4.5. A pointed profunctor $(P,p) \colon (\mathbb{A},X) \to (\mathbb{B},Y)$ between two pointed categories with a chosen object $X \in \mathbb{A}_{obj}$ and $Y \in \mathbb{B}_{obj}$ is a profunctor $P \colon \mathbb{A} \to \mathbb{B}$ together with an element $p \in P(A,B)$ of the profunctor evaluated on the chosen object of the categories.

From now on, we work using pointed profunctors instead of plain profunctors.

4.4 The Tricategory of Pointed Bimodular Profunctors

We call *collages of string diagrams* to the diagrams of the tricategory of pointed bimodular profunctors.

Definition 4.6. The tricategory of pointed bimodular profunctors, $\mathbb{B}\mathbb{m}\mathbb{P}\mathbb{r}\mathbb{o}\mathbb{f}_{\mathtt{pt}}$, has as 0-cells the monoidal categories, $\mathbb{M}, \mathbb{N}, \mathbb{O}, \dots$. The 1-cells between two monoidal categories $\mathbb{M}$ and $\mathbb{N}$ are *pointed bimodular categories*, $(\mathbb{A}, \triangleright, \triangleleft, A)$, consisting of a $(\mathbb{M},\mathbb{N})$-bimodular category with two actions $(\mathbb{A}, \triangleright, \triangleleft)$ and some object of that category, $A \in \mathbb{A}$. Pointed bimodular categories compose by the tensor of bimodular categories,

$$(\mathbb{A}, \triangleright, \triangleleft, A) \otimes_{\mathbb{N}} (\mathbb{B}, \triangleright, \triangleleft, B) = (\mathbb{A} \otimes_{\mathbb{N}} \mathbb{B}, \triangleright, \triangleleft, A \otimes_{\mathbb{N}} B).$$

The 2-cells between two pointed bimodular categories $(\mathbb{A}, \triangleright, \triangleleft, A)$ and $(\mathbb{B}, \triangleright, \triangleleft, B)$ are *pointed bimodular profunctors* (P, t, p), consisting of a profunctor $P \colon \mathbb{A} \to \mathbb{B}$ together with a point $p \in P(A, B)$ that are moreover bimodular with compatible natural transformations $t_M \colon P(A;B) \to P(M \triangleright A; M \triangleright B)$, and $t_N \colon P(A;B) \to P(A \triangleleft N; B \triangleleft N)$. These 2-cells compose by profunctor composition and by the tensor of bimodular profunctors.

Finally, the 3-cells between two pointed bimodular profunctors (P, t, p) and (Q, r, q) are bimodular natural transformations that preserve the point, consisting of a natural transformation $\alpha \colon P \to Q$ such that the $\alpha(p) = q$ and, moreover, $t_M \fatsemi \alpha = \alpha \fatsemi r_M$ and $t_N \fatsemi \alpha = \alpha \fatsemi r_N$.

Remark 4.7. At the moment of writing, it is unclear to the authors whether a string diagrammatic calculus for tricategories, described by transformations of the string diagrammatic calculus of bicategories, has been fully described and proved sound and complete. However, there seems to be a consensus that this would be the right language for tricategories: much literature assumes it. Let us close this section by tracking explicitly the assumptions we need to employ a diagrammatic syntax for bimodular profunctors.

Conjecture 4.8. *The previous data satisfies all coherence conditions of a tricategory. Moreover, we can reason with tricategories using the calculus of deformations of string diagrams, extending the string diagrams for quasistrict monoidal 2-categories of Bartlett [2].*

4.5 Functor Boxes via Collages of String Diagrams

The following Figure 8 details how to interpret functor boxes as collages of string diagrams. The colored region represents the domain of the lax monoidal functor; the white region represents the codomain. Morphisms of both categories are interpreted as elements of their respective hom-profunctors, and the laxators are used to merge colored regions. The only element that we will explicitly detail is the bimodular category that appears in the closing and opening wires of a functor box.

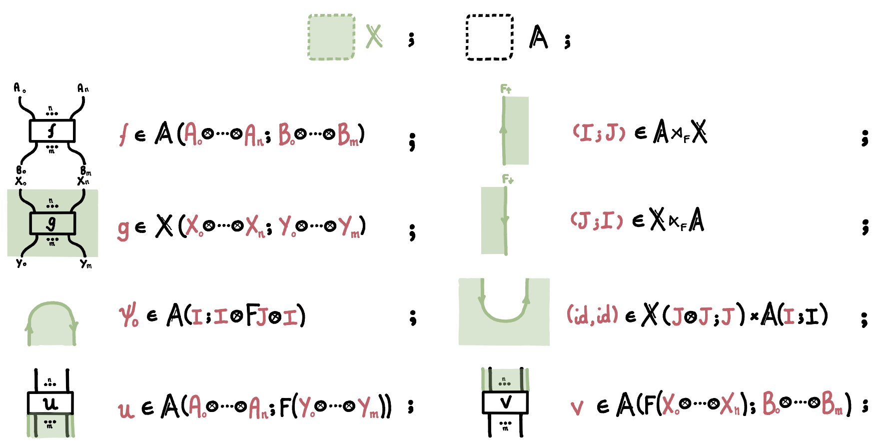


Figure 8: Semantics for functor boxes in terms of pointed bimodular profunctors.

Proposition 4.9 (Bimodular categories of a lax monoidal functor)**.** *Let $\mathbb{X}$ and $\mathbb{A}$ be two monoidal categories and let $F \colon \mathbb{X} \to \mathbb{A}$ be a monoidal functor between them, endowed with natural transformations $\psi_0 \colon J \to FI$ and $\psi_2 \colon FX \otimes FY \to F(X \otimes Y)$. The following profunctors, $\mathbb{A} \rtimes_F \mathbb{X} \colon \mathbb{A} \times \mathbb{X} \to \mathbb{A} \times \mathbb{X}$ and*

$\mathbb{X} \ltimes_F \mathbb{A} \colon \mathbb{X} \times \mathbb{A} \to \mathbb{X} \times \mathbb{A}$ *determine two promonads, and therefore two Kleisli categories.*

$$\mathbb{A} \rtimes_F \mathbb{X}(A,X;B,Y) = \int^{M \in \mathbb{X}} \mathbb{A}(A;B \otimes FM) \times \mathbb{X}(M \otimes X;Y);$$
$$\mathbb{X} \ltimes_F \mathbb{A}(X,A;Y,B) = \int^{M \in \mathbb{X}} \mathbb{A}(A;FM \otimes B) \times \mathbb{X}(M \otimes A;B);$$

These two Kleisli categories are $(\mathbb{A},\mathbb{X})$ *and* $(\mathbb{X},\mathbb{A})$*-bimodular, respectively.*

5 String Diagrams of Internal Diagrams

The tubular 3-dimensional cobordisms of internal diagrams are first described as a Frobenius algebra by Bartlett, Douglas, Schommer-Pries and Vicary [3]. We are indebted to this first introduction, which made internal diagrams into a convenient graphical notation in topological quantum field theory [3]. Internal diagrams themselves were later given explicit semantics in a monoidal bicategory of pointed profunctors; this was the subject of this second author's contribution to *Applied Category Theory 2020* [40]. An important aspect of the syntax of internal diagrams is their 3-dimensional nature: the syntax not only contains string diagrams but also reductions between them.

We introduce here a novel syntactic presentation of *internal diagrams* that has the advantage of treating each piece of an internal diagram (including the closing and opening of tubes) as a separate entity in a tricategory. That is, the identity tube or the multiplication and comultiplication tubes are constructed out of smaller pieces in Figure 9. As a consequence, we are later able to introduce, for the first time, a more refined semantics in terms of a tricategory of *pointed bimodular profunctors*.

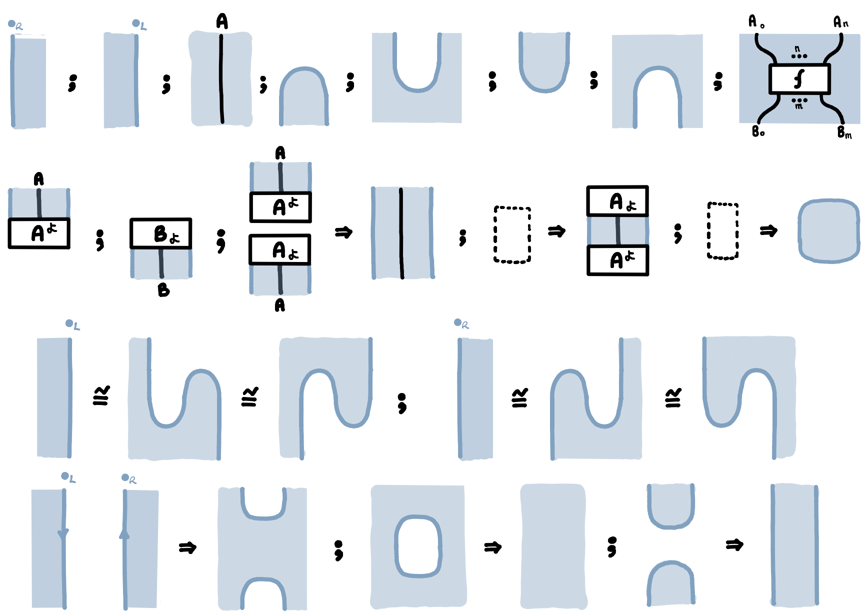


Figure 9: Syntax for open internal diagrams.

Definition 5.1. A *polygraph*, $\mathscr{G}$, is the signature for the string diagrams of a monoidal category. It consists of a set of objects, $\mathscr{G}_{obj}$, and a set of morphisms $\mathscr{G}(A_0,...,A_n;B_0,...,B_m)$ between any two lists of objects, $A_0,...,A_n,B_0,...,B_m \in \mathscr{G}_{obj}$.

Definition 5.2. The *syntactic 3-category of internal diagrams* over a polygraph $\mathscr{G}$ is the 3-category $\mathbf{G}$ presented by the cells in Figure 9. In other words, it contains two 0-cells, $\mathscr{I}$ and $\mathscr{G}$, in white and blue in the figure, respectively. It contains a 1-cell $A \colon \mathscr{G} \to \mathscr{G}$ for each object $A \in \mathscr{G}_{obj}$ and two 1-cells, $L_\bullet \colon \mathscr{I} \to \mathscr{G}$ and $R_\bullet \colon \mathscr{G} \to \mathscr{I}$ forming two 2-adjunctions $(L_\bullet) \dashv (R_\bullet)$ and $(R_\bullet) \dashv (L_\bullet)$ up to a 3-cell. It contains the following 2-cells,

- two 2-cells $n_1 \colon \mathrm{id} \to L_\bullet \fatsemi R_\bullet$ and $e_1 \colon R_\bullet \fatsemi L_\bullet \to \mathrm{id}$ witnessing the 2-adjunction $(L_\bullet) \dashv (R_\bullet)$ and two 2-cells $n_2 \colon 1 \to R_\bullet \fatsemi L_\bullet$ and $e_2 \colon L_\bullet \fatsemi R_\bullet \to \mathrm{id}$ witnessing the 2-adjunction $(R_\bullet) \dashv (L_\bullet)$ – see Vicary and Heunen [22] for a reference on 2-adjunctions and the swallowtail equations;
- two 2-cells, $A^\mathsf{y} \colon L_\bullet \fatsemi A \fatsemi R_\bullet \to \mathrm{id}$ and $A_\mathsf{y} \colon \mathrm{id} \to L_\bullet \fatsemi A \fatsemi R_\bullet$, forming an adjunction $A^\mathsf{y} \dashv A_\mathsf{y}$ for each object $A \in \mathscr{G}_{obj}$; and a 2-cell, $f \colon A_0 \fatsemi \ldots \fatsemi A_n \to B_0 \fatsemi \ldots \fatsemi B_m$, for each edge $f \in \mathscr{G}(A_0, \ldots, A_n; B_0, \ldots, B_m)$.

Finally, it contains the following 3-cells,

- two invertible 3-cells, $\alpha_1 \colon (1 \otimes n_1) \fatsemi (e_1 \otimes 1) \to 1$ and $\beta_1 \colon (n_1 \otimes 1) \fatsemi (1 \otimes e_1) \to 1$, witnessing the 2-adjunction $(L_\bullet) \dashv (R_\bullet)$ and satisfying the swallowtail equations; and two invertible 3-cells, $\alpha'_2 \colon (1 \otimes n_2) \fatsemi (e_2 \otimes 1) \to 1$ and $\beta_2 \colon (n_1 \otimes 1) \fatsemi (1 \otimes e_1) \to 1$, witnessing the 2-adjunction $(R_\bullet) \dashv (L_\bullet)$ and and satisfying the swallowtail equations;
- two 3-cells, $c \colon A^\mathsf{y} \fatsemi A_\mathsf{y} \to 1$ and $i \colon 1 \to A_\mathsf{y} \fatsemi A^\mathsf{y}$, witnessing the adjunction $A^\mathsf{y} \dashv A_\mathsf{y}$ and satisfying the snake equations;
- two 3-cells, $u_i \colon n_1 \fatsemi e_2 \to 1$ and $v_i \colon 1 \to e_2 \fatsemi n_1$ witnessing an adjunction $e_2 \dashv n_1$ and satisfying the snake equations; two 3-cells $u_j \colon 1 \to n_2 \fatsemi e_1$ and $v_i \colon e_1 \fatsemi n_2 \to 1$ witnessing an adjunction $n_2 \dashv e_1$ and satisfying the snake equations.

Theorem 5.3. *For any interpretation of a polygraph into a monoidal category, there exists a 3-functor from the syntactic tricategory of internal diagrams into pointed bimodular profunctors that preserves this interpretation.*

Remark 5.4. This syntax can be exemplified by evaluating a quantum comb [10], or a monoidal lens [39] with a morphism, in terms of internal string diagrams [24], see Figure 10. It has been used more generally to reason about coends in monoidal categories [41] and topological quantum field theory [3].

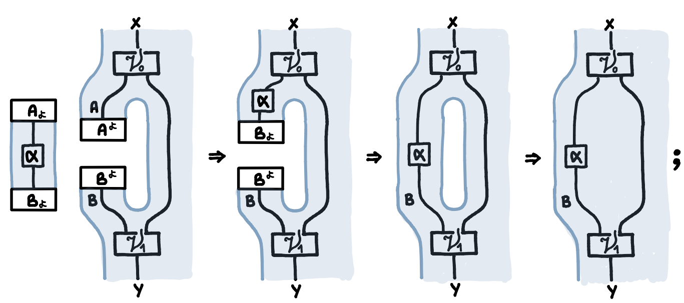


Figure 10: Evaluating a comb in terms of internal string diagrams.

6 Conclusions

Collages of string diagrams provide an abundant graphical calculus. Functor boxes, tensors of bimodular categories and internal diagrams all exist in the graphical calculus of collages. Their technical underpinning is complex: we characterized them as diagrams of pointed bimodular profunctors, but these arrange themselves into a tricategory, which may be difficult to reason about.

Apart from introducing the technique of collages and formalizing multiple extensions to string diagrams, we would like to call attention to the techniques we use: most of our results on soundness and completeness of diagrams are arranged into adjunctions, which allows us to prove them by reusing the better-known results on soundness and completeness for monoidal categories and bicategories.

Related work. An important line of research revolves around *module categories* and *fusion categories*, some specific enriched categories with actions with applications in topological quantum field theories [17, 18, 35]. Especially relevant and recent is Hoek's work, which constructs diagrams for a bimodule category [23, Theorem 3.5.2]. We follow the more elementary notion of bimodular category, called "*biactegory*" in the taxonomy of Capucci and Gavranović [9]. Cockett and Pastro [12] have used instead *linear actions* for concurrency, and even when we take inspiration from their work, their approach is more sophisticated and expressive than our toy example demonstrating bimodular categories (Figure 5).

Most work has been presented for some particular cases of collages: functor boxes have been extensively employed, but never reduced to string diagrams [13, 32]; internal diagrams have served both quantum theory and category theory [3, 24, 28], and can be given semantics into pointed profunctors [40], but again a presentation as string diagrams was missing. A convenient algebra of lenses [39], a particular type of incomplete diagram, has been recently introduced [19], but this is still independent of the semantics of arbitrary internal diagrams.

Finally, the first author has published a blog post that accompanies this manuscript [8].

Further work. It should be possible to "destrictify" many of the results of this paper. We have only presented a 1-adjunction between strict bimodular categories and bipointed 2-categories, but a higher adjunction would allow us to reuse coherence for bicategories to automatically obtain coherence for bimodular categories. We indicated along the paper the conjectures where further work is warranted.

We conjecture that pointed bimodular profunctors form a compact closed tricategory, with the dual of each monoidal category being the *reverse monoidal category*, $A \otimes_{Rev} B = B \otimes A$. Even when it may be conceptually clear what a compact tricategory should be, it is technically challenging to come up with a concrete definition for it in terms of coherence equations.

Acknowledgements

The authors want to thank David A. Dalrymple for discussion on the string diagrammatic interpretation of functor boxes; and Matteo Capucci for several insightful conversations about notions of 2-dimensional profunctor, that helped us understand how to tie disparate aspects of this story together. The authors thank John Baez, the editors, and the anonymous reviewers at ACT23 for multiple comments and suggestions that improved this manuscript.

Dylan Braithwaite was supported by an Industrial CASE studentship from the UK Engineering and Physical Sciences Research Council (EPSRC) and the National Physical Laboratory. Mario Román was supported by the European Union through the ESF Estonian IT Academy research measure (2014-2020.4.05.19-0001).

References

[1] Mario Alvarez-Picallo, Dan R. Ghica, David Sprunger & Fabio Zanasi (2021): *Functorial String Diagrams for Reverse-Mode Automatic Differentiation*. arXiv:2107.13433.

[2] Bruce Bartlett (2014): *Quasistrict symmetric monoidal 2-categories via wire diagrams*. arXiv:1409.2148.

[3] Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries & Jamie Vicary (2015): *Modular categories as representations of the 3-dimensional bordism 2-category*. arXiv:1509.06811.

[4] Guillaume Boisseau & Jeremy Gibbons (2018): *What you needa know about Yoneda: profunctor optics and the Yoneda lemma (functional pearl)*. *Proc. ACM Program. Lang.* 2(ICFP), pp. 84:1–84:27, doi:10.1145/3236779.

[5] Guillaume Boisseau & Pawel Sobocinski (2021): *String Diagrammatic Electrical Circuit Theory*. In Kohei Kishida, editor: *Proceedings of the Fourth International Conference on Applied Category Theory, ACT 2021, Cambridge, United Kingdom, 12-16th July 2021*, *EPTCS* 372, pp. 178–191, doi:10.4204/EPTCS.372.13.

[6] Joe Bolt, Jules Hedges & Philipp Zahn (2019): *Bayesian open games*. *CoRR* abs/1910.03656. arXiv:1910.03656.

[7] Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński & Fabio Zanasi (2019): *Diagrammatic algebra: from linear to concurrent systems*. *Proc. ACM Program. Lang.* 3(POPL), pp. 25:1–25:28, doi:10.1145/3290338.

[8] Dylan Braithwaite (2023): *Diagrams for Actegories*. Available at `https://dylanbraithwaite.github.io/2023/01/31/diagrams-for-actegories.html`.

[9] Matteo Capucci & Bruno Gavranović (2022): *Actegories for the working amthematician*. *arXiv preprint arXiv:2203.16351*.

[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] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore & Mario Román (2022): *Profunctor Optics, a Categorical Update*. arXiv:2001.07488.

[12] Robin B. Cockett & Craig A. Pastro (2009): *The logic of message-passing*. *Science of Computer Programming* 74(8), pp. 498–533, doi:10.1016/j.scico.2007.11.005.

[13] Robin B. Cockett & Robert A. G. Seely (1999): *Linearly distributive functors*. *Journal of Pure and Applied Algebra* 143(1-3), pp. 155–203, doi:10.1016/S0022-4049(98)00110-8.

[14] 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.

[15] 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, p. 14, doi:10.1145/3531130.3533365.

[16] Edsger W. Dijkstra (1962): *Over de sequentialiteit van procesbeschrijvingen*. *Unpublished. Transcribed by Gerrit Jan Veltink for the E.W. Dijkstra Archive, Center for American History.* Available at `https://www.cs.utexas.edu/users/EWD/ewd00xx/EWD35.PDF`.

[17] Christopher L. Douglas, Christopher Schommer-Pries & Noah Snyder (2019): *The balanced tensor product of module categories*. *Kyoto Journal of Mathematics* 59(1), doi:10.1215/21562261-2018-0006.

[18] 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.

[19] Matt Earnshaw, James Hefford & Mario Román (2023): *The Produoidal Algebra of Process Decomposition*. arXiv:2301.11867.

[20] 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*, LICS '18, Association for Computing Machinery, New York, NY, USA, p. 472–481, doi:10.1145/3209108.3209165.

[21] James Hefford & Cole Comfort (2022): *Coend Optics for Quantum Combs*. arXiv:2205.09027.

[22] Chris Heunen & Jamie Vicary (2019): *Categories for Quantum Theory: an introduction*. Oxford University Press, doi:10.1093/oso/9780198739623.001.0001.

[23] Keeley Hoek (2019): *Drinfeld centers for bimodule categories*. Ph.D. thesis, MSc. thesis, The Australian National University.

[24] Nick Hu & Jamie Vicary (2021): *Traced Monoidal Categories as Algebraic Structures in Prof*. In Ana Sokolova, editor: *Proceedings 37th Conference on Mathematical Foundations of Programming Semantics, MFPS 2021, Hybrid: Salzburg, Austria and Online, 30th August - 2nd September, 2021*, *EPTCS* 351, pp. 84–97, doi:10.4204/EPTCS.351.6.

[25] David A Huffman (1954): *The Synthesis of Sequential Switching Circuits*. *Journal of the Franklin Institute* 257(3), pp. 161–190, doi:10.1016/0016-0032(54)90574-8.

[26] 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.

[27] Paul Blain Levy (2022): *Call-by-Push-Value*. *ACM SIGLOG News* 9(2), p. 7–29, doi:10.1145/3537668.3537670.

[28] Leo Lobski & Fabio Zanasi (2022): *String Diagrams for Layered Explanations*. *CoRR* abs/2207.03929, doi:10.48550/arXiv.2207.03929.

[29] Fosco Loregian (2021): *(Co)end Calculus*. London Mathematical Society Lecture Note Series, Cambridge University Press, doi:10.1017/9781108778657.

[30] Saunders Mac Lane (1971): *Categories for the Working Mathematician*. *Graduate Texts in Mathematics* 5, Springer Verlag, doi:10.1007/978-1-4757-4721-8.

[31] Paddy McCrudden (2000): *Categories of representations of coalgebroids*. *Advances in Mathematics* 154(2), pp. 299–332, doi:10.1006/aima.2000.1926.

[32] Paul-André Melliès (2006): *Functorial Boxes in String Diagrams*. In Zoltán Ésik, editor: *Computer Science Logic, 20th International Workshop, CSL 2006, 15th Annual Conference of the EACSL, Szeged, Hungary, September 25-29, 2006, Proceedings*, *Lecture Notes in Computer Science* 4207, Springer, pp. 1–30, doi:10.1007/11874683_1.

[33] Eugenio Moggi (1991): *Notions of Computation and Monads*. *Inf. Comput.* 93(1), pp. 55–92, doi:10.1016/0890-5401(91)90052-4.

[34] Chad Nester (2020): *A Foundation for Ledger Structures*. In Emmanuelle Anceaume, Christophe Bisière, Matthieu Bouvard, Quentin Bramas & Catherine Casamatta, editors: *2nd International Conference on Blockchain Economics, Security and Protocols, Tokenomics 2020, October 26-27, 2020, Toulouse, France*, *OASIcs* 82, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 7:1–7:13, doi:10.4230/OASIcs.Tokenomics.2020.7.

[35] Victor Ostrik (2003): *Module categories, weak Hopf algebras and modular invariants*. *Transformation groups* 8, pp. 177–206, doi:10.1007/s00031-003-0515-6.

[36] Craig Pastro & Ross Street (2007): *Doubles for monoidal categories*. *arXiv preprint arXiv:0711.1859*.

[37] Evan Patterson, David I. Spivak & Dmitry Vagner (2021): *Wiring diagrams as normal forms for computing in symmetric monoidal categories*. *Electronic Proceedings in Theoretical Computer Science* 333, pp. 49–64, doi:10.4204/eptcs.333.4.

[38] Maciej Piróg & Nicolas Wu (2016): *String diagrams for free monads (functional pearl)*. In Jacques Garrigue, Gabriele Keller & Eijiro Sumii, editors: *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, ACM, pp. 490–501, doi:10.1145/2951913.2951947.

[39] Mitchell Riley (2018): *Categories of optics*. *arXiv preprint arXiv:1809.00738*.

[40] Mario Román (2020): *Comb Diagrams for Discrete-Time Feedback*. *CoRR* abs/2003.06214. arXiv:2003.06214.

[41] Mario Román (2020): *Open Diagrams via Coend Calculus*. In David I. Spivak & Jamie Vicary, editors: *Proceedings of the 3rd Annual International Applied Category Theory Conference 2020, ACT 2020, Cambridge, USA, 6-10th July 2020*, *EPTCS* 333, pp. 65–78, doi:10.4204/EPTCS.333.5.

[42] Mario Román (2022): *Promonads and String Diagrams for Effectful Categories*. *CoRR* abs/2205.07664, doi:10.48550/arXiv.2205.07664. arXiv:2205.07664.

[43] Michael Shulman (2016): *Categorical logic from a categorical point of view*. Available on the web. Available at `https://mikeshulman.github.io/catlog/catlog.pdf`.

[44] Abraham Silberschatz, Peter Baer Galvin & Greg Gagne (2018): *Operating System Concepts, 10th Edition*. Wiley. Available at `http://os-book.com/OS10/index.html`.

[45] David I. Spivak (2022): *Generalized Lens Categories via functors $\mathscr{C}^{\mathrm{op}} \to \mathsf{Cat}$*. arXiv:1908.02202.

[46] 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.

[47] Ross Street (1981): *Cauchy characterization of enriched categories*. *Rendiconti del Seminario Matematico e Fisico di Milano* 51(1), pp. 217–233, doi:10.1007/BF02924823.

[48] Zoran Škoda (2009): *Some equivariant constructions in noncommutative algebraic geometry*. *Georgian Mathematical Journal* 16(1), pp. 183–202, doi:10.1515/GMJ.2009.183. arXiv:0811.4770.

The Algebraic Weak Factorisation System for Delta Lenses

Bryce Clarke
Inria Saclay
Palaiseau, France
bryce.clarke@inria.fr

Delta lenses are functors equipped with a suitable choice of lifts, and are used to model bidirectional transformations between systems. In this paper, we construct an algebraic weak factorisation system whose R-algebras are delta lenses. Our approach extends a semi-monad for delta lenses previously introduced by Johnson and Rosebrugh, and generalises to any suitable category equipped with an orthogonal factorisation system and an idempotent comonad. We demonstrate how the framework of an algebraic weak factorisation system provides a natural setting for understanding the lifting operation of a delta lens, and also present an explicit description of the free delta lens on a functor.

1 Introduction

Delta lenses were first introduced by Diskin, Xiong, and Czarnecki [17] as an algebraic framework for *bidirectional transformations* [1, 13] between systems, particularly in the context of *model-driven engineering* [16, 28]. The original motivation behind delta lenses came from adapting the classical notion of a lens [19] from a "state-based" setting to a "delta-based" setting. Instead of treating a system as a mere *set* of states, it should be regarded as a *category*, whose objects are the states of the system and whose morphisms are the updates (or deltas) between them. The purpose of delta lenses is to model the notion of *synchronisation* between systems through specifying how certain updates between states are propagated.

A delta lens is a functor $f\colon A \to B$ equipped with a *lifting operation*, see (1), that satisfies certain axioms. The lifting operation specifies, for each object a in A and for each morphism $u\colon fa \to b$ in B, a morphism $\varphi(a,u)\colon a \to a'$, often called the *chosen lift*, such that $f\varphi(a,u) = u$. The axioms placed on the lifting operation ensure that it respects identities and composition. Thus a delta lens is a functor equipped with additional *algebraic structure*, and it is natural to wonder if delta lenses arise as algebras for a monad. In this paper, we provide an answer in the affirmative.

$$\begin{tikzcd}
\{0\} \arrow[r, "a"] \arrow[d] & A \arrow[d, "f"] \\
\{0 \to 1\} \arrow[r, "u"] \arrow[ur, dashed, "\varphi(a,u)"] & B
\end{tikzcd} \tag{1}$$

The question of asking whether certain kinds of lenses are algebras for a monad is not new. Classical state-based lenses [19] were characterised by Johnson, Rosebrugh, and Wood [26] as algebras for a monad on the slice category $\mathcal{S}\mathrm{et}/B$. The same authors later introduced the notion of a *c-lens* [25], better known as a *split opfibration*, and characterised them as algebras for a monad, first introduced by Street [29], on the slice category $\mathcal{C}\mathrm{at}/B$. Delta lenses generalise state-based lenses and split opfibrations [24], however they were only shown by Johnson and Rosebrugh [23] to be certain algebras for a *semi-monad* (a monad without a unit) on $\mathcal{C}\mathrm{at}/B$. One of the contributions of the current paper is resolve this gap in the literature.

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 54–69, doi:10.4204/EPTCS.397.4

Although it is generally useful to know when a mathematical structure arises as an algebra for a monad, in isolation this result provides limited benefit towards a deeper understanding of lenses. One reason is that we wish to study lenses as *morphisms* of a category, rather than *objects* in a category of algebras. The knowledge that lenses are morphisms with algebraic structure does not provide any information of how to sequentially *compose* them, nor justification for why this algebraic structure encodes a notion of *lifting*.

Cofunctors[1] are a natural kind of morphism between categories [2, 22] which fundamentally involve a lifting operation and admit a straightforward sequential composition. The characterisation of delta lenses as a compatible functor and cofunctor [3, 7], together with related characterisations of state-based lenses and split opfibrations [8], provides a clear understanding of their composition and lifting, and has led to several fruitful developments in the study of lenses in applied category theory [6, 10, 14]. However the question remains: why do lenses frequently arise as algebras for a monad?

An *algebraic weak factorisation system* [4], also known as a *natural weak factorisation system* [21], generalises the notion of an orthogonal factorisation system (OFS) on a category. An algebraic weak factorisation system (AWFS) on a category $\mathcal{C}$ consists of a comonad (L, ε, Δ) and a monad (R, η, μ) on $\mathcal{C}^{\mathbf{2}}$ that are suitably compatible. The categories of *L-coalgebras* and *R-algebras* of an AWFS (L,R) replace the usual *left* and *right* classes of morphisms of an OFS. In particular, every morphism factors into a cofree L-coalgebra followed by free R-algebra, and every lifting problem (2), where (f, p) is a L-coalgebra and (g,q) is a R-algebra, admits a chosen lift $\varphi_{f,g}\langle h,k\rangle$ making the diagram commute. Crucially, these chosen lifts also induce a canonical composition of R-algebras [4, Section 2.8]. Both classical state-based lenses and split opfibrations arise as R-algebras for an AWFS on $\mathcal{S}\mathrm{et}$ and $\mathcal{C}\mathrm{at}$, respectively.

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "{(f,p)}"'] & C \arrow[d, "{(g,q)}"] \\ B \arrow[r, "k"'] \arrow[ur, dashed, "{\varphi_{f,g}\langle h,k\rangle}" description] & D \end{tikzcd} \tag{2}$$

The main contribution of this paper is to construct an algebraic weak factorisation system (L,R) on $\mathcal{C}\mathrm{at}$ whose R-algebras are precisely delta lenses. The principal benefit is a new framework for understanding lenses as algebras for a monad that naturally incorporates the fundamental aspects of composition and lifting. In addition, we are able to generalise the notion of delta lens to any suitable category equipped with an orthogonal factorisation system and idempotent comonad, as well as present an explicit description of the free delta lens on the functor. This approach to lenses as algebras for a monad also highlights an interesting duality with their recent characterisation as coalgebras for a comonad [9].

Overview of the paper. In Section 2 we review the necessary background material on delta lenses and factorisation systems. In particular, we recall two important structures on $\mathcal{C}\mathrm{at}$, the *comprehensive factorisation system* (Example 6) and the *discrete category comonad* (Example 12), which are generalised in our main constructions to an orthogonal factorisation system and an idempotent comonad, respectively. In Section 3 we utilise these structures on a category $\mathcal{C}$ to build a semi-monad on $\mathcal{C}^{\mathbf{2}}$ (Proposition 13), and show that when $\mathcal{C} = \mathcal{C}\mathrm{at}$ (Example 15), we recover delta lenses as certain algebras for this semi-monad (Theorem 17 and Appendix A). In Section 4 we enhance this construction to a monad (Theorem 19) using pushouts in $\mathcal{C}$, and prove that when $\mathcal{C} = \mathcal{C}\mathrm{at}$, the algebras for this monad are delta lenses (Theorem 23). We also describe the free delta lens on a functor (Example 27). Section 5 completes the construction of an algebraic weak factorisation system on $\mathcal{C}$ (Theorem 29) and shows how delta lenses lift against the L-coalgebras when $\mathcal{C} = \mathcal{C}\mathrm{at}$. Section 6 presents some concluding remarks and avenues for future work.

[1]The term *retrofunctor* proposed by Di Meglio [14] is preferred, but not yet in widespread use.

2 Background

2.1 Delta lenses

We introduce the category $\mathcal{L}\mathrm{ens}$ whose objects are delta lenses, which we will later show is the category of algebras for a monad on $\mathcal{C}\mathrm{at}^{\mathbf{2}}$. For further details and examples, we refer the reader to [11, Chapter 2].

Definition 1. A *delta lens* $(f,\varphi)\colon A \to B$ consists of a functor $f\colon A \to B$ together with a *lifting operation*

$$(a \in A, u\colon fa \to b \in B) \qquad \longmapsto \qquad \varphi(a,u)\colon a \to p(a,u),$$

where $p(a,u) = \mathrm{cod}(\varphi(a,u))$, that satisfies the following three axioms:

(L1) $f\varphi(a,u) = u$

(L2) $\varphi(a,1_{fa}) = 1_a$

(L3) $\varphi(a, v\circ u) = \varphi(p(a,u),v)\circ\varphi(a,u)$

Example 2. A *discrete opfibration* is a functor $f\colon A \to B$ such that for each pair $(a \in A, u\colon fa \to b \in B)$ there is a unique morphism $\bar{f}(a,u)\colon a \to a'$ in A for which $f\bar{f}(a,u) = u$. Thus each discrete opfibration f admits a unique lifting operation $\bar{f}$ such that the pair $(f,\bar{f})$ is a delta lens. Conversely, the underlying functor f of a delta lens (f,φ) is a discrete opfibration if $\varphi(a,fw) = w$ for all morphisms $w\colon a \to a'$ in A.

Definition 3. Let $\mathcal{L}\mathrm{ens}$ denote the category whose objects are delta lenses and whose morphisms $\langle h,k\rangle$ from (f,φ) to (g,ψ) consist of a pair of functors h and k such that $k\circ f = g\circ h$ and $h\varphi(a,u) = \psi(ha,ku)$.

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "{(f,\varphi)}"'] & C \arrow[d, "{(g,\psi)}"] \\ B \arrow[r, "k"'] & D \end{tikzcd} \quad \rightsquigarrow \quad \begin{tikzcd} \{0\} \arrow[r, "a"] \arrow[d] & A \arrow[r, "h"] \arrow[d, "f"] & C \arrow[d, "g"] \\ \{0\to 1\} \arrow[r, "u"] \arrow[ur, dashed, "{\varphi(a,u)}"] & B \arrow[r, "k"] & D \end{tikzcd} \quad = \quad \begin{tikzcd} \{0\} \arrow[r, "ha"] \arrow[d] & C \arrow[d, "g"] \\ \{0\to 1\} \arrow[r, "ku"] \arrow[ur, dashed, "{\psi(ha,ku)}"] & D \end{tikzcd}$$

Let $U\colon \mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}^{\mathbf{2}}$ denote the canonical forgetful functor that sends (f,φ) to f.

2.2 Factorisation systems

We recall two related notions of factorisation system on a category: *orthogonal factorisations systems* [20] and *algebraic weak factorisation systems* [4, 21]. For a full account, we refer the reader to [4].

Definition 4. An *orthogonal factorisation system* $(\mathcal{E},\mathcal{M})$ on a category $\mathcal{C}$ consists of two classes of morphisms $\mathcal{E}$ and $\mathcal{M}$, both containing the isomorphisms and closed under composition, such that:

(i) **Factorisation**: Every morphism f of $\mathcal{C}$ admits a factorisation $f = m\circ e$ with $e \in \mathcal{E}$ and $m \in \mathcal{M}$;

(ii) **Orthogonality**: For each solid commutative square in $\mathcal{C}$ below such that $e \in \mathcal{E}$ and $m \in \mathcal{M}$, there exists a unique morphism h such that $f = h\circ e$ and $g = m\circ h$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "e"'] & C \arrow[d, "m"] \\ B \arrow[r, "g"'] \arrow[ur, dashed, "\exists!", "h"'] & D \end{tikzcd}$$

Notation 5. As an aid when diagram-chasing, the morphisms in the left class $\mathcal{E}$ and the right class $\mathcal{M}$ of an orthogonal factorisation system on $\mathcal{C}$ will be decorated in the remainder of the paper as follows.

$$\bullet \xrightarrow{e\in\mathcal{E}} \mathrel{\!\!\rightarrow}\!\bullet \qquad\qquad \bullet \rightarrowtail^{m\in\mathcal{M}} \bullet$$

Example 6. A functor $f\colon A \to B$ is called *initial* if, for each object $b \in B$, the comma category f/b is connected. The *comprehensive factorisation system* [31] is an orthogonal factorisation system $(\mathcal{E},\mathcal{M})$ on $\mathcal{C}\mathrm{at}$ in which $\mathcal{E}$ is the class of initial functors and $\mathcal{M}$ is the class of discrete opfibrations.

Lemma 7. *If $(\mathcal{E},\mathcal{M})$ be an orthogonal factorisation system on $\mathcal{C}$, then the following properties hold:*

(1) The class $\mathcal{E}$ is stable under pushouts in $\mathcal{C}$.

(2) If $g \circ f$ and f are in $\mathcal{E}$, then g is in $\mathcal{E}$. Dually, if $g \circ f$ and g are in $\mathcal{M}$, then f is in $\mathcal{M}$.

Definition 8. A *functorial factorisation* (L,E,R) on a category $\mathcal{C}$ is a section $(L,E,R)\colon \mathcal{C}^{\mathbf{2}} \to \mathcal{C}^{\mathbf{3}}$ to the composition functor $\mathcal{C}^{\mathbf{3}} \to \mathcal{C}^{\mathbf{2}}$. The factorisation of a morphism in $\mathcal{C}^{\mathbf{2}}$ is denoted as follows.

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] & C \arrow[d, "g"] \\ B \arrow[r, "k"'] & D \end{tikzcd} \quad \longmapsto \quad \begin{tikzcd} A \arrow[r, "h"] \arrow[d, "Lf"'] \arrow[dd, bend right=60, "f"'] & C \arrow[d, "Lg"] \arrow[dd, bend left=60, "g"] \\ Ef \arrow[r, "E\langle h,k\rangle"] \arrow[d, "Rf"'] & Eg \arrow[d, "Rg"] \\ B \arrow[r, "k"'] & D \end{tikzcd}$$

Remark 9. Each functorial factorisation (L,E,R) on $\mathcal{C}$ induces a copointed endofunctor (L,ε) and a pointed endofunctor (R,η) on $\mathcal{C}^{\mathbf{2}}$, where the components of $\varepsilon\colon L \Rightarrow 1$ and $\eta\colon 1 \Rightarrow R$ at f are given below.

$$\begin{tikzcd} A \arrow[r, equal] \arrow[d, "Lf"'] & A \arrow[d, "f"] \\ Ef \arrow[r, "Rf"'] & B \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "Lf"] \arrow[d, "f"'] & Ef \arrow[d, "Rf"] \\ B \arrow[r, equal] & B \end{tikzcd} \tag{3}$$

Definition 10. [4, Section 2.2] An *algebraic weak factorisation system* (L,R) on a category $\mathcal{C}$ consists of:

(i) A functorial factorisation (L,E,R) on $\mathcal{C}$;

(ii) An extension of (L,ε) to a comonad (L,ε,Δ) on $\mathcal{C}^{\mathbf{2}}$;

(iii) An extension of (R,η) to a monad (R,η,μ) on $\mathcal{C}^{\mathbf{2}}$;

(iv) A distributive law $\lambda\colon LR \Rightarrow RL$ of the comonad L over the monad R with $\lambda_f = \langle \Delta_f, \mu_f\rangle$.

2.3 Idempotent comonads and weak equivalences

Given an idempotent comonad (M,ι) on a category $\mathcal{C}$, let $\mathcal{W} = \{f \in \mathcal{C} \mid Mf \text{ is invertible}\}$ denote the class of morphisms in $\mathcal{C}$ whose members are called *weak equivalences*. This class satisfies the *2-out-of-3 property*, and contains the isomorphisms, thus making $\mathcal{C}$ a *category with weak equivalences* [18]. Since the comonad M is idempotent, each counit component ι_A is inverted by M and therefore a morphism of $\mathcal{W}$. If M preserves pushouts, the morphisms in $\mathcal{W}$ are stable under pushout along morphisms in $\mathcal{C}$.

Notation 11. As a visual aid when diagram-chasing, the morphisms in the class $\mathcal{W}$ of weak equivalences of a category $\mathcal{C}$ will be decorated in the remainder of the paper as follows.

$$\bullet \xrightarrow[\sim]{w \in \mathcal{W}} \bullet$$

Example 12. Let $(-)_0\colon \mathcal{C}\mathrm{at} \to \mathcal{C}\mathrm{at}$ denote the idempotent comonad that assigns a category A to its corresponding *discrete category* A_0 with counit component $\iota_A\colon A_0 \to A$. The endofunctor $(-)_0$ has a right adjoint (the *codiscrete category* monad) and therefore preserves all colimits. A functor $f\colon A \to B$ is called *bijective-on-objects* if f_0 is invertible; these are the weak equivalences with respect to $(-)_0$.

3 Delta lenses as certain algebras for a semi-monad

Throughout this section, let $(\mathcal{E},\mathcal{M})$ be an orthogonal factorisation system on a category $\mathcal{C}$, and let (M,ι) be an idempotent comonad on $\mathcal{C}$ with corresponding class $\mathcal{W}$ of weak equivalences.

3.1 Constructing a semi-monad for delta lenses

We now construct a semi-monad (T,ν) on the category $\mathcal{C}^{\mathbf{2}}$, for a category $\mathcal{C}$ equipped with an idempotent comonad (M,ι) and an orthogonal factorisation system $(\mathcal{E},\mathcal{M})$. We show that when $\mathcal{C}=\mathcal{C}\mathrm{at}$ equipped with the discrete category comonad and the comprehensive factorisation system, this specialises to the semi-monad defined on $\mathcal{C}\mathrm{at}^{\mathbf{2}}$ by Johnson and Rosebrugh [23, Section 6].

We begin by constructing an endofunctor $T\colon \mathcal{C}^{\mathbf{2}}\to\mathcal{C}^{\mathbf{2}}$. Given a morphism $f\colon A\to B$ in $\mathcal{C}$, we first pre-compose with the counit component $\iota_A\colon MA\to A$ and then choose an $(\mathcal{E},\mathcal{M})$-factorisation of the resulting morphism as depicted in commutative square (i) below; this defines the action of T on objects in $\mathcal{C}^{\mathbf{2}}$. Given a morphism $\langle h,k\rangle\colon f\to g$ in $\mathcal{C}^{\mathbf{2}}$, there exists a unique morphism $J\langle h,k\rangle\colon Jf\to Jg$ in $\mathcal{C}$ by applying the orthogonality property; the action of T on the morphism $\langle h,k\rangle$ is given by the commutative square (ii) depicted below. Note that the equation (4) holds by naturality of $\iota\colon M\Rightarrow 1$ at the morphism h.

$$
\begin{tikzcd}
MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] & A \arrow[r, "h"] \arrow[dd, "f"] & C \arrow[dd, "g"] \\
Jf \arrow[d, tail, "Tf"'] \arrow[r, phantom, "\text{(i)}"] & & \\
B \arrow[r, equal] & B \arrow[r, "k"'] & D
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
MA \arrow[r, "Mh"] \arrow[d, two heads, "Sf"'] & MC \arrow[r, "\iota_C", "\sim"'] \arrow[d, two heads, "Sg"] & C \arrow[dd, "g"] \\
Jf \arrow[r, dashed, "J\langle h,k\rangle"] \arrow[d, tail, "Tf"'] \arrow[dr, phantom, "\text{(ii)}"] & Jg \arrow[d, tail, "Tg"] & \\
B \arrow[r, "k"'] & C \arrow[r, equal] & C
\end{tikzcd}
\tag{4}
$$

Applying the functor T to the morphism $Tf\colon Jf\to B$ and using the orthogonality property, we obtain the component ν_f of the multiplication $\nu\colon T^2\Rightarrow T$ at f as depicted in the commutative square (iii) below. Naturality of ν at follows from noticing in (5) that $J\langle h,k\rangle\circ\nu_f=\nu_g\circ J\langle J\langle h,k\rangle,k\rangle$ by orthogonality.

$$
\begin{tikzcd}
MJf \arrow[r, "\iota_{Jf}", "\sim"'] \arrow[d, two heads, "STf"'] & Jf \arrow[r, "J\langle h,k\rangle"] \arrow[dd, tail, "Tf"] & Jg \arrow[dd, tail, "Tg"] \\
JTf \arrow[ur, tail, dashed, "\nu_f"'] \arrow[d, tail, "T^2f"'] \arrow[dr, phantom, "\text{(iii)}"] & & \\
B \arrow[r, equal] & B \arrow[r, "k"'] & D
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
MJf \arrow[r, "MJ\langle h,k\rangle"] \arrow[d, two heads, "STf"'] & MJg \arrow[r, "\iota_{Jg}", "\sim"'] \arrow[d, two heads, "STg"] & Jg \arrow[dd, tail, "Tg"] \\
JTf \arrow[r, dashed, "J\langle J\langle h,k\rangle,k\rangle"] \arrow[d, tail, "T^2f"'] & JTg \arrow[ur, tail, dashed, "\nu_g"'] \arrow[d, tail, "T^2g"] & \\
B \arrow[r, "k"'] & D \arrow[r, equal] & D
\end{tikzcd}
\tag{5}
$$

The associative law for ν follows from observing in (6) that $\nu_f\circ\nu_{Tf}=\nu_f\circ J\langle\nu_f,1_B\rangle$ by orthogonality.

$$
\begin{tikzcd}
MJTf \arrow[r, "\iota_{JTf}", "\sim"'] \arrow[d, two heads, "ST^2f"'] & JTf \arrow[r, tail, "\nu_f"] \arrow[dd, tail, "T^2f"] & Jf \arrow[dd, tail, "Tf"] \\
JT^2f \arrow[ur, tail, dashed, "\nu_{Tf}"'] \arrow[d, tail, "T^3f"'] & & \\
B \arrow[r, equal] & B \arrow[r, equal] & B
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
MJTf \arrow[r, "M\nu_f"] \arrow[d, two heads, "ST^2f"'] & MJf \arrow[r, "\iota_{Jf}", "\sim"'] \arrow[d, two heads, "STf"] & Jf \arrow[dd, tail, "Tf"] \\
JT^2f \arrow[r, tail, dashed, "J\langle\nu_f,1_B\rangle"] \arrow[d, tail, "T^3f"'] & JTf \arrow[ur, tail, dashed, "\nu_f"'] \arrow[d, tail, "T^2f"] & \\
B \arrow[r, equal] & B \arrow[r, equal] & B
\end{tikzcd}
\tag{6}
$$

We have thus constructed an endofunctor $T\colon\mathcal{C}^{\mathbf{2}}\to\mathcal{C}^{\mathbf{2}}$ with an associative multiplication $\nu\colon T^2\Rightarrow T$.

Proposition 13. *The pair (T,ν) is a semi-monad on $\mathcal{C}^2$.*
Corollary 14. *The semi-monad (T,ν) on $\mathcal{C}^{\mathbf{2}}$ restricts to a semi-monad in the 2-category $\mathcal{C}\mathrm{AT}/\mathcal{C}$ on the codomain functor* $\mathrm{cod}\colon \mathcal{C}^{\mathbf{2}} \to \mathcal{C}$. *In particular, (T,ν) induces a semi-monad on each slice category $\mathcal{C}/B$.*
Example 15. Consider the category $\mathcal{C}\mathrm{at}$ equipped with the comprehensive factorisation system and the discrete category comonad. Given a functor $f\colon A \to B$, the category Jf defined in (4) is given by the coproduct $\sum_{a\in A_0} fa/B$ of the coslice categories indexed by the discrete category A_0. The objects in Jf are pairs $(a \in A, u\colon fa \to b \in B)$, while morphisms $\langle 1_a, v\rangle\colon (a,u_1) \to (a,u_2)$ are given by morphisms $v \in B$ such that $u_2 = v \circ u_1$. The functor $Sf\colon A_0 \twoheadrightarrow J_f$ has an assignment on objects $a \mapsto (a, 1_{fa})$, and is an *initial functor* since each slice category $Sf/(a,u)$ is isomorphic to the terminal category and hence connected. The functor $Tf\colon Jf \rightarrowtail B$ is given by the codomain projection with assignment on objects $(a,u) \mapsto \mathrm{cod}(u)$, and is a *discrete opfibration*. In this setting, restricting the semi-monad (T,ν) to the slice categories $\mathcal{C}\mathrm{at}/B$ coincides with semi-monad for delta lenses defined by Johnson and Rosebrugh [23].

3.2 Delta lenses as certain semi-monad algebras

An *algebra* (f,p) for the semi-monad (T,ν) on the codomain functor $\mathrm{cod}\colon \mathcal{C}^{\mathbf{2}} \to \mathcal{C}$ (or, equivalently, on the slice category $\mathcal{C}/B$) consists of a pair of morphisms $f\colon A \to B$ and $p\colon Jf \to A$ such that the following diagrams commute.

$$\begin{tikzcd} Jf \arrow[r, "p"] \arrow[d, tail, "Tf"'] & A \arrow[d, "f"] \\ B \arrow[r, equal] & B \end{tikzcd} \qquad \begin{tikzcd} JTf \arrow[r, "J\langle p, 1_B\rangle"] \arrow[d, tail, "\nu_f"'] & Jf \arrow[d, "p"] \\ Jf \arrow[r, "p"'] & A \end{tikzcd} \tag{7}$$

Johnson and Rosebrugh (JR) introduced an additional condition on the algebras for the semi-monad (T,ν) on $\mathcal{C}\mathrm{at}/B$ which we now adapt to our more general setting under the name *JR-algebra*. The intuition is that this additional condition replaces the missing "unit law" that an algebra for a monad would satisfy.
Definition 16. A *JR-algebra* is an algebra (f,p) for the semi-monad (T,ν) on the codomain functor $\mathrm{cod}\colon \mathcal{C}^{\mathbf{2}} \to \mathcal{C}$ such that the following diagram commutes.

$$\begin{tikzcd} MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] & A \\ Jf \arrow[ur, "p"'] & \end{tikzcd} \tag{8}$$

A *morphism* $\langle h,k\rangle\colon (f,p) \to (g,q)$ of algebras for the semi-monad (T,ν) consists of a pair of morphisms h and k such that the following equation in $\mathcal{C}^{\mathbf{2}}$ holds.

$$\begin{tikzcd} Jf \arrow[r, "p"] \arrow[d, tail, "Tf"'] & A \arrow[r, "h"] \arrow[d, "f"] & C \arrow[d, "g"] \\ B \arrow[r, equal] & B \arrow[r, "k"'] & D \end{tikzcd} = \begin{tikzcd} Jf \arrow[r, "J\langle h,k\rangle"] \arrow[d, tail, "Tf"'] & Jg \arrow[r, "q"] \arrow[d, tail, "Tg"] & C \arrow[d, "g"] \\ B \arrow[r, "k"'] & D \arrow[r, equal] & D \end{tikzcd} \tag{9}$$

Let $\mathcal{A}\mathrm{lg}(T,\nu)$ denote the category of algebras for the semi-monad (T,ν) on the codomain functor $\mathrm{cod}\colon \mathcal{C}^{\mathbf{2}} \to \mathcal{C}$, and let $\mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$ denote the full subcategory of JR-algebras.
Theorem 17. *If $\mathcal{C} = \mathcal{C}\mathrm{at}$ equipped with the discrete category comonad and the comprehensive factorisation system, then there is an isomorphism of categories $\mathcal{L}\mathrm{ens} \cong \mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$.*

Proof. This result is due to Johnson and Rosebrugh [23]. See Appendix A for a proof in our notation. $\square$

4 Delta lenses as algebras for a monad

Throughout this section, let $(\mathcal{E},\mathcal{M})$ be an orthogonal factorisation system on a category $\mathcal{C}$ with (chosen) pushouts, and let (M,ι) be an idempotent comonad on $\mathcal{C}$ such that $M\colon \mathcal{C}\to\mathcal{C}$ preserves pushouts.

4.1 Constructing a monad for delta lenses

We now extend the semi-monad (T,ν) to a monad (R,η,μ) on $\mathcal{C}^{\mathbf{2}}$, for a category $\mathcal{C}$ as described above. Our approach is to utilise the universal properties of pushouts and orthogonal factorisation systems, as well as properties of the class of weak equivalences for the idempotent comonad, to construct the necessary data for the monad from that of the semi-monad (T,ν).

We begin by constructing an endofunctor $R\colon \mathcal{C}^{\mathbf{2}}\to\mathcal{C}^{\mathbf{2}}$. Given a morphism $f\colon A\to B$ in $\mathcal{C}$, first construct the pushout of ι_A along Sf from (4), and then use the universal property of the pushout to define $Rf\colon Ef\to B$ as depicted on the left below; this defines the action of R on objects in $\mathcal{C}^{\mathbf{2}}$.

$$
\begin{tikzcd}
MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] & A \arrow[d, two heads, "Lf"] \arrow[dd, bend left=60, "f"] \\
Jf \arrow[r, "\alpha_f", "\sim"'] \arrow[d, tail, "Tf"'] \arrow[ur, phantom, "\ulcorner"] & Ef \arrow[d, dashed, "Rf"] \\
B \arrow[r, equal] & B
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
MEf \arrow[r, "\iota_{Ef}", "\sim"'] \arrow[d, two heads, "SRf"'] & Ef \arrow[d, two heads, "LRf"] \arrow[dd, bend left=60, "Rf"] \\
JRf \arrow[r, "\alpha_{Rf}", "\sim"'] \arrow[d, tail, "TRf"'] \arrow[ur, phantom, "\ulcorner"] & ERf \arrow[d, dashed, "R^2f"] \\
B \arrow[r, equal] & B
\end{tikzcd}
\tag{10}
$$

Given a morphism $\langle h,k\rangle\colon f\to g$ in $\mathcal{C}^{\mathbf{2}}$, there exists a unique morphism $E\langle h,k\rangle\colon Jf\to Jg$ in $\mathcal{C}$, as depicted below, by the universal property of the pushout, where $J\langle h,k\rangle$ is defined in (4). It is not difficult to show through diagram-chasing that $Rg\circ E\langle h,k\rangle = k\circ Rf$, thus defining the action of R on morphisms of $\mathcal{C}^{\mathbf{2}}$.

$$
\begin{tikzcd}
MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] & A \arrow[d, two heads, "Lf"] \arrow[dr, "h"] & \\
Jf \arrow[r, "\alpha_f", "\sim"'] \arrow[dr, "J\langle h,k\rangle"'] \arrow[ur, phantom, "\ulcorner"] & Ef \arrow[dr, dashed, "E\langle h,k\rangle"] & C \arrow[d, two heads, "Lg"] \\
& Jg \arrow[r, "\alpha_g", "\sim"'] & Eg
\end{tikzcd}
=
\begin{tikzcd}
MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] \arrow[dr, "Mh"] & A \arrow[dr, "h"] & \\
Jf \arrow[dr, "J\langle h,k\rangle"'] & MC \arrow[r, "\iota_C", "\sim"'] \arrow[d, two heads, "Sg"'] & C \arrow[d, two heads, "Lg"] \\
& Jg \arrow[r, "\alpha_g", "\sim"'] \arrow[ur, phantom, "\ulcorner"] & Eg
\end{tikzcd}
\tag{11}
$$

Lemma 18. *The triple (L,E,R) constructed in (10) and (11) is functorial factorisation on $\mathcal{C}$.*

By Remark 9, this functorial factorisation induces a pointed endofunctor (R,η) on $\mathcal{C}^{\mathbf{2}}$ where the component of η at f is given by the morphism $Lf\colon A\to Ef$ as depicted in (3). To extend this pointed endofunctor to a monad, all that remains is to define a suitable multiplication $\mu\colon R^2\Rightarrow R$.

To construct this multiplication, we first observe that the morphism $\alpha_f\colon Jf\to Ef$ constructed in (10) is a weak equivalence, and therefore the morphism $M\alpha_f\colon MJf\to MEf$ is invertible. It follows from the orthogonality property that the morphism $J\langle\alpha_f,1_B\rangle\colon JTf\to JRf$ is invertible as depicted below.

$$
\begin{tikzcd}
MJf \arrow[r, "M\alpha_f", "\cong"'] \arrow[d, "\iota_{Jf}"', "\wr"] & MEf \arrow[d, "\iota_{Ef}", "\wr"'] \\
Jf \arrow[r, "\alpha_f", "\sim"'] \arrow[d, tail, "Tf"'] & Ef \arrow[d, "Rf"] \\
B \arrow[r, equal] & B
\end{tikzcd}
=
\begin{tikzcd}
MJf \arrow[r, "M\alpha_f", "\cong"'] \arrow[d, two heads, "STf"'] & MEf \arrow[d, two heads, "SRf"] \\
JTf \arrow[r, dashed, "J\langle\alpha_f,1_B\rangle", "\cong"'] \arrow[d, tail, "T^2f"'] & JRf \arrow[d, tail, "TRf"] \\
B \arrow[r, equal] & B
\end{tikzcd}
\tag{12}
$$

Using the universal property of the pushout, the morphism ν_f defined in (5), and the morphism $J\langle \alpha_f, 1_B\rangle^{-1}$ defined in (12), we obtain the component μ_f of the multiplication $\mu\colon R^2 \Rightarrow R$ at f as depicted below.

$$
\begin{tikzcd}
MEf \arrow[r, "\iota_{Ef}", "\sim"'] \arrow[d, two heads, "SRf"'] & Ef \arrow[d, two heads, "LRf"'] \arrow[ddr, bend left, equal] & \\
JRf \arrow[r, "\alpha_{Rf}", "\sim"'] \arrow[d, "J\langle \alpha_f, 1_B\rangle^{-1}"', "\cong"] \arrow[ur, phantom, "\ulcorner"] & ERf \arrow[dr, dashed, two heads, "\mu_f"] & \\
JTf \arrow[r, tail, "\nu_f"] & Jf \arrow[r, "\alpha_f", "\sim"'] & Ef
\end{tikzcd}
=
\begin{tikzcd}
MEf \arrow[r, "\iota_{Ef}", "\sim"'] \arrow[d, two heads, "SRf"'] \arrow[dr, "(M\alpha_f)^{-1}", "\cong"'] & Ef \arrow[ddr, bend left, equal] & \\
JRf \arrow[d, "J\langle \alpha_f, 1_B\rangle^{-1}"', "\cong"] & MJf \arrow[dl, "STf"'] \arrow[d, "\iota_{Jf}", "\wr"'] & \\
JTf \arrow[r, tail, "\nu_f"] & Jf \arrow[r, "\alpha_f", "\sim"'] & Ef
\end{tikzcd}
\tag{13}
$$

A tedious, yet routine, exercise in diagram-chasing using the morphisms defined in (11) and (13), and applying the universal property of the pushout shows that $Rf \circ \mu_f = R^2 f$ and that μ is natural as depicted below.

$$
\begin{tikzcd}
ERf \arrow[r, two heads, "\mu_f"] \arrow[d, "R^2 f"'] & Ef \arrow[r, "E\langle h,k\rangle"] \arrow[d, "Rf"] & Eg \arrow[d, "Rg"] \\
B \arrow[r, equal] & B \arrow[r, "k"'] & D
\end{tikzcd}
=
\begin{tikzcd}
ERf \arrow[r, "E\langle E\langle h,k\rangle, k\rangle"] \arrow[d, "R^2 f"'] & ERg \arrow[r, two heads, "\mu_g"] \arrow[d, "R^2 g"] & Eg \arrow[d, "Rg"] \\
B \arrow[r, "k"'] & D \arrow[r, equal] & D
\end{tikzcd}
$$

Showing that the diagrams below commute, and thus establishing that the multiplication μ is unital and associative, is also a straightforward application of definitions and the universal property of the pushout.

$$
\begin{tikzcd}
Ef \arrow[r, two heads, "LRf"] \arrow[dr, equal] & ERf \arrow[d, two heads, "\mu_f"] & Ef \arrow[l, two heads, "E\langle Lf, 1_B\rangle"'] \arrow[dl, equal] \\
& Ef &
\end{tikzcd}
\qquad
\begin{tikzcd}
ER^2 f \arrow[r, two heads, "E\langle \mu_f, 1_B\rangle"] \arrow[d, two heads, "\mu_{Rf}"'] & ERf \arrow[d, two heads, "\mu_f"] \\
ERf \arrow[r, two heads, "\mu_f"'] & Ef
\end{tikzcd}
$$

Theorem 19. *The triple (R, η, μ) is a monad on $\mathcal{C}^{\mathbf{2}}$.*

Corollary 20. *The monad (R, η, μ) on $\mathcal{C}^{\mathbf{2}}$ restricts to a monad in the 2-category* $\mathcal{C}\mathrm{AT}/\mathcal{C}$ *on the codomain functor* $\mathrm{cod}\colon \mathcal{C}^{\mathbf{2}} \to \mathcal{C}$. *In particular, (R, η, μ) induces a monad on each slice category $\mathcal{C}/B$.*

Remark 21. The morphisms α_f defined as pushout injections in (10) assemble into a natural transformation $\alpha\colon T \Rightarrow R$ which underlies a morphism of semi-monads $(T, \nu) \to (R, \mu)$. We conjecture that (R, η, μ) is actually the *free monad* on the semi-monad (T, ν), in a suitable sense, however leave this for future work.

4.2 Delta lenses as monad algebras

We now construct the algebras for the monad (R, η, μ) on $\mathcal{C}^{\mathbf{2}}$ and show they are the same as JR-algebras for the semi-monad (T, ν). When $\mathcal{C} = \mathcal{C}\mathrm{at}$ equipped with the comprehensive factorisation system and the discrete category comonad, this result establishes that delta lenses are algebras for the monad (R, η, μ).

An *algebra* $(f, \hat{p})$ for the monad (R, η, μ) on $\mathcal{C}^{\mathbf{2}}$ consists of a pair of morphisms $f\colon A \to B$ and $\hat{p}\colon Ef \to A$ such that the following diagrams commute:

$$
\begin{tikzcd}
A \arrow[r, equal] \arrow[d, two heads, "Lf"'] & A \arrow[d, "f"] \\
Ef \arrow[ur, two heads, "\hat{p}"] \arrow[r, "Rf"'] & B
\end{tikzcd}
\qquad
\begin{tikzcd}
ERf \arrow[r, two heads, "E\langle \hat{p}, 1_B\rangle"] \arrow[d, two heads, "\mu_f"'] & Ef \arrow[d, two heads, "\hat{p}"] \\
Ef \arrow[r, two heads, "\hat{p}"'] & A
\end{tikzcd}
\tag{14}
$$

A *morphism* $\langle h,k\rangle\colon (f,\hat{p}) \to (g,\hat{q})$ of algebras for the monad (R,η,μ) consists of a pair of morphisms h and k such that the following equation in $\mathcal{C}^{\mathbf{2}}$ holds.

$$\begin{tikzcd} Ef \arrow[r, two heads, "\hat{p}"] \arrow[d, "Rf"'] & A \arrow[r, "h"] \arrow[d, "f"] & C \arrow[d, "g"] \\ B \arrow[r, equal] & B \arrow[r, "k"'] & D \end{tikzcd} \quad = \quad \begin{tikzcd} Ef \arrow[r, "E\langle h,k\rangle"] \arrow[d, "Rf"'] & Eg \arrow[r, two heads, "\hat{q}"] \arrow[d, "Rg"] & C \arrow[d, "g"] \\ B \arrow[r, "k"'] & D \arrow[r, equal] & D \end{tikzcd} \tag{15}$$

Let $\mathcal{A}\mathrm{lg}(R,\eta,\mu)$ denote the category of algebras for the monad (R,η,μ).

Proposition 22. *There is an isomorphism of categories $\mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu) \cong \mathcal{A}\mathrm{lg}(R,\eta,\mu)$.*

Proof. Let $(f\colon A \to B, p\colon Jf \to A)$ be a JR-algebra for the semi-monad (T,ν). Using the diagram (8) and the universal property of the pushout, we obtain a morphism $[p,1_A]\colon Ef \to A$ as depicted below.

$$\begin{tikzcd} MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] & A \arrow[d, two heads, "Lf"] \arrow[ddr, bend left, "1_A"] & \\ Jf \arrow[r, "\alpha_f", "\sim"'] \arrow[drr, bend right, "p"'] & Ef \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, dashed, "{[p,1_A]}"] & \\ & & A \end{tikzcd}$$

Using the universal property of the pushout and the axioms for the JR-algebra (f,p), it is straightforward to prove that the pair $(f,[p,1_A])$ is an algebra for the monad (R,η,μ).

Now consider an algebra $(f\colon A \to B, \hat{p}\colon Ef \to A)$ for the monad (R,η,μ). Pre-composing the structure map of the algebra with α_f we obtain a morphism $\hat{p}\circ\alpha_f\colon Jf \to B$. Using the axioms for the algebra $(f,\hat{p})$ and appropriate pasting of commutative diagrams, one may easily show that the pair $(f,\hat{p}\circ\alpha_f)$ is a JR-algebra for the semi-monad (T,ν).

The JR-algebras for (T,ν) and the algebra for (R,η,μ) are in bijective correspondence with each other, since $[\hat{p}\circ\alpha_f,1_A] = \hat{p}$ by the universal property of the pushout, and $[p,1_A]\circ\alpha_f = p$ by construction. One may extend this correspondence to the morphisms (9) and (15) of the respective categories and show it is functorial, thus establishing the stated isomorphism of categories. □

The following theorem establishes a key result of the paper: delta lenses are algebras for a monad.

Theorem 23. *There is an isomorphism of categories $\mathcal{L}\mathrm{ens} \cong \mathcal{A}\mathrm{lg}(R,\eta,\mu)$.*

Proof. Follows directly from Theorem 17 and interpreting Proposition 22 in the setting of $\mathcal{C} = \mathcal{C}\mathrm{at}$ equipped with the discrete category comonad and the comprehensive factorisation system. □

Corollary 24. *The forgetful functor $U\colon \mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}^{\mathbf{2}}$ is strictly monadic.*

4.3 The free delta lens on a functor

We now construct a left adjoint to the functor $U\colon \mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}^{\mathbf{2}}$ which defines the *free delta lens* on a functor $f\colon A \to B$. This amounts to providing an explicit description of the category Ef together with a lifting operation on the functor $Rf\colon Ef \to B$. First we recall [7, Corollary 20] the following result which represents an delta lens as a certain commutative diagram (see [11, Section 2.4] for a detailed proof).

Proposition 25. *Each delta lens $(f,\varphi)\colon A \to B$ determines a commutative diagram in $\mathcal{C}\mathrm{at}$, as depicted on the left below, such that φ is bijective-on-objects and $f\varphi$ is a discrete opfibration.*

$$\begin{tikzcd} & \Lambda(f,\varphi) \arrow[dl, "\varphi"', "\sim"] \arrow[dr, tail, "f\varphi"] & \\ A \arrow[rr, "f"'] & & B \end{tikzcd} \qquad \begin{tikzcd} & X \arrow[dl, "g"', "\sim"] \arrow[dr, tail, "fg"] & \\ A \arrow[rr, "f"'] & & B \end{tikzcd}$$

Conversely, each commutative diagram on the right above, where g is bijective-on-objects and fg is a discrete opfibration, uniquely determines a delta lens structure on f.

Remark 26. The above result may be understood as a consequence of an equivalence of double categories [11, Section 3.4], however the details are outside the scope of this paper.

Using Proposition 25, the *free delta lens* on a functor $f\colon A \to B$ corresponds to the following commutative diagram in $\mathcal{C}\mathrm{at}$ constructed in (10). An immediate benefit of this presentation of the free delta lens is that it condenses the three commutative diagrams (14) for the (free) R-algebra to a single diagram.

$$\begin{tikzcd} & Jf \arrow[dl, "\alpha_f"', "\sim"] \arrow[dr, tail, "Tf"] & \\ Ef \arrow[rr, "Rf"'] & & B \end{tikzcd}$$

In Example 15, we unpacked the definition of the category Jf and the discrete opfibration Tf. We now provide an explicit characterisation of the category Ef and the delta lens structure on $Rf\colon Ef \to B$.

Example 27. The objects of Ef are pairs $(a \in A, u\colon fa \to b \in B)$. The morphisms are generated by pairs $\langle w, fw\rangle\colon (a, 1_{fa}) \to (a', 1_{fa'})$ and $\langle 1_a, v\rangle\colon (a,u) \to (a, v\circ u)$ for $w \in A$ and $v \in B$, respectively, as depicted below. The identity morphisms are well-defined since $f(1_a) = 1_{fa}$. As Jf has the same objects as Ef and consists of morphisms of the form $\langle 1_a, v\rangle$, the functor $\alpha_f\colon Jf \to Ef$ is identity-on-objects and faithful.

$$\begin{array}{cc} \begin{tikzcd} a \arrow[r, "w"] & a' \\ fa \arrow[r, "f(w)"] \arrow[d, "1_{fa}"'] & fa' \arrow[d, "1_{fa'}"] \\ fa \arrow[r, "fw"'] & fa' \end{tikzcd} & \begin{tikzcd} a \arrow[r, "1_a"] & a \\ fa \arrow[r, "f(1_a)"] \arrow[d, "u"'] & fa \arrow[d, "v\circ u"] \\ b \arrow[r, "v"'] & b' \end{tikzcd} \end{array} \tag{16}$$

The functor $Rf\colon Ef \to B$ is projection in the second component; on the generators this is given by $Rf\langle w, fw\rangle = fw$ and $Rf\langle 1_a, v\rangle = v$. The lifting operation on Rf takes an object (a,u) in Ef and a morphism $v\colon \mathrm{cod}(u) \to b$ in B to the chosen lift $\langle 1_a, v\rangle\colon (a,u) \to (a, v\circ u)$ in Ef.

Although, in principle, the morphisms in Ef are finite sequences of the generators (16), one may show that each morphism $(a_1,u_1) \to (a_2,u_2)$ is actually just one of the following two kinds depicted below: either a retraction v of u_1 followed by morphism $w\colon a_1 \to a_2$, or a morphism $v\colon \mathrm{cod}(u_1) \to \mathrm{cod}(u_2)$ such that $v\circ u_1 = u_2$. The functor Rf sends these morphisms to $u_2 \circ fw \circ v$ and v, respectively.

$$\begin{tikzcd} a_1 \arrow[r, equal] & a_1 \arrow[r, "w"] & a_2 \arrow[r, equal] & a_2 \\ fa_1 \arrow[r, equal] \arrow[d, "u_1"'] \arrow[dr, phantom, "\circlearrowright"] & fa_1 \arrow[r, "f(w)"] \arrow[d, "1"'] & fa_2 \arrow[r, equal] \arrow[d, "1"] & fa_2 \arrow[d, "u_2"] \\ b_1 \arrow[r, "v"'] & fa_1 \arrow[r, "fw"'] & fa_2 \arrow[r, "u_2"'] & b_2 \end{tikzcd} \qquad \begin{tikzcd} a_1 \arrow[r, equal] & a_2 \\ fa_1 \arrow[r, equal] \arrow[d, "u_1"'] \arrow[dr, phantom, "\circlearrowright"] & fa_2 \arrow[d, "u_2"] \\ b_1 \arrow[r, "v"'] & b_2 \end{tikzcd}$$

5 Delta lenses as the R-algebras of an algebraic weak factorisation system

In this section, let $(\mathcal{E},\mathcal{M})$ be an orthogonal factorisation system on a category $\mathcal{C}$ with (chosen) pushouts, and let (M,ι) be an idempotent comonad on $\mathcal{C}$ such that $M\colon \mathcal{C}\to\mathcal{C}$ preserves pushouts.

5.1 Constructing the AWFS for delta lenses

Thus far we have constructed a functorial factorisation (L,E,R) on $\mathcal{C}$ (Lemma 18), and extended the pointed endofunctor (R,η) to a monad (R,η,μ) on $\mathcal{C}^{\mathbf{2}}$ (Theorem 19). We now show that the copointed endofunctor (L,ε) extends to a comonad (L,ε,Δ), therefore completing the data required to describe an algebraic weak factorisation system on $\mathcal{C}$. For $\mathcal{C}=\mathcal{C}at$ equipped with the comprehensive factorisation system and the discrete category monad, this yields an AWFS whose R-algebras are precisely delta lenses.

First we construct the morphism $L^2f\colon A\to ELf$ as on the left below. Using this diagram and (10), it follows that $TLf\circ SLf = Lf\circ\iota_A = \alpha_f\circ Sf$ and there is solid commutative diagram as on the right below. By the orthogonality property, there exists a unique morphism $\delta_f\colon Jf\to JLf$ as shown.

$$
\begin{tikzcd}
MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "SLf"'] & A \arrow[d, two heads, "L^2 f"] \arrow[dd, bend left=60, "Lf"] \\
JLf \arrow[r, "\alpha_{Lf}", "\sim"'] \arrow[d, tail, "TLf"'] \arrow[ur, phantom, "\ulcorner", very near end] & ELf \arrow[d, "RLf"] \\
Ef \arrow[r, equal] & Ef
\end{tikzcd}
\qquad
\begin{tikzcd}
MA \arrow[r, two heads, "SLf"] \arrow[d, two heads, "Sf"'] & JLf \arrow[d, tail, "TLf"] \\
Jf \arrow[r, "\alpha_f", "\sim"'] \arrow[ur, dashed, two heads, "\delta_f"] & Ef
\end{tikzcd}
\tag{17}
$$

Using the diagrams (17) and the universal property of the pushout, we obtain the component Δ_f of the comultiplication $\Delta\colon L\Rightarrow L^2$ at f as depicted below. For each morphism $\langle h,k\rangle\colon f\to g$ in $\mathcal{C}^{\mathbf{2}}$, we may show that $\Delta_g\circ E\langle h,k\rangle = E\langle h,E\langle h,k\rangle\rangle\circ\Delta_f$, providing us with a well-defined transformation $\Delta\colon L\Rightarrow L^2$.

$$
\begin{tikzcd}
MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] & A \arrow[d, two heads, "Lf"] \arrow[ddr, bend left, two heads, "L^2 f"] & \\
Jf \arrow[r, "\alpha_f", "\sim"'] \arrow[dr, two heads, "\delta_f"'] \arrow[ur, phantom, "\ulcorner", very near end] & Ef \arrow[dr, dashed, two heads, "\Delta_f"] & \\
& JLf \arrow[r, "\alpha_{Lf}", "\sim"'] & ELf
\end{tikzcd}
=
\begin{tikzcd}
MA \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] \arrow[ddr, bend left, two heads, "SLf"] & A \arrow[ddr, bend left, two heads, "L^2 f"] & \\
Jf \arrow[dr, two heads, "\delta_f"'] & & \\
& JLf \arrow[r, "\alpha_{Lf}", "\sim"'] \arrow[ur, phantom, "\ulcorner", very near end] & ELf
\end{tikzcd}
$$

Showing that the diagrams below commute, and thus establishing that the comultiplication Δ is counital and coassociative, is a straightforward application of definitions and the universal property of a pushout.

$$
\begin{tikzcd}
& Ef \arrow[dl, equal] \arrow[d, "\Delta_f"] \arrow[dr, equal] & \\
Ef & ELf \arrow[l, "RLf"] \arrow[r, "\langle 1_A, Rf\rangle"'] & Ef
\end{tikzcd}
\qquad
\begin{tikzcd}
Ef \arrow[r, "\Delta_f"] \arrow[d, "\Delta_f"'] & ELf \arrow[d, "\Delta_{Lf}"] \\
ELf \arrow[r, "E\langle 1_A, \Delta_f\rangle"'] & EL^2 f
\end{tikzcd}
$$

Proposition 28. *The triple (L,ε,Δ) is a comonad on $\mathcal{C}^{\mathbf{2}}$.*

Theorem 29. *The pair (L,R) is an algebraic weak factorisation system on $\mathcal{C}$.*

Proof. The data of the algebraic weak factorisation system follows from Lemma 18, Theorem 19, and Proposition 28. Checking that there is a distributive law $\lambda\colon LR\Rightarrow RL$ of the comonad L over the monad R with components $\lambda_f=\langle\Delta_f,\mu_f\rangle$ involves routine diagram-chasing and applying universal properties. □

5.2 Coalgebras and lifting

A *coalgebra* (f,q) for the comonad (L,ε,Δ) consists of a pair of morphisms $f\colon A \to B$ and $q\colon B \to Ef$ such that the following diagrams commute:

$$\begin{tikzcd} A \arrow[r, two heads, "Lf"] \arrow[d, "f"'] & Ef \arrow[d, "Rf"] \\ B \arrow[ur, "q"] \arrow[r, equal] & B \end{tikzcd} \qquad \begin{tikzcd} B \arrow[r, "q"] \arrow[d, "q"'] & Ef \arrow[d, "\Delta_f"] \\ Ef \arrow[r, "E\langle 1_A, q\rangle"'] & ELf \end{tikzcd}$$

Remark 30. In contrast to the algebras for the monad (R,η,μ), the coalgebras above cannot be easily simplified since q is a morphism *into* a pushout. For $\mathcal{C} = \mathcal{C}\mathrm{at}$, one may show that for a functor f to admit a coalgebra structure, it must be a left-adjoint-right-inverse (LARI) and is therefore also injective-on-objects and fully faithful. A complete characterisation of the L-coalgebras is left for future work.

We now provide a simple diagrammatic proof that delta lenses, in the form of Proposition 25 rather than as R-algebras, lift against L-coalgebras. Consider a morphism $\langle h,k\rangle\colon f \to g$ such that (f,q) is an L-coalgebra and (g,ψ) is a delta lens. Since ψ is bijective-on-objects, ψ_0 is invertible, and there is a morphism $\iota_\Lambda \circ \psi_0^{-1} \circ h_0\colon A_0 \to \Lambda(g,\psi)$ making the diagram, depicted below, commute. Then by the orthogonality property, there exists a unique morphism $\ell\colon Jf \to \Lambda(g,\psi)$ such that $\ell \circ Sf = \iota_\Lambda \circ \psi_0^{-1} \circ h_0$ and $g \circ \psi \circ \ell = k \circ Rf \circ \alpha_f$. Finally, by the universal property of the pushout, there exists a unique morphism $[\psi \circ \ell, h]\colon Ef \to C$. Thus, there is a specified morphism $q \circ [\psi \circ \ell, h]\colon B \to C$ as on the left below.

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] & C \arrow[d, "g"] \\ B \arrow[ur, "q \circ [\psi\circ\ell, h]"] \arrow[r, "k"'] & D \end{tikzcd} \qquad \begin{tikzcd} & & \Lambda(g,\psi) \arrow[d, "\psi", "\wr"'] \arrow[dd, bend left, "g\circ\psi"] \\ A_0 \arrow[urr, dashed, bend left, "\iota_\Lambda \circ \psi_0^{-1} \circ h_0"] \arrow[r, "\iota_A", "\sim"'] \arrow[d, two heads, "Sf"'] & A \arrow[r, "h"] \arrow[d, two heads, "Lf"] & C \arrow[d, "g"] \\ Jf \arrow[r, "\alpha_f", "\sim"'] & Ef \arrow[ul, phantom, "\ulcorner", very near start] \arrow[ur, dashed, "\psi\circ l, h"] \arrow[r, "k\circ Rf"'] & D \end{tikzcd}$$

Therefore we have shown that delta lenses lift against functors with the structure of a L-coalgebra, which is stronger than one would expect from their simple axiomatic definition. It also demonstrates how the notion of lifting is intrinsic to delta lenses as the R-algebras of an AWFS. The sequential composition of delta lenses as R-algebras may also be defined from this notion of lifting against L-coalgebras, providing further clarification of this essential operation.

6 Concluding remarks and future work

In this paper, we have shown that delta lenses are algebras for a monad (R,η,μ), and that this monad arises from an algebraic weak factorisation system on $\mathcal{C}\mathrm{at}$. Moreover, we have shown that this AWFS exists on any suitable category equipped with an orthogonal factorisation system and an idempotent comonad which preserves pushouts. These results generalise immediately to *internal lenses* [7, 8] using the internal comprehensive factorisation system [30], however an analogous result for *enriched lenses* [12] or *weighted lenses* [27] is unknown. There are many avenues for future work. One example is the relationship between the *proxy pullbacks* [14] of delta lenses and the canonical pullback of R-algebras [4]. Another is the connection between spans of delta lenses [9] and the categories of weak maps for an

AWFS [5]. The *double category of delta lenses* [11], which is naturally induced by the AWFS, provides a rich setting studying the properties of delta lenses previously considered in a 1-categorical setting [6, 15].

Acknowledgements. This research first appeared in Chapter 6 of my PhD thesis [11], and I would like to thank my supervisor, Michael Johnson, for his helpful feedback during my PhD studies. I am also very grateful to Richard Garner who first sketched the construction of the free delta lens, and also suggested the approach using AWFS, after I presented this work at the Australian Category Seminar. I would also like to thank John Bourke, Matthew Di Meglio, Tim Hosgood, and Noam Zeilberger for sharing useful insights which contributed to the development of this research.

References

[1] Faris Abou-Saleh, James Cheney, Jeremy Gibbons, James McKinna & Perdita Stevens (2018): *Introduction to Bidirectional Transformations*. In Jeremy Gibbons & Perdita Stevens, editors: *Bidirectional Transformations*, *Lecture Notes in Computer Science* 9715, pp. 1–28, doi:10.1007/978-3-319-79108-1_1.

[2] Marcelo Aguiar (1997): *Internal categories and quantum groups*. Ph.D. thesis, Cornell University. Available at `http://pi.math.cornell.edu/~maguiar/thesis2.pdf`.

[3] Danel Ahman & Tarmo Uustalu (2017): *Taking Updates Seriously*. In Romina Eramo & Michael Johnson, editors: *Proceedings of the 6th International Workshop on Bidirectional Transformations*, *CEUR Workshop Proceedings* 1827, pp. 59–73. Available at `https://ceur-ws.org/Vol-1827/paper11.pdf`.

[4] John Bourke & Richard Garner (2016): *Algebraic weak factorisation systems I: Accessible AWFS*. *Journal of Pure and Applied Algebra* 220(1), doi:10.1016/j.jpaa.2015.06.002.

[5] John Bourke & Richard Garner (2016): *Algebraic weak factorisation systems II: Categories of weak maps*. *Journal of Pure and Applied Algebra* 220(1), doi:10.1016/j.jpaa.2015.06.003.

[6] Emma Chollet, Bryce Clarke, Michael Johnson, Maurine Songa, Vincent Wang & Gioele Zardini (2022): *Limits and Colimits in a Category of Lenses*. In Kohei Kishida, editor: *Proceedings of the Fourth International Conference on Applied Category Theory*, *Electronic Proceedings in Theoretical Computer Science* 372, pp. 164–177, doi:10.4204/EPTCS.372.12.

[7] Bryce Clarke (2020): *Internal lenses as functors and cofunctors*. In John Baez & Bob Coecke, editors: *Proceedings Applied Category Theory 2019*, *Electronic Proceedings in Theoretical Computer Science* 323, pp. 183–195, doi:10.4204/EPTCS.323.13.

[8] Bryce Clarke (2020): *Internal split opfibrations and cofunctors*. *Theory and Applications of Categories* 35(44). Available at `http://www.tac.mta.ca/tac/volumes/35/44/35-44abs.html`.

[9] Bryce Clarke (2021): *Delta Lenses as Coalgebras for a Comonad*. In Leen Lambers & Meng Wang, editors: *9th International Workshop on Bidirectional Transformations*, *CEUR Workshop Proceedings* 2999, pp. 18–27. Available at `https://ceur-ws.org/Vol-2999/bxpaper2.pdf`.

[10] 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*, *Electronic Proceedings in Theoretical Computer Science* 333, pp. 79–91, doi:10.4204/EPTCS.333.6.

[11] Bryce Clarke (2022): *The double category of lenses*. Ph.D. thesis, Macquarie University, doi:10.25949/22045073.v1.

[12] Bryce Clarke & Matthew Di Meglio (2022): *An introduction to enriched cofunctors*. arXiv:2209.01144.

[13] Krzysztof Czarnecki, J. Nathan Foster, Zhenjiang Hu, Ralf Lämmel, Andy Schürr & James F. Terwilliger (2009): *Bidirectional Transformations: A Cross-Discipline Perspective*. In Richard F. Paige, editor: *Theory and Practice of Model Transformations*, *Lecture Notes in Computer Science* 5563, pp. 260–283, doi:10.1007/978-3-642-02408-5_19.

[14] Matthew Di Meglio (2021): *The category of asymmetric lenses and its proxy pullbacks*. Master's thesis, Macquarie University, doi:10.25949/20236449.v1.

[15] Matthew Di Meglio (2022): *Coequalisers under the Lens*. In Kohei Kishida, editor: *Proceedings of the Fourth International Conference on Applied Category Theory*, *Electronic Proceedings in Theoretical Computer Science* 372, pp. 149–163, doi:10.4204/EPTCS.372.11.

[16] Zinovy Diskin & Tom Maibaum (2012): *Category Theory and Model-Driven Engineering: From Formal Semantics to Design Patterns and Beyond*. In Ulrike Golas & Thomas Soboll, editors: *Proceedings Seventh ACCAT Workshop on Applied and Computational Category Theory*, *Electronic Proceedings in Theoretical Computer Science* 93, pp. 1–21, doi:10.4204/EPTCS.93.1.

[17] Zinovy Diskin, Yingfei Xiong & Krzysztof Czarnecki (2011): *From State- to Delta-Based Bidirectional Model Transformations: the Asymmetric Case*. *Journal of Object Technology* 10(6), doi:10.5381/jot.2011.10.1.a6.

[18] William G. Dwyer, Philip S. Hirschhorn, Daniel M. Kan & Jeffrey H. Smith (2004): *Homotopy Limit Functors on Model Categories and Homotopical Categories*. *Mathematical Surveys and Monographs* 113, American Mathematical Society, doi:10.1090/surv/113.

[19] J. Nathan Foster, Michael B. Greenwald, Jonathan T. Moore, Benjamin C. Pierce & Alan Schmitt (2007): *Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem*. *ACM Transactions on Programming Languages and Systems* 29(3), doi:10.1145/1232420.1232424.

[20] P.J. Freyd & G.M. Kelly (1972): *Categories of continuous functors I*. *Journal of Pure and Applied Algebra* 2(3), doi:10.1016/0022-4049(72)90001-1.

[21] Marco Grandis & Walter Tholen (2006): *Natural weak factorization systems*. *Archivum mathematicum* 42(4). Available at `https://www.emis.de/journals/AM/06-4/tholen.pdf`.

[22] Philip J. Higgins & Kirill C. H. Mackenzie (1993): *Duality for base-changing morphisms of vector bundles, modules, Lie algebroids and Poisson structures*. *Mathematical Proceedings of the Cambridge Philosophical Society* 114(3), doi:10.1017/S0305004100071760.

[23] Michael Johnson & Robert Rosebrugh (2013): *Delta Lenses and Opfibrations*. In Perdita Stevens & James F. Terwilliger, editors: *Proceedings of the Second International Workshop on Bidirectional Transformations*, *Electronic Communications of the EASST* 57, pp. 1–18, doi:10.14279/tuj.eceasst.57.875.

[24] Michael Johnson & Robert Rosebrugh (2016): *Unifying Set-Based, Delta-Based and Edit-Based Lenses*. In Anthony Anjorin & Jeremy Gibbons, editors: *Proceedings of the Fifth International Workshop on Bidirectional Transformations*, *CEUR Workshop Proceedings* 1571, pp. 1–13. Available at `https://ceur-ws.org/Vol-1571/paper_13.pdf`.

[25] Michael Johnson, Robert Rosebrugh & R. J. Wood (2012): *Lenses, fibrations and universal translations*. *Mathematical Structures in Computer Science* 22(1), doi:10.1017/S0960129511000442.

[26] Michael Johnson, Robert Rosebrugh & Richard Wood (2010): *Algebras and Update Strategies*. *Journal of Universal Computer Science* 16(5), doi:10.3217/jucs-016-05-0729.

[27] Paolo Perrone (2021): *Lifting couplings in Wasserstein spaces*. arXiv:2110.06591.

[28] Alberto Rodrigues da Silva (2015): *Model-driven engineering: A survey supported by the unified conceptual model*. *Computer Languages, Systems & Structures* 43, doi:10.1016/j.cl.2015.06.001.

[29] Ross Street (1974): *Fibrations and Yoneda's lemma in a 2-category*. In G. M. Kelly, editor: *Category Seminar*, *Lecture Notes in Mathematics* 420, pp. 104–133, doi:10.1007/BFb0063102.

[30] Ross Street & Dominic Verity (2010): *The comprehensive factorization and torsors*. *Theory and Applications of Categories* 23(3). Available at `http://www.tac.mta.ca/tac/volumes/23/3/23-03abs.html`.

[31] Ross Street & R. F. C. Walters (1973): *The comprehensive factorization of a functor*. *Bulletin of the American Mathematical Society* 79(5), doi:10.1090/S0002-9904-1973-13268-9.

A Appendix

In this section, we provide a proof of Theorem 17. The correspondence between JR-algebras and delta lenses was first shown by Johnson and Rosebrugh [23, Proposition 3]; we reprove this correspondence in our notation, and extend it to an isomorphism of categories. We refer the reader to Example 15 for an explicit description of the category Jf and the functor $Tf\colon Jf \to B$.

Theorem 31. *If $\mathcal{C} = \mathcal{C}\mathrm{at}$ equipped with the discrete category comonad and the comprehensive factorisation system, then there is an isomorphism of categories $\mathcal{L}\mathrm{ens} \cong \mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$.*

We prove this theorem in two parts: first defining the functor $\mathcal{L}\mathrm{ens} \to \mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$, then defining the functor $\mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu) \to \mathcal{L}\mathrm{ens}$ and showing that they are mutually inverse.

Proof. We begin by constructing a functor $\mathcal{L}\mathrm{ens} \to \mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$.

Given a delta lens $(f,\varphi)\colon A \to B$ as in Definition 1, we define a functor $p\colon Jf \to B$ whose assignment on morphisms $\langle 1_a, v\rangle\colon (a,u_1) \to (a,u_2)$ is given below, where we recall that $p(a,u) = \mathrm{cod}(\varphi(a,u))$.

$$\begin{tikzcd} a \arrow[r, equal] & a \\ fa \arrow[r, equal] \arrow[d, "u_1"'] \arrow[dr, phantom, "\circlearrowright"] & fa \arrow[d, "u_2"] \\ b_1 \arrow[r, "v"] & b_2 \end{tikzcd} \qquad \longmapsto \qquad p(a,u_1) \xrightarrow{\varphi(p(a,u_1),v)} p(a,u_2) \tag{18}$$

This functor preserves identities and composition by the axioms (L2) and (L3) of a delta lens, respectively. Moreover, the equation $f \circ p = Tf$ from the left diagram of (7) is satisfied by axiom (L1). The equation $p \circ Sf = \iota_A$ from the diagram (8) also holds since $Sf(a) = (a, 1_{fa})$ and $p(a,1_{fa}) = a$ by axiom (L2).

To verify the remaining condition for a JR-algebra given by the right diagram of (7), we first describe the category JTf and the functors $\nu_f, \langle p, 1_B\rangle\colon JTf \to Jf$.

The category JTf has objects given by triples $(a \in A, u\colon fa \to b, u'\colon b \to b')$ and morphisms given by triples $\langle 1_a, 1_b, v\rangle$ as depicted below. The functor ν_f has an assignment on objects $(a,u,u') \mapsto (a, u' \circ u)$ and an assignment on morphisms $\langle 1_a, 1_b, v\rangle \mapsto \langle 1_a, v\rangle$, while the functor $\langle p, 1_B\rangle$ has corresponding assignments on objects and morphisms given by $(a,u,u') \mapsto (p(a,u), u')$ and $\langle 1_a, 1_b, v\rangle \mapsto \langle 1_{p(a,u)}, v\rangle$ which are well-defined by (L1). The equation $p \circ \mu_f = p \circ \langle p, 1_B\rangle$ holds since $p(a, u' \circ u) = p(p(a,u), u')$ by axiom (L3). Therefore, we have a JR-algebra (f,p) and the functor $\mathcal{L}\mathrm{ens} \to \mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$ is well-defined on objects.

$$\begin{tikzcd} a \arrow[r, equal] & a \\ fa \arrow[r, equal] \arrow[d, "u"'] & fa \arrow[d, "u"] \\ b \arrow[r, equal] \arrow[d, "u'_1"'] \arrow[dr, phantom, "\circlearrowright"] & b \arrow[d, "u'_2"] \\ b'_1 \arrow[r, "v"] & b'_2 \end{tikzcd} \tag{19}$$

Consider a pair of delta lenses $(f,\varphi)\colon A \to B$ and $(g,\psi)\colon C \to D$ with corresponding JR-algebras (f,p) and (g,q), respectively. Given a morphism of delta lenses $\langle h,k\rangle\colon (f,\varphi) \to (g,\psi)$, we want to show that there is a morphism of JR-algebras $\langle h,k\rangle\colon (f,p) \to (g,q)$. First note that the functor

$J\langle h,k\rangle\colon Jf \to Jg$ has an assignment on objects $(a,u) \mapsto (ha,ku)$ and an assignment on morphisms $\langle 1_a, v\rangle \mapsto \langle 1_{ha}, kv\rangle$. As we have $h\varphi(a,u) = \psi(ha,ku)$ by the definition of a morphism of delta lenses, it follows that $hp(a,u) = \mathrm{cod}(h\varphi(a,u)) = \mathrm{cod}(\psi(ha,ku)) = q(ha,ku)$. A similar argument on morphisms of Ef establishes that $q \circ J\langle h,k\rangle = h \circ p$ and thus the equation (9) for a morphism of JR-algebras holds. □

Proof. We now construct a functor $\mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu) \to \mathcal{L}\mathrm{ens}$ and show that it is inverse to $\mathcal{L}\mathrm{ens} \to \mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$.

Given a JR-algebra determined by the pair of functors $f\colon A \to B$ and $p\colon Jf \to A$, we define a delta lens $(f,\varphi)\colon A \to B$ whose lifting operation φ is given below, where $p(a,1_{fa}) = a$ by (8).

$$\begin{tikzcd} a \arrow[r, equal] & a \\ fa \arrow[r, equal] \arrow[d, "1_{fa}"'] & fa \arrow[d, "u"] \\ fa \arrow[r, "u"] & b \end{tikzcd} \qquad \longmapsto \qquad p(a,1_{fa}) = a \xrightarrow{p\langle 1_a,u\rangle} p(a,u) \tag{20}$$

By (8) on morphisms, it follows that axiom (L2) for a delta lens holds. By the left diagram of (7), it is also immediate that axiom (L1) holds. For axiom (L3) to hold, we need to show that

$$p\langle 1_a, v\circ u\rangle = p\langle 1_a, v\rangle \circ p\langle 1_a,u\rangle = p\langle 1_{p(a,u)}, v\rangle \circ p\langle 1_a,u\rangle.$$

This amounts to proving that the morphism $p\langle 1_a, v\rangle\colon p(a,u) \to p(a, v\circ u)$ is equal to the morphism $p\langle 1_{p(a,u)}, v\rangle\colon p(p(a,u),1_b) \to p(p(a,u),v)$, which follows directly from the right diagram in (7).

Given a morphism of JR-algebras $\langle h,k\rangle\colon (f,p) \to (g,q)$, we have $hp\langle 1_a,u\rangle = q\langle 1_{ha},ku\rangle$ from (9). Therefore there is a well-defined morphism $\langle h,k\rangle$ between the corresponding delta lenses.

To show that the functors $\mathcal{L}\mathrm{ens} \to \mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu)$ and $\mathcal{A}\mathrm{lg}_{\mathsf{JR}}(T,\nu) \to \mathcal{L}\mathrm{ens}$ are inverse, it is enough to show it holds on the objects as the morphisms consist of the same data.

First consider a delta lens (f,φ) and define a functor $p\colon Jf \to B$ as in (18). Applying this functor at a morphism $\langle 1_a,u\rangle\colon (a,1_{fa}) \to (a,u)$ in Jf, we obtain $\varphi(p(a,1_{fa}),u) = \varphi(a,u)$ by (L2) as desired. Now consider a JR-algebra (f,p) and define a lifting operation φ for a delta lens as in (20). Defining a functor $\hat{p}\colon Jf \to A$ from this delta as in (18) and applying it to a morphism $\langle 1_a,v\rangle\colon (a,u_1) \to (a,u_2)$ we find that $\hat{p}\langle 1_a,v\rangle = p\langle 1_{p(a,u)},v\rangle = p\langle 1_a,v\rangle$ by the right diagram in (7) as desired. This completes the proof. □

Normalizing Resistor Networks

Robin Cockett
Department of Computer Science
University of Calgary
Alberta, Canada
robin@ucalgary.ca

Amolak Ratan Kalra
Institute for Quantum Computing (IQC)
Cheriton School of Computer Science
University of Waterloo
Ontario, Canada
arkalra@uwaterloo.ca

Priyaa Varshinee Srinivasan
University of Calgary
Alberta, Canada
National Institute of Standards and Technology,
Maryland, USA
priyaavarshinee@gmail.com

Star to mesh transformations are well-known in electrical engineering, and are reminiscent of local complementation for graph states in qudit stabilizer quantum mechanics. This paper describes a rewriting system for resistor circuits over any positive division rig using general star to mesh transformations. We show how these transformations can be organized into a confluent and terminating rewriting system on the category of resistor circuits. Furthermore, based on the recently established connections between quantum and electrical circuits, this paper pushes forward the quest for approachable normal forms for stabilizer quantum circuits.

1 Introduction

Electrical circuits are well-studied and, indeed, the basis of an eponymous engineering discipline. One would therefore expect that there is not much more that can be usefully said about the simplest and most basic of these circuits, namely circuits consisting of just resistors. However, it turns out that there is always more to say! Indeed, it seems possible that modern mathematical methods can even provide new insight into what is an old and well-studied subject. Furthermore, by considering resistor networks with resistance values in finite fields – which is not the most natural direction of generalization from an electrical engineering perspective – has a tantalizing connection to the theory of qudit stabilizer quantum mechanics.

A categorical description for electrical circuits was provided in Brendan Fong's thesis [8], described in a paper with John Baez [3], and was also the subject of Brandon Coya's thesis [7]. Following the work of Cole Comfort and Alex Kissinger [6], the current authors with Shiroman Prakash investigated the relationship between electrical circuits and quantum circuits [5]. There it was noted that the structure of parity check matrices arising from resistor networks and graph states are precisely the same. This work, in turn, relied on the developments of Graphical Linear Algebra [4] where it was realized that there was a natural encoding of resistors (and electrical circuits) into categories of linear relations.

Resistor networks (or circuits) form a hypergraph category, [9], which we call Resist: this is a symmetric monoidal category in which each object is a commutative Fröbenius algebra (coherent with the tensor product). It is an open issue as to whether the equality of maps in Resist can in general be resolved by a simple rewriting system [2]. We resolve this question in this article for resistor circuits over a positive division rig. Our rewriting system uses an important identity for electrical circuits of resistors called

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 70–83, doi:10.4204/EPTCS.397.5

the star/mesh or (Y/Δ) identity, which asserts that a "star-shaped" circuit (with n-points) is equivalent to a "mesh-shaped" circuit (on n points). This is a classical observation in electrical engineering with proofs going back almost a century [16].

While it is well-known that the $(Y/\Delta)_3$ transform for three nodes is a two-way identity, in the sense that any three pointed star can be transformed into a triangular mesh and conversely any such mesh can be transformed into a star, this fails for $n > 3$. It fails for a simple reason: meshes of resistors with n-nodes when $n > 3$ have more degrees of freedom than stars with n-nodes. Meshes on n-nodes have $n(n-1)/2$ resistors while stars have only n: only at $n = 3$ do they have the same number of resistors! Thus, it is not the case that *every* n-mesh is be equivalent to a n-star for $n > 3$. However, it does remain the case that (for every $n \in \mathbb{N}$) every n-star can be transformed into an equivalent n-mesh, thereby, suggesting a natural orientation for these identities.

Not surprisingly the normalizing procedure we introduce using the general star/mesh identity is oriented in the star to mesh direction: we prove that this forms part of a confluent rewriting system on the category of resistor circuits, Resist. Regrettably, this rewriting does not make the circuits more efficient in terms of hardware real-estate, however, it certainly does provide a simple, easily automated, decision procedure for equality. The resulting normal form for circuits is a family of meshes (with "extra inputs"); a form foreshadowed not only by the work in [8, 3, 7] but also in the work using parity matrices [5].

While this paper provides a rewriting system for resistor circuits in which the conductances are taken from an arbitrary *positive division rig*, a more desirable objective would be to show that our results hold for *all* division rigs. All fields, including finite fields, are examples of division rigs. Resistors over finite fields can be interpreted as (special) stabilizer quantum circuits [12]: thus, obtaining such a generalization would provide normal forms for these quantum circuits which in turn could possibly be generalized to arbitrary stabilizer circuits.

We choose positive rigs in our formulation because the normalization procedure demand division by sums, hence, in a non-positive rig one can possibly run into a division by zero situation during rewriting. Clearly, $\mathbb{R}_{>0}$, the usual "base" for resistors in electrical engineering is a positive division rig. For our normalization procedure, we have also chosen to using conductances rather than impedances (or resistances) of resistors in an attempt to simplify the calculations. However, the calculation in its impedance form may also provide some advantages as there is only one case in which division by a sum may occur — during a parallel rewrite. Unfortunately, simply switching to impedance form does not quite suffice to allow the generalization to arbitrary division rigs!

Reducing resistor networks using series, parallel, and (Y-Δ) transformations, or a combination of these to eliminate the internal nodes of the network is a well-studied problem in electrical engineering. Indeed, there exist several studies on general methods and efficient algorithms to automate the reduction process of large resistor networks [15, 10, 11, 14]. However, to the best of our knowledge, a terminating confluent rewriting system for such networks is yet to be found and have been suggested not to exist due to the inherent directionality of the reduction rules [2]. The main contributions of this paper are a categorical presentation of resistor networks as hypergraph categories, and a terminating confluent rewriting system for these networks based on star-mesh transformations.

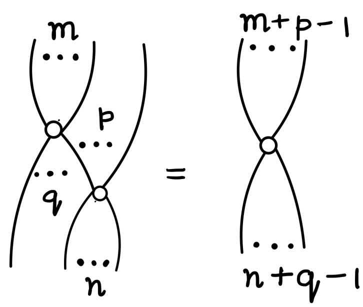
Notation: Throughout this paper composition is written in diagrammatic order: fg means apply f followed by g. The string diagrams are to be read from top to bottom (following the direction of gravity) or left to right.

2 Background

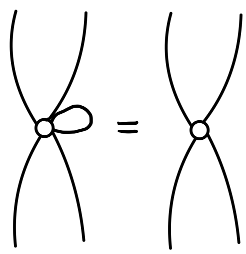
2.1 Spider Rewriting

A category is a **hypergraph category** in case it is a symmetric monoidal category in which every object is coherently a special commutative Fröbenius algebra, [9], see Appendix A for details. There is a well-known rewriting system on any hypergraph category, often called "spider" rewriting, which normalizes Fröbenius operations, $\circ^m_n$, called "spiders" which have m inputs and n outputs (whose order does not matter). Within this rewriting system, two spiders which are connected can then be amalgamated to form a bigger spider, see equation Spider-(a). The "special" rule allows loops, to be eliminated, see equation Spider-(b). The main rewriting rules for spiders are:

[Spider] (*a*)

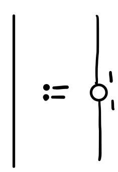


(*b*)



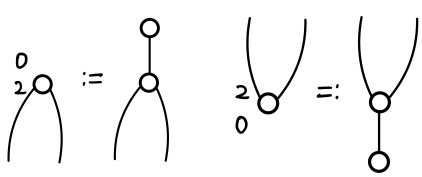
The following unusual rewrite is an *expansion* which replaces a wire with a wire with a $\circ^1_1$ junction:

[Expansion]

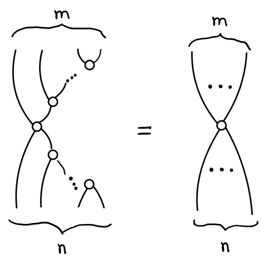


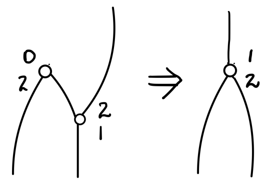
Clearly this, as a rewrite, can be performed indefinitely, however, an expansion is only used when there are no rewrites which can immediately undo it: thus, expansion rules are used only *irreducibly*. For example, an expansion of a wire on which there is already a spider can always be reduced and so is not irreducible.

The spiders $\circ^0_1$, $\circ^1_0$, and $\circ^1_1$ are given by the unit, the counit, and the identity maps respectively. The spiders $\circ^0_2$ and $\circ^2_0$ are given as follows:



A few examples of the spider rewriting are as follows:

(a) 

(b) 

2.2 Rigs

The category Resist_R is built atop a rig, R, which must satisfy some special properties: the elements of the rig will represent conductances (recall we shall work with conductances rather than impedances) as this makes for slightly simpler calculations. This means that the rule for amalgamating parallel resistors is achieved by simply adding their conductances. However, composing resistors in series then becomes more complicated and uses the ability to "divide".

Recall that a **rig** is a "ring without negatives" in the sense that, under addition it is a commutative monoid, and under multiplication a monoid. These operations must satisfy the distributive laws: $r\cdot(p+q) = r\cdot p + r\cdot q$, $(p+q)\cdot r = p\cdot r + q\cdot r$, and $0\cdot r = 0 = s\cdot 0$. Here we shall consider only **commutative** rigs in which $p\cdot q = q\cdot p$. The paradigmatic and initial rig is the rig of natural numbers $\mathbb{N}$.

A **division rig** is a rig in which all non-zero elements have a multiplicative inverse. Fields are clearly examples of division rigs as are the positive rationals, $\mathbb{Q}_{>0}$, and the positive reals, $\mathbb{R}_{>0}$. Furthermore, the two element lattice with join as addition and meet as multiplication is also a division rig.

A rig is **positive** if $x+y=0$ implies that both $x=0$ and $y=0$. $\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$, and the two element lattice are all positive division rigs in this sense. However, fields and, in particular, finite fields are not positive rigs.

Another important example of a positive division rig is the so called "tropical" rig, $(\mathbb{R}\cup\{-\infty\},\vee,+)$, where the addition of the rig uses the maximum (with unit $-\infty$) and multiplication uses addition (with $-\infty$ as a zero).

Below we show how to build a category of resistors based on a positive division rig, R.

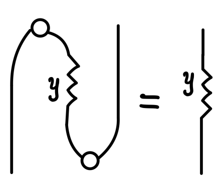
3 The category Resist_R

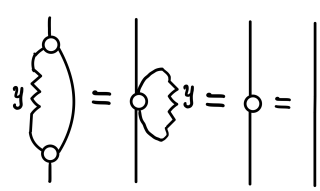
Definition 3.1. The category Resist_R, where R is a positive division rig, is a hypergraph category that consists of:

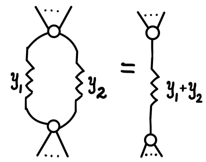
Objects: Natural numbers

Maps: Generated by **conductances** $y : 1 \longrightarrow 1$ where $y \in R$ with $y \neq 0$ and "junctions" $\circ_m^n$.

Together with the [Spider] rules, the maps satisfy the following three identities (and a family of star-mesh identities):

[Self − adjoint] 

[Short circuit] 

[Parallel] 

Thus, resistors are self-adjoint. The [Short circuit] rule states that if there is an infinite conductance in parallel with a conductance of any finite value, the current would take the path of "least resistance" can flow through the infinite conductance wire. The [Parallel rule] is used to collapse a number of parallel conductance into one conductance. A wire may be thought of as a resistor with infinite conductance.

The resistors in addition satisfy a family of *star-mesh identities*, $(Y/\Delta)_n$. Each of these identities equates a star resistor network, that is a network which has one *internal node* (a node which has connections only within the circuit) to a completely connected graph of resistors with no internal nodes which is referred to as a *mesh*.

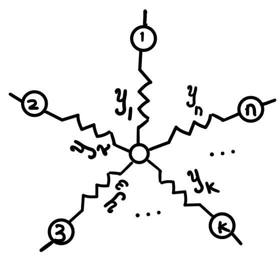
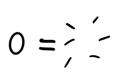


Figure 1: n-node star network

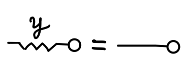
Given an n-node star network as shown in Figure 1, the corresponding mesh network consists of a completely connected graph with nodes $1,2,3,\cdots,n$ in which the edge between each pair of nodes i and j have conductance value,

$$Y_{ij} = \frac{y_i y_j}{\sum_{k=1}^{n} y_k} \tag{3.1}$$

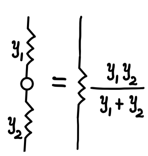
A few special cases of the star-mesh transformation are shown below:



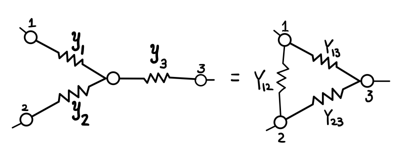
$(Y/\Delta)_0$



$(Y/\Delta)_1$



$(Y/\Delta)_2$



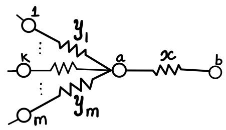
$(Y/\Delta)_3$

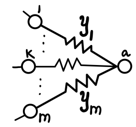
4 Rewriting for Resist_R

Given any resistor circuit with non-zero conductance on each wire, one can reduce the circuit to a family of meshes, thus to a normal form, by removing all the parallel resistors and the internal nodes using the identities of Resist_R. Our goal in this section is to prove that the resulting rewriting systems terminates and is confluent, which is the main result of this paper. In order to prove the main result, we observe the following.

Lemma 4.1. *The unit and star-mesh critical pairs in* Resist_R *resolve.*

Proof. Consider the star network composed on one of its outgoing edges with the unit, see circuit (a).

(a) 

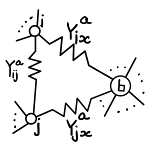
(b) 

To rewrite circuit (a), node b shall be eliminated first by applying $(\mathrm{Y}/\Delta)_1$ or node a shall be eliminated first by applying $(\mathrm{Y}/\Delta)_{m+1}$, thereby resulting in a critical pair.

Resolving the node b first by applying $(\mathrm{Y}/\Delta)_1$ results in an m-node star network (with one internal node a), see circuit (b) above. Applying $(Y/\Delta)_m$ to circuit (b) to eliminate node a results in a mesh in which for each pair of nodes $1 \le i, j \le m$, the resistor edge connecting them has conductance Y_{ij}^{ba} with value:

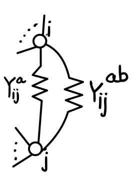
$$Y_{ij}^{ba} = \frac{y_i y_j}{\sum_k^m y_k}$$

On the other hand, resolving node a first by applying $(Y/\Delta)_{m+1}$ results in a mesh network with m external nodes, each one of which are connected to the internal node b, see the figure below. In the resulting circuit, each external node i is connected to node b via a resistor with conductance Y_{ix}^a, see equation 4.1-(a). Every pair of external nodes i and j are connected by a resistor with conductance Y_{ij}^a, see 4.1-(b).



$$(a) \quad Y_{ij}^a = \frac{y_i y_j}{\sum_k^m y_k + x} \qquad (b) \quad Y_{ix}^a = \frac{y_i x}{\sum_k^m y_k + x} \tag{4.1}$$

Now, applying $(Y/\Delta)_m$ to resolve node b in the resulting circuit, leads to parallel conductances Y_{ij}^a and Y_{ij}^{ab} between any two nodes $1 \le i, j \le m$, see the diagram below.



The value of Y_{ij}^{ab} is computed as follows:

$$Y_{ij}^{ab} = \frac{Y_{ix}^a Y_{jx}^a}{\sum_k^m Y_{kx}^a} = \frac{\frac{y_i x y_j x}{(\sum_k^m y_k + x)^2}}{\frac{\sum_k^m y_k x}{(\sum_k^m y_k + x)}} = \frac{\frac{y_i y_j x}{(\sum_k^m y_k + x)}}{(\sum_k^m y_k)} = \frac{y_i y_j x}{(\sum_k^m y_k)(\sum_k^m y_k + x)}$$

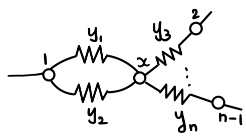
Combining the parallel edges,

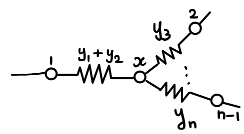
$$\begin{aligned} Y_{ij}^a + Y_{ij}^{ab} &= \frac{y_i y_j}{\sum_k^m y_k + x} + \frac{y_i y_j x}{(\sum_k^m y_k)(\sum_k^m y_k + x)} = \frac{y_i y_j (\sum_k^m y_k) + y_i y_j x}{(\sum_k^m y_k)(\sum_k^m y_k + x)} \\ &= \frac{y_i y_j (\sum_k^m y_k + x)}{(\sum_k^m y_k)(\sum_k^m y_k + x)} = \frac{y_i y_j}{(\sum_k^m y_k)} = Y_{ij}^{ba} \end{aligned}$$

Thus the two orders of rewriting the circuit in diagram (a) produce equivalent results. □

Lemma 4.2. *The parallel and star-mesh critical pairs in* Resist_R *resolve.*

Proof. Consider the star network composed with two of its edges connected in parallel, see circuit (a).

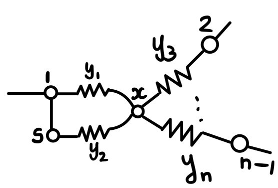
(a) 

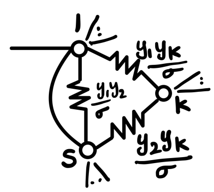
(b) 

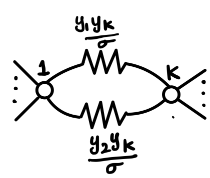
To rewrite circuit (a), one can first apply [Parallel] rewrite rule resulting in circuit (b). Applying $(\mathrm{Y}/\Delta)_{n-1}$ on circuit (b) results in a mesh with $n-1$ nodes with conductance value between any two nodes as follows: For all $2 \leq j,k \leq n-1$,

$$(a)\;\; Y_{1k} = \frac{(y_1+y_2)y_k}{\sum_i^n y_i} \qquad (b)\;\; Y_{jk} = \frac{y_j y_k}{\sum_i^n y_i}$$

Alternatively, to rewrite circuit (a), one can apply the spider [Expansion] rule resulting in circuit (c) shown below. Applying, $(\mathrm{Y}/\Delta)_n$ on circuit (c), results in n-node mesh with a parallel edge of infinite conductance between nodes 1 and s as shown in circuit (d). Applying [Short circuit] rule on circuit (d) results in parallel edge between node 1, and any other node k where $2 \leq k \leq$, see circuit (f). In the pictures below we use σ to denote the sum of the conductances, $\sigma = \sum_{i=1}^n y_i$

(c) 

(d) 

(f) 

Using [Parallel] rule in the mesh, for all $2 \leq k \leq n-1$,

$$Y_{1k} = \frac{(y_1+y_2)y_k}{\sum_i^n y_i}$$

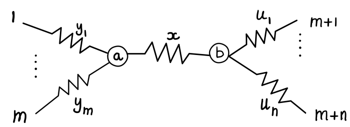
Moreover, by $(\mathrm{Y}/\Delta)_n$ rule, for all $2 \leq j,k \leq n-1$,

$$Y_{jk} = \frac{y_j y_k}{\sum_i^n y_i}$$

□

Lemma 4.3. *The star-mesh critical pairs in* Resist_R *can be resolved.*

Proof. Consider the circuit shown below. We must show that eliminating node a first by applying $(Y/\Delta)_{m+1}$ and then node b by applying $(Y/\Delta)_{n+m}$ yields the same result as eliminating node b first by applying $(Y/\Delta)_{n+1}$ and then node a by applying $(Y/\Delta)_{n+m}$.



Note that, in order to rewrite a star network all its arms must have non-zero conductances (these would result in an open circuit) nor can there be any infinite conductances (these would be a bare wire).

We first develop a general algorithm for naming the new edges resulting from each rewrite of the circuit. The algorithm is as follows:

Algorithm for labelling edges:

1. At each step, the new edges always carry the names of the eliminated nodes as superscript in the order of their elimination.
2. The subscripts (in general) refer to the index of resistors being combined – Y^a_{ij} is given by combining y_i and y_j by eliminating a with $i \leq j$; combining the edges Y^a_{ix} and Y^a_{jx} by eliminating b gives Y^{ab}_{ij}; Y^a_{ix} is given by combining the edge y_i with x eliminating node a.
3. Whenever only 'y_i' resistors are combined create to a new edge, the new edge carries label 'Y' with appropriate super and subscripts; label 'φ' means a y and a u resistor are combined; label 'U' means only u resistors have been combined.

To make the naming procedure clear, we consider a simple case shown in Figure 2:

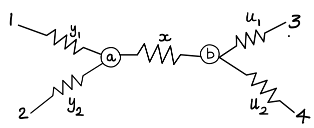
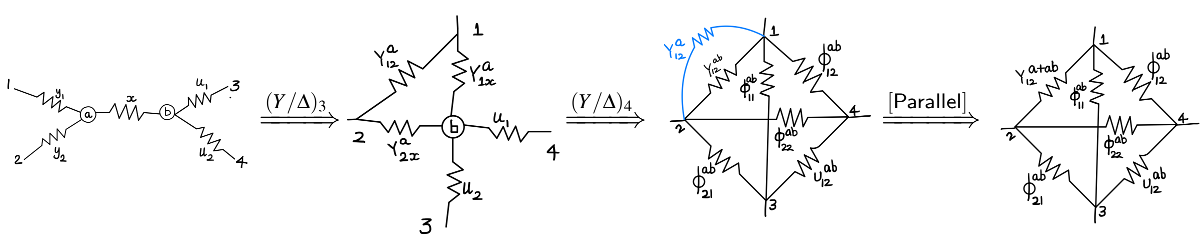
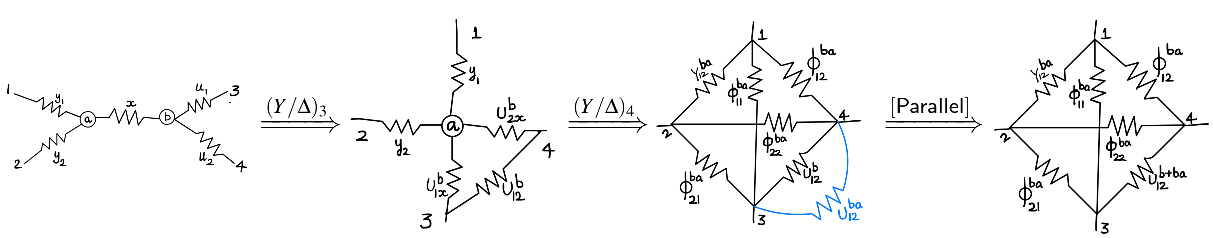


Figure 2: Node a and node b connected by resistor x

Two possible ways of reducing the circuit in Figure 2 are shown below:
Eliminate internal node a followed by internal node b:



Eliminate internal node b followed by internal node a:



The two ways of reducing the circuit are equal if:

$$\text{for } 1 \le i,j \le 2, \quad \varphi_{ij}^{ab} = \varphi_{ij}^{ba}$$
$$\text{for } 1 \le i,j \le 2, \quad U_{ij}^{ab} = U_{ij}^{b+ba}$$
$$\text{for } 1 \le i,j \le 2, \quad Y_{ij}^{a+ab} = Y_{ij}^{ba}$$

Following this approach, the two ways of reducing the general circuit (given in the beginning of this proof) are equal if:

$$\text{for } 1 \le i \le m, \quad 1 \le j \le n \quad \varphi_{ij}^{ab} = \varphi_{ij}^{ba} \tag{4.2}$$
$$\text{for } 1 \le i,j \le n \quad U_{ij}^{ab} = U_{ij}^{b+ba} \tag{4.3}$$
$$\text{for } 1 \le i,j \le m \quad Y_{ij}^{a+ab} = Y_{ij}^{ba} \tag{4.4}$$

where,

$$\text{for } 1 \le i,j \le m, \quad Y_{ij}^{a+ab} = Y_{ij}^{a} + Y_{ij}^{ab} \qquad \text{for } 1 \le i,j \le n, \quad U_{ij}^{ba+a} = U_{ij}^{b} + U_{ij}^{ba}$$

$$\text{for } 1 \le i \le m \text{ and } 1 \le j \le n, \quad \varphi_{ij}^{ab} = \frac{Y_{ix}^{a} u_j}{\sum_k^n u_k + \sum_k^m Y_{kx}^{a}}$$
$$\text{for } 1 \le i \le m \text{ and } 1 \le j \le n, \quad \varphi_{ij}^{ba} = \frac{y_i^b U_{jx}^{b}}{\sum_k^m y_k + \sum_k^n U_{kx}^{b}}$$

$$\text{for } 1 \le i,j \le m, \quad Y_{ij}^{ab} = \frac{Y_{ix}^{a} Y_{jx}^{a}}{\sum_k^n u_k + \sum_k^m Y_{kx}^{a}} \qquad \text{for } 1 \le i,j \le n, \quad U_{ij}^{ba} = \frac{U_{ix}^{b} U_{jx}^{b}}{\sum_k^m y_k + \sum_k^n U_{kx}^{b}}$$
$$\text{for } 1 \le i,j \le n, \quad U_{ij}^{ab} = \frac{u_i u_j}{\sum_k^n u_k + \sum_k^m Y_{kx}^{a}} \qquad \text{for } 1 \le i,j \le m, \quad Y_{ij}^{ba} = \frac{y_i^a y_j^a}{\sum_k^m y_k + \sum_k^n U_{kx}^{b}}$$
$$\text{for } 1 \le i,j \le m, \quad Y_{ij}^{a} = \frac{y_i y_j}{\sum_k^m y_k + x} \qquad \text{for } 1 \le i,j \le n, \quad U_{ij}^{b} = \frac{u_i u_j}{\sum_k^n u_k + x}$$
$$\text{for } 1 \le i \le m, \quad Y_{ix}^{a} = \frac{y_i x}{\sum_k^m y_k + x} \qquad \text{for } 1 \le i \le n, \quad U_{ix}^{b} = \frac{u_i x}{\sum_k^n u_k + x}$$

Now, proving 4.2, $\varphi_{ij}^{ab} = \varphi_{ij}^{ba}$:

$$\begin{aligned}
\varphi_{ij}^{ab} &= \frac{\frac{y_i x}{\sum_k^m y_k + x} u_j}{\sum_k^n u_k + \sum_k^m \frac{y_k x}{y_k + x}} = \frac{y_i x u_j}{(\sum_k^n u_k)(\sum_k^m y_k + x) + \sum_k^m y_k x} \\
&= \frac{y_i x u_j}{(\sum_k^m y_k)(\sum_k^n u_k) + \sum_k^n u_k x + \sum_k^m y_k x} = \frac{y_i x u_j}{\sum_k^n u_k x + \sum_k^m y_k(\sum_k^n u_k + x)} \\
&= \frac{\frac{y_i x u_j}{\sum_k^n u_k + x}}{\frac{\sum_k^n u_k x}{\sum_k^n u_k + x} + \sum_k^m y_k} = \frac{y_i U_{jx}^{b}}{\sum_k^n U_{kx}^{b} + \sum_k^m y_k} = \varphi_{ij}^{ba}
\end{aligned}$$

Now, proving 4.3, $U_{ij}^{ab} = U_{ij}^{b+ba}$:

$$U_{ij}^{ab} = \frac{u_i u_j}{\sum_k^n u_k + \sum_k^m Y_{kx}^a} = \frac{u_i u_j}{\sum_k^n u_k + \frac{\sum_k^m Y_k x}{\sum_k^m y_k + x}} = \frac{u_i u_j}{\sum_k^n u_k(\sum_k^m y_k + x) + \sum_k^m Y_k x}$$
$$\overset{(*)}{=} \frac{u_i u_j}{\sum_k^m y_k(\sum_k^n u_k + x) + \sum_k^n u_k x} = \frac{\frac{u_i u_j}{\sum_k^n u_k + x}(\sum_k^m y_k + x)}{\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x}} \tag{4.5}$$

For step $(*)$, see computation of denominator in the proof of 4.2, $\varphi_{ij}^{ab} = \varphi_{ij}^{ba}$.

$$U_{ij}^{b+ba} = U_{ij}^{b} + U_{ij}^{ba} = \frac{u_i u_j}{\sum_k^n u_k + x} + \frac{\frac{u_i u_j x^2}{(\sum_k^n u_k + x)^2}}{\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x}} = \frac{u_i u_j \left(\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x}\right) + \left(\frac{u_i u_j x^2}{\sum_k^n u_k + x}\right)}{(\sum_k^n u_k + x)\left(\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^m y_k + x}\right)}$$
$$= \frac{u_i u_j \left(\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x} + \frac{x^2}{\sum_k^n u_k + x}\right)}{(\sum_k^n u_k + x)\left(\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x}\right)} = \frac{\frac{u_i u_j}{\sum_k^n u_k + x}\left(\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x} + \frac{x^2}{\sum_k^n u_k + x}\right)}{\left(\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x}\right)} \tag{4.6}$$

The denominators of equations 4.5 and 4.6 are the same. Hence, multiplying the numerators of these equations by $\frac{\sum_k^n u_k + x}{u_i u_j}$, it suffices to prove that:

$$\sum_k^m y_k + \frac{\sum_k^n u_k x}{\sum_k^n u_k + x} + \frac{x^2}{\sum_k^n u_k + x} = \sum_k^m y_k + x\left(\frac{\sum_k^n u_k + x}{\sum_k^n u_k + x}\right) = \sum_k^m y_k + x \tag{4.7}$$

The proof for equation 4.4 is analogous.

□

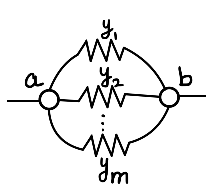
The following is the main result of this paper:

Theorem 4.4. Resist_R *has a confluent terminating rewriting system on maps.*

Proof. First observe that the reduction using rewriting rules of Resist_R must terminate. This may be observed by keeping track of the number of nodes N (after expansion) and the number of parallel arrows P and using lexicographical ordering on the pairs (N,P). The [Spider] and [Parallel] rules reduce both N and P. The star-mesh family of identities reduce the number of nodes N as an internal node is always removed – note that the star-mesh rule, on the other hand, can increase the number of parallel connections. As the lexicographical ordering on $\mathbb{N} \times \mathbb{N}$ is a well-ordering, this shows that the reduction process must eventually terminate.

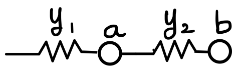
Since the rewriting terminates, local confluence, that is *resolutions* of diverging single step rewrites, implies global confluence. Hence, it suffices to prove the local confluence property for the distinct pairs of terms produced by overlapping divergent one step rewrites: these are often called critical pairs. This amounts to proving that the order of reducing overlapping rewrites does not matter. The critical pairs (excluding the spider rewrites) that occur in the this rewriting system are drawn below:

(a) **Overlapping [Parallel] rewrites:**

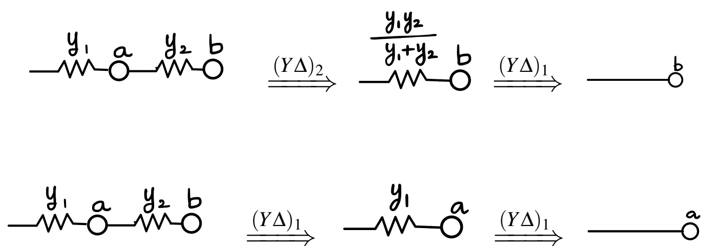


Rewriting the network on the left requires $m-1$ overlapping applications of the [Parallel] rule. Since, combining two parallel resistors involves adding their conductances, and addition is associative, all the orders of application of the [Parallel] rule yield the same final circuit. Hence, local confluence holds in this case.

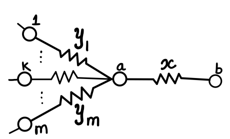
(b) **Overlapping $(Y/\Delta)_1$ and $(Y/\Delta)_2$ rewrite:**



To rewrite the network on the left, one may apply $(Y/\Delta)_1$ to eliminate node b first, or apply $(Y/\Delta)_2$ to eliminate node a first. This results in a critical pair $((Y/\Delta)_1,(Y/\Delta)_2)$ in the rewriting process of such networks. However, the critical pair is locally confluent, see below:

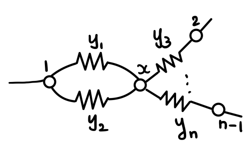


(c) **Overlapping $(Y/\Delta)_1$ and star-mesh rewrite:**



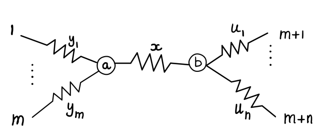
To rewrite the network on the left, one may apply $(Y/\Delta)_1$ to eliminate node b first (and then apply $(Y/\Delta)_m$ to eliminate a), or apply $(Y/\Delta)_{m+1}$ to eliminate node a first (and then apply $(Y/\Delta)_m$ to eliminate b). This results in a critical pair $((Y/\Delta)_1,(Y/\Delta)_m)$ in the rewriting of such networks. However, by Lemma 4.1, local confluence holds for this critical pair.

(d) **Overlapping** [Parallel] **and star-mesh rewrites:**



To rewrite the network on the left, [Parallel] to combine resistors y_1 and y_2 first (followed by $(Y/\Delta)_{n-1}$ to eliminate node x), or apply $(Y/\Delta)_n$ may be applied to eliminate node x first. This results in a critical pair $([\text{Parallel}],(Y/\Delta)_n)$ in the rewriting process of such networks. However, by Lemma 4.2, local confluence holds for this critical pair.

(e) **Overlapping two star-mesh rewrites:**



To rewrite the network on the left, $(Y/\Delta)_{n+1}$ may be applied first to eliminate node b first (followed by $(Y/\Delta)_{m+n}$ to eliminate node a), or $(Y/\Delta)_{m+1}$ may be applied to eliminate node a first (followed by $(Y/\Delta)_{m+n}$ to eliminate node b). This results in a critical pair $((Y/\Delta)_{n+1},(Y/\Delta)_{m+1})$ in the rewriting process of such networks. However, by Lemma 4.3, local confluence holds for this critical pair.

□

An immediate consequence is:

Corollary 4.5. Modulo the decidability of the positive division rig R, Resist_R has a decidable equality by reduction to normal form.

5 Discussion

In this paper we have provided a normal form for resistor networks over a positive division rig and, thereby, a decision procedure for equality of resistor circuits (given decidability of equality for the rig). Even though modest, as far as we know, ours is the first such result in the literature.

Of course, our motivation came from the difficulty of working with the existing ‘normal forms’ for stabilizer circuits [1]. As pointed out by Kissinger in [13], these normal forms for stabilizer circuits are hard to work with and *“almost a decade after completeness was proven for the stabiliser fragment of the ZX calculus, new ideas are still needed”*. Based on the recently established connection between quantum and electrical circuits [6, 5], our thought was that, studying simpler cases such as resistor circuits might provide new insights into normal forms for stabilizer circuits.

An interesting and a more challenging question is whether the results in this paper can be generalized to arbitrary division rigs so as to cover the ‘resistor’ case arising from qudit stabilizer quantum mechanics. As has been mentioned, to apply these ideas to stabilizer circuits a necessary step is to generalize these results to division rigs and so, in particular, to finite fields. The technical difficulty of applying these ideas verbatim is the question of how one handles zeros and divisions by zero.

An example of this difficulty over a finite field, arises when resolving the critical pair $(Y/\Delta)_1$ with $(Y/\Delta)_n$ (for $n \geq 3$) (essentially Lemma 4.1 above). The rewriting of the n-star to an n-mesh can involve a division by zero (when $\sum_{i=1}^{n} y_i = 0$): it is tempting to think that this should be interpreted as giving “infinite conductances” in the mesh. However, removing a point of the star using $(Y/\Delta)_1$ will also remove the division by zero in the subsequent $(Y/\Delta)_{n-1}$ rewriting showing such an interpretation is not valid.

The point is that for finite fields and division rings the star/mesh transformations are only valid when the sum of the conductances of the star is non-zero. This, of course complicates the rewriting story and also reopens the question what a convenient presentation of resistor circuits over finite fields might be! We leave resolving these issues for future work.

A surprisingly basic – and as far as we know open – question which arises from this work concerns whether there is a finite presentation of Resist_R in terms of generators and relation. In order to provide a presentation for Resist_R, we assume an infinite family of star-mesh identities, $(Y/\Delta)_n$, for each $n \in \mathbb{N}$. While we have shown that this infinite set of identities completely characterize equality between circuits, it is an open question whether there is a finite presentation of the category. We conjecture that there isn’t one.

6 Acknowledgements

ARK would like to thank NTT Research for financial and technical support. Research at IQC is supported in part by the Government of Canada through Innovation, Science and Economic Development Canada (ISED).

References

[1] Miriam Backens (2016): *Completeness and the ZX-calculus*. Ph.D. thesis, Oxford University, doi:10.48550/arXiv.1602.08954. ArXiv preprint arXiv:1602.08954.

[2] John Baez & Brendan Fong (2015): *Circuits, Categories and Rewrite rules*. Available at `https://math.ucr.edu/home/baez/networks_warsaw/circuits_web_warsaw.pdf`. Higher-Dimensional Rewriting and Applications, Warsaw, June 2015.

[3] John C Baez & Brendan Fong (2015): *A compositional framework for passive linear networks*. arXiv preprint arXiv:1504.05625, doi:10.48550/arXiv.1504.05625.

[4] 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), pp. 1–12, doi:10.1109/LICS.2019.8785877.

[5] Robin Cockett, Amolak Ratan Kalra & Shiroman Prakash (2022): *Categories of Kirchhoff relations*. arXiv preprint arXiv:2205.05870, doi:10.48550/arXiv.2205.05870.

[6] Cole Comfort & Aleks Kissinger (2021): *A graphical calculus for Lagrangian relations*. arXiv preprint arXiv:2105.06244, doi:10.4204/EPTCS.372.24.

[7] Brandon Coya (2018): *Circuits, Bond Graphs, and Signal-Flow Diagrams: A Categorical Perspective*, doi:10.48550/ARXIV.1805.08290. Available at `https://arxiv.org/abs/1805.08290`.

[8] Brendan Fong (2016): *The algebra of open and interconnected systems*. arXiv preprint arXiv:1609.05382, doi:10.48550/arXiv.1609.05382.

[9] Brendan Fong & David I Spivak (2019): *Hypergraph categories*. Journal of Pure and Applied Algebra 223(11), pp. 4746–4777, doi:10.48550/arXiv.1806.08304.

[10] D. J. Frank & C. J. Lobb (1988): *Highly efficient algorithm for percolative transport studies in two dimensions*. Phys. Rev. B 37, pp. 302–307, doi:10.1103/PhysRevB.37.302. Available at `https://link.aps.org/doi/10.1103/PhysRevB.37.302`.

[11] Ghassan George Batrouni & Alex Hansen (1988): *Fourier acceleration of iterative processes in disordered systems*. Journal of statistical physics 52, pp. 747–773, doi:10.1007/BF01019728.

[12] Amolak Ratan Kalra (2022): *The Category of Kirchhoff Relations*. Master's thesis, University of Calgary, doi:10.11575/PRISM/39953.

[13] Aleks Kissinger (2022): *Phase-free ZX diagrams are CSS codes (... or how to graphically grok the surface code)*. arXiv preprint arXiv:2204.14038, doi:10.48550/arXiv.2204.14038.

[14] Henning Arendt Knudsen & Sándor Fazekas (2006): *Robust algorithm for random resistor networks using hierarchical domain structure*. Journal of computational physics 211(2), pp. 700–718, doi:10.1016/j.jcp.2005.06.007.

[15] Joost Rommes & Wil HA Schilders (2009): *Efficient methods for large resistor networks*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 29(1), pp. 28–39, doi:10.1109/TCAD.2009.2034402.

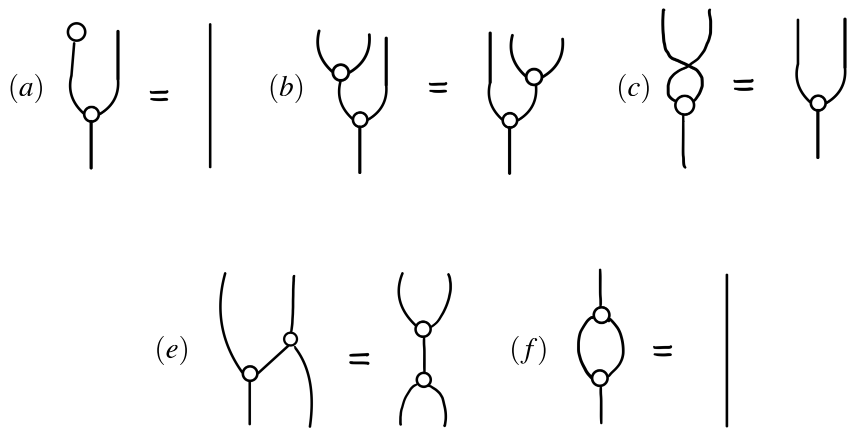
[16] D. W. C. Shew (1947): *XXVII. Generalized star and mesh transformations*. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 38(279), pp. 267–275, doi:10.1080/14786444708521594.

A Hypergraph categories

A category is a **hypergraph category** in case it is a symmetric monoidal category in which every object is coherently a special commutative Fröbenius algebra, [9]. This means that each object X in the category has an associated special Fröbenius algebra structure $(X, \nabla_X : X \otimes X \longrightarrow X, \eta_X : I \longrightarrow X, \Delta_X : X \longrightarrow X \otimes X, \varepsilon_X : X \longrightarrow I)$, whose identities are graphically depicted below (with $\circ$ indicating both the multiplication, comultiplication, and units):

The comultiplication, the counit, the multiplication and the unit are drawn as follows:

The maps satisfy the following equations and their vertically flipped image:



(A.1)

The Frobenius structure is not natural but is "coherent" in the sense that the multiplication on the tensor of two objects is given by $\Delta_{X\otimes Y} := (X\otimes Y)\otimes(X\otimes Y) \xrightarrow[\simeq]{\mathsf{ex}} (X\otimes X)\otimes(Y\otimes Y) \xrightarrow{\Delta_X\otimes\Delta_Y} X\otimes Y$ and the comultiplication and units are similarly given.

Hypergraph categories are automatically compact closed: each object is self-dual. This has the effect that the directionality of inputs and outputs is not as important as the connectivity.

Monoidal Structures on Generalized Polynomial Categories

Joseph Dorta
Louisiana State University
Baton Rouge, LA, USA
jdorta1@lsu.edu

Samantha Jarvis
CUNY Graduate Center
New York, NY, USA
sjarvis@gradcenter.cuny.edu

Nelson Niu
University of Washington
Seattle, WA, USA
nsniu@uw.edu

Recently, there has been renewed interest in the theory and applications of de Paiva's dialectica categories and their relationship to the category of polynomial functors. Both fall under the theory of generalized polynomial categories, which are free coproduct completions of free product completions of (monoidal) categories. Here we extend known monoidal structures on polynomial functors and dialectica categories to generalized polynomial categories. We highlight one such monoidal structure, an asymmetric operation generalizing composition of polynomial functors, and show that comonoids with respect to this structure correspond to categories enriched over a related free coproduct completion. Applications include modeling compositional bounds on dynamical systems.

1 Introduction

Categories whose morphisms (often referred to as *lenses*) model bidirectional data flows are ubiquitous in applied category theory, with applications to such diverse fields as logic [21, 27], database management [3, 11, 14], game theory [4, 5, 13], dynamical and distributed systems [16, 19, 23–25], and machine learning [7, 9, 10]. Moss observed that we can obtain a general class of such categories via free product and coproduct completions, universal constructions with convenient concrete characterizations [18]. That is, starting from a category $\mathbb{C}$, we can form a category $\Sigma\Pi\mathbb{C}$ whose objects are formal coproducts of products of objects in $\mathbb{C}$, or *polynomials* in $\mathbb{C}$ for short; then the morphisms between these coproducts of products naturally have both a forward component and a backward component in addition to subsuming the original morphisms from $\mathbb{C}$. Examples of such generalized polynomial categories include the category **Poly** of polynomial functors, which may be used to model interaction protocols [20]; and a category whose *homogeneous* polynomials span a full subcategory equivalent to de Paiva's dialectica category on sets, a model for intuitionistic linear logic [21]. We review the construction of $\Sigma\Pi\mathbb{C}$ and exhibit these examples in Section 2.

The utility of these examples lies not only in their bidirectional morphisms but also in the assorted ways in which such morphisms can be combined via monoidal products. There are several ways to lift a monoidal structure on $\mathbb{C}$ to a monoidal structure on $\Sigma\Pi\mathbb{C}$. We present two such ways in Section 3—one classical, given by an iterated Day convolution [8]; and one we believe is new in the literature, generalizing functor composition in **Poly**.

Many applications of polynomial functors (such as those in [20]) depend on a remarkable result by Ahman and Uustalu [2, 3]: the category of comonoids in **Poly** with respect to the composition product is equivalent to the category whose objects are small categories and whose morphisms are *cofunctors*, as introduced by Aguiar [1]. Our main result, Theorem 4.3, is that Ahman and Uustalu's statement naturally generalizes to $\Sigma\Pi\mathbb{C}$. By replacing **Poly** with $\Sigma\Pi\mathbb{C}$ equipped with our generalized composition product, comonoids become small *enriched* categories whose base of enrichment is $\Sigma\mathbb{C}^{\mathsf{op}}$ with Day convolution. Then morphisms of these comonoids generalize cofunctors to the enriched setting in a way

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 84–97, doi:10.4204/EPTCS.397.6

that coincides with Clarke and Di Meglio's recent definition of *enriched cofunctors* [6]. We review the necessary definitions before presenting this correspondence in Section 4 via an explicit construction.

In Section 5, we take $\mathbb{C}$ to be the extended nonnegative reals to demonstrate how morphisms in $\Sigma\Pi\mathbb{C}$ may be used to model dynamical systems with boundedness conditions preserved by the generalized composition product. Such morphisms can be lifted to enriched cofunctors via a right adjoint to the forgetful functor from comonoids to their underlying objects; we review a few examples before stating the general result as Theorem 5.6. Finally, we suggest directions for future work in Section 6.

Acknowledgments

The authors are indebted to the mentorship of Valeria de Paiva at the 2022 AMS MRC and to insight and feedback from our fellow mentees: Charlotte Aten, Colin Bloomfield, Eric Bond, Matteo Capucci, Bruno Gavranović, Jérémie Koenig, Abdullah Malik, Francisco Rios, Jan Rooduijn, and Jonathan Weinberger. Additionally, the authors are grateful for the comments provided by the anonymous reviewers.

This material is based upon work supported by the National Science Foundation under Grant Number DMS 1641020. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

2 Free (co)product completions and polynomial categories

We begin by recalling two constructions on a category $\mathbb{C}$: the free product completion and its dual, the free coproduct completion. Here we follow Moss [18]; we omit proofs for standard results.

Definition 2.1. The ***free product completion*** of a category $\mathbb{C}$ is the category $\Pi\mathbb{C}$, where

- an object, denoted $\prod_{i\in I} c_i$, consists of
  - a set I;
  - for each $i \in I$, an object c_i in $\mathbb{C}$;
- a morphism $\varphi\colon \prod_{i\in I} c_i \to \prod_{j\in J} d_j$ consists of
  - a function $\varphi^\sharp\colon J \to I$;
  - for each $j \in J$, a morphism $\varphi_j\colon c_{\varphi^\sharp j} \to d_j$ in $\mathbb{C}$. ◊

The category $\mathbb{C}$ embeds into $\Pi\mathbb{C}$ as a full subcategory via $c \mapsto \Pi_{*\in 1} c$, where $1 := \{*\}$ is the singleton set. As implied by the name "free product completion," the category $\Pi\mathbb{C}$ equipped with the embedding $\mathbb{C} \hookrightarrow \Pi\mathbb{C}$ is universal among categories $\mathbb{D}$ with small products equipped with functors $\mathbb{C} \to \mathbb{D}$.

We may alternatively characterize $\Pi\mathbb{C}$ as follows, using the fact that $[\mathbb{C},\mathbf{Set}]^{\mathsf{op}}$ equipped with the Yoneda embedding $\mathbb{C} \hookrightarrow [\mathbb{C},\mathbf{Set}]^{\mathsf{op}}$ is the free limit completion of $\mathbb{C}$ and restricting to products.

Proposition 2.2. *The category $\Pi\mathbb{C}$ is equivalent to the full subcategory of $[\mathbb{C},\mathbf{Set}]^{\mathsf{op}}$ spanned by products of representable functors.*

Definition 2.3. The ***free coproduct completion*** of a category $\mathbb{C}$ is the category $\Sigma\mathbb{C}$, where

- an object, denoted $\sum_{i\in I} c_i$, consists of
  - a set I;
  - for each $i \in I$, an object c_i in $\mathbb{C}$;
- a morphism $\varphi\colon \sum_{i\in I} c_i \to \sum_{j\in J} d_j$ consists of
  - a function $\varphi\colon I \to J$;
  - for each $i \in I$, a morphism $\varphi_i\colon c_i \to d_{\varphi i}$ in $\mathbb{C}$. ◊

There is a fully faithful functor $\mathbb{C} \hookrightarrow \Sigma\mathbb{C}$ sending $c \mapsto \sum_{*\in 1} c$. Comparing the definitions, we find that $(\Sigma\mathbb{C}^{\mathsf{op}})^{\mathsf{op}} \approx \Pi\mathbb{C}$; in particular, dualizing Proposition 2.2 yields the following.

Proposition 2.4. *The category $\Sigma\mathbb{C}$ is equivalent to the full subcategory of $[\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$ spanned by coproducts of representable functors.*

The category $\Sigma\mathbb{C}$ equipped with the embedding $\mathbb{C} \hookrightarrow \Sigma\mathbb{C}$ is universal among categories $\mathbb{D}$ with small coproducts equipped with functors $\mathbb{C} \to \mathbb{D}$; in particular $\Sigma\mathbb{C}$ has small coproducts. As for products in $\Sigma\mathbb{C}$, we have the following proposition.

Proposition 2.5. *If $\mathbb{C}$ has all small products, then $\Sigma\mathbb{C}$ has all small products given by a distributive law*

$$\prod_{i\in I}\sum_{j\in J_i} c_{i,j} \cong \sum_{\bar{j}\in\prod\limits_{i\in I} J_i}\prod_{i\in I} c_{i,\bar{j}_i}. \tag{1}$$

Proof. Eq. (1) holds in **Set**, so since (co)products are computed pointwise in $[\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$, it holds there as well. When every $c_{i,j}$ is representable, the right hand side is a coproduct of representables, as products of representables are themselves representable. Hence Eq. (1) also holds in the full subcategory of $[\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$ spanned by coproducts of representables. Then the conclusion follows from Proposition 2.4. □

If we freely add products, then freely add coproducts, we obtain the central construction of this paper.

Definition 2.6. The category $\Sigma\Pi\mathbb{C}$ of ***polynomials in*** $\mathbb{C}$ is the category where

- an object, denoted $\sum_{i\in I}\prod_{a\in A_i} c_{i,a}$, consists of
  - a set I of **positions**;
  - for each $i \in I$, a set A_i of **directions** at i;
  - a doubly-indexed family $(c_{i,a})_{i\in I, a\in A}$ of objects of $\mathbb{C}$, called **predicates**;
- a morphism $\varphi\colon \sum_{i\in I}\prod_{a\in A_i} c_{i,a} \to \sum_{j\in J}\prod_{b\in B_j} d_{j,b}$ consists of
  - an **on-positions function** $\varphi\colon I \to J$;
  - for each $i \in I$, an **on-directions function** $\varphi_i^\sharp\colon B_{\varphi i} \to A_i$;
  - for each $i \in I$ and $b \in B_{\varphi i}$, an **on-predicates map** $\varphi_{i,b}\colon c_{i,\varphi_i^\sharp b} \to d_{\varphi i, b}$. ◊

Unraveling the definitions, we see that $\Sigma\Pi\mathbb{C}$ is indeed the free coproduct completion of the free product completion of $\mathbb{C}$. The following characterization of the hom-sets of $\Sigma\Pi\mathbb{C}$ is immediate.

Proposition 2.7. *The hom-sets of $\Sigma\Pi\mathbb{C}$ are given by*

$$\Sigma\Pi\mathbb{C}\Big(\sum_{i\in I}\prod_{a\in A_i} c_{i,a}, \sum_{j\in J}\prod_{b\in B_j} d_{j,b}\Big) \cong \prod_{i\in I}\sum_{j\in J}\prod_{b\in B_j}\sum_{a\in A_i}\mathbb{C}(c_{i,a}, d_{j,b}).$$

Even though we added products before we added coproducts, Proposition 2.5 ensures that $\Sigma\Pi\mathbb{C}$ has small products in addition to having small coproducts and that these products distribute over coproducts.

The construction of $\Sigma\Pi\mathbb{C}$ generalizes two particularly versatile categories: the category of *polynomial functors* and one of de Paiva's *dialectica categories* [21]. In the remainder of this section, we review each of these categories in turn, observing how they arise from categories of polynomials.

The category of polynomial functors

We consider $\Sigma\Pi\mathbb{C}$ a *generalized polynomial category* because it generalizes the category **Poly** of polynomial functors, which we recall below.

Definition 2.8. A ***polynomial functor*** $p\colon \mathbf{Set} \to \mathbf{Set}$ is a coproduct of representable functors. That is, there exist $I \in \mathbf{Set}$ and $p[i] \in \mathbf{Set}$ for each $i \in I$ such that, for $y^{p[i]} := \mathbf{Set}(p[i], -)$,

$$p \cong \sum_{i \in I} y^{p[i]}.$$

We call the elements of $p(1) \cong I$ the **positions** of p and the elements of $p[i]$ the **directions** of p at i.[1] We denote the category of polynomial functors and the natural transformations between them by **Poly**. $\Diamond$

It turns out that **Poly** is the category of polynomials in the terminal category $\mathbb{1}$, consisting of one object and no non-identity morphisms.

Proposition 2.9. $\mathbf{Poly} \approx \Sigma\Pi\mathbb{1}$.

Proof. By definition, $\Pi\mathbb{1} \approx \mathbf{Set}^{\mathsf{op}}$. Then Proposition 2.4 implies that $\Sigma\Pi\mathbb{1}$ is the full subcategory of $[\mathbf{Set}, \mathbf{Set}]$ spanned by coproducts of representables. $\square$

Viewing **Poly** as $\Sigma\Pi\mathbb{1}$, we can characterize the morphisms of **Poly** as follows.

Example 2.10. A morphism $\varphi\colon p \to q$ in $\mathbf{Poly} \approx \Sigma\Pi\mathbb{1}$ consists of
- an **on-positions function** $\varphi_1\colon p(1) \to q(1)$;[2]
- for each $i \in p(1)$, an ***on-directions function*** $\varphi_i^\sharp\colon q[\varphi i] \to p[i]$. $\Diamond$

The dialectica category on sets

Rather than working with the entire category of polynomials in $\mathbb{C}$, it is sometimes easier to work with one of its full subcategories, which we define below.

Definition 2.11. A polynomial in $\mathbb{C}$ is ***homogeneous***[3] if it can be written in the form

$$\sum_{i \in I} \prod_{a \in A} u_{i,a},$$

where the set A does not depend on $i \in I$. We let $\mathbf{Hmg}(\mathbb{C})$ denote the full subcategory of $\Sigma\Pi\mathbb{C}$ spanned by homogeneous polynomials. $\Diamond$

As an example, let $\mathbb{2}$ denote the walking arrow category, which has two objects $\bot$ and $\top$ and one non-identity arrow $\bot \to \top$.

Example 2.12. In the category $\mathbf{Hmg}(\mathbb{2})$,
- an object, denoted $\sum_{i \in I} \prod_{a \in A} c_{i,a}$, consists of
  - two sets, I and A;
  - for each $(i,a) \in I \times A$, an object $c_{i,a} \in \{\bot, \top\}$;
- a morphism $\varphi\colon \sum_{i \in I} \prod_{a \in A} c_{i,a} \to \sum_{j \in J} \prod_{b \in B} d_{j,b}$ consists of
  - a function $\varphi\colon I \to J$;
  - a function $\varphi^\sharp\colon I \times B \to A$; such that
  - for each $i \in I$ and $b \in B$, if $c_{i,\varphi^\sharp(i,b)} = \top$, then $d_{\varphi i,b} = \top$. $\Diamond$

This is precisely de Paiva's original dialectica category on **Set** [21].

Proposition 2.13. $\mathbf{Hmg}(\mathbb{2}) \approx \mathbf{Dial}(\mathbf{Set})$.

[1] The "positions" and "directions" terminology for polynomial functors was introduced by Spivak [25].
[2] We use a subscript 1 for the on-positions function as it is the 1-component of φ as a natural transformation [20].
[3] The terminology comes from algebra, where a *homogeneous* polynomial is one whose summands all have the same degree.

3 Monoidal structures on polynomial categories

Most of the applications of **Poly** and **Dial**(**Set**) rely on their monoidal structures; in this section, we will generalize such structures to $\Sigma\Pi\mathbb{C}$. Throughout, let $(\mathbb{C}, e, \cdot)$ be a monoidal category with unit $e \in \mathbb{C}$ and product $\cdot\colon \mathbb{C} \times \mathbb{C} \to \mathbb{C}$. The monoidal structure on $\mathbb{C}$ then induces monoidal structures on $\Sigma\Pi\mathbb{C}$.

3.1 The parallel product

A monoidal product $\cdot$ on $\mathbb{C}$ always induces a monoidal product $\odot$ on the free colimit completion $[\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$ of $\mathbb{C}$: the Day convolution [8], which agrees with $\cdot$ on the full subcategory $\mathbb{C} \hookrightarrow [\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$.

Proposition 3.1. *The Day convolution $\odot$ on $[\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$ restricts to a monoidal product on the free coproduct completion $\Sigma\mathbb{C}$ of $\mathbb{C}$, yielding a distributive monoidal category $(\Sigma\mathbb{C}, e, \odot)$.*

Proof. The Day convolution is a coend construction and thus preserves coproducts. Hence $\Sigma\mathbb{C}$ is closed under $\odot$, and $\odot$ distributes over coproducts:

$$\Big(\sum_{i\in I} c_i\Big) \odot \Big(\sum_{j\in J} d_j\Big) \cong \sum_{i\in I}\sum_{j\in J}(c_i \odot d_j) \cong \sum_{(i,j)\in I\times J}(c_i \cdot d_j). \tag{2}$$

□

Eq. (2) tells us how to evaluate $\odot$ on arbitrary objects in $\Sigma\mathbb{C}$. We dualize this construction to obtain an analogous monoidal product on $\Pi\mathbb{C} \approx (\Sigma\mathbb{C}^{\mathsf{op}})^{\mathsf{op}}$.

Proposition 3.2. *There is a monoidal structure on $\Pi\mathbb{C}$ with unit e whose monoidal product $\circledcirc$ is given by*

$$\Big(\prod_{a\in A} c_a\Big) \circledcirc \Big(\prod_{b\in B} d_b\Big) \cong \prod_{(a,b)\in A\times B}(c_a \cdot d_b).$$

Thus, to obtain a monoidal structure on $\Sigma\Pi\mathbb{C}$, we may first lift the monoidal structure on $\mathbb{C}$ to $\Pi\mathbb{C}$, then lift the monoidal structure on $\Pi\mathbb{C}$ to $\Sigma\Pi\mathbb{C}$.

Proposition 3.3. *There is a monoidal structure on $\Sigma\Pi\mathbb{C}$ with unit e whose monoidal product, which we call the **parallel product** and denote by $\otimes$, is given by*

$$\Big(\sum_{i\in I}\prod_{a\in A_i} c_{i,a}\Big) \otimes \Big(\sum_{j\in J}\prod_{b\in B_j} d_{j,b}\Big) \cong \sum_{i\in I}\sum_{j\in J}\prod_{a\in A_i}\prod_{b\in B_j}(c_{i,a} \cdot d_{j,b}).$$

Proof. Apply Proposition 3.2 on $(\mathbb{C}, e, \odot)$ to obtain $(\Pi\mathbb{C}, e, \circledcirc)$, then apply Proposition 3.1 on $(\Pi\mathbb{C}, e, \circledcirc)$ to obtain $(\Sigma\Pi\mathbb{C}, e, \otimes)$. □

Example 3.4. To justify our use of the name "parallel product," we consider an example. Let $\mathbb{C} := \mathbb{1}$, whose unique object we call y. There is a unique monoidal structure on $\mathbb{C}$ given by $y \cdot y = y$.

Following [20], in $\Sigma\Pi\mathbb{1} \approx$ **Poly**, an object $\sum_{i\in I}\prod_{a\in A_i} y$, which we denote by $\sum_{i\in I} y^{A_i}$ for short, can be thought of as an ***interface***, with a number of possible *positions* from I it could expose and, according to the position $i \in I$ it is currently exposing, a number of possible *directions* from A_i it could receive. A morphism $\varphi\colon \sum_{i\in I} y^{A_i} \to \sum_{i'\in I'} y^{A'_{i'}}$ in **Poly** can then be viewed as an ***interaction protocol*** between interfaces. On positions, φ converts any position $i \in I$ that the domain could expose to a position $\varphi i \in I'$ for the codomain to expose; then on directions, φ converts any direction $a' \in A'_{\varphi i}$ that the codomain could receive to a direction $\varphi_i^\sharp a' \in A_i$ for the domain to receive.

Then taking the parallel product of two such interaction protocols yields a single interaction protocol that models the two original protocols simultaneously—or in *parallel*. More concretely, given interaction protocols $\varphi\colon \sum_{i\in I} y^{A_i} \to \sum_{i'\in I'} y^{A'_{i'}}$ and $\psi\colon \sum_{j\in J} y^{B_j} \to \sum_{j'\in J'} y^{B'_{j'}}$, their parallel product $\varphi \otimes \psi$ converts a pair of positions $(i,j) \in I \times J$ from its domain

$$\Big(\sum_{i\in I} y^{A_i}\Big) \otimes \Big(\sum_{j\in J} y^{B_j}\Big) \cong \sum_{(i,j)\in I\times J} (y \cdot y)^{A_i \times B_j} \cong \sum_{(i,j)\in I\times J} y^{A_i \times B_j}$$

to the pair of positions $(\varphi i, \psi j) \in I' \times J'$ from its codomain

$$\Big(\sum_{i'\in I'} y^{A'_{i'}}\Big) \otimes \Big(\sum_{j'\in J'} y^{B'_{j'}}\Big) \cong \sum_{(i',j')\in I'\times J'} y^{A'_{i'} \times B'_{j'}}$$

by applying the on-positions functions of φ and ψ in parallel; then converts a pair of directions $(a',b') \in A'_{\varphi i} \times B'_{\psi j}$ from its codomain to the pair of directions $(\varphi_i^\sharp a', \psi_j^\sharp b')$ from its domain by applying the on-directions functions of φ and ψ in parallel. ◊

3.2 The composition product

Here we introduce another monoidal structure on $\Sigma\Pi\mathbb{C}$ induced by the monoidal product on $\mathbb{C}$.

Definition 3.5. The ***composition product*** $\triangleleft$ of two objects in $\Sigma\Pi\mathbb{C}$ is given by

$$\Big(\sum_{i\in I}\prod_{a\in A_i} u_{i,a}\Big) \triangleleft \Big(\sum_{j\in J}\prod_{b\in B_j} v_{j,b}\Big) \coloneqq \sum_{i\in I}\prod_{a\in A_i}\sum_{j\in J}\prod_{b\in B_j}(u_{i,a}\cdot v_{j,b}). \qquad \Diamond$$

We call this the *composition* product as it generalizes the composition operation on polynomial functors when $\mathbb{C} = \mathbb{1}$: composing $\sum_{i\in I}\prod_{a\in A_i} y$ with $\sum_{j\in J}\prod_{b\in B_j} y$ yields the functor $\sum_{i\in I}\prod_{a\in A_i}\sum_{j\in J}\prod_{b\in B_j} y$. Distributivity, as given by Eq. (1), yields the following alternate form for this product.

Lemma 3.6. *The composition product can be rewritten as*

$$\Big(\sum_{i\in I}\prod_{a\in A_i} u_{i,a}\Big) \triangleleft \Big(\sum_{j\in J}\prod_{b\in B_j} v_{j,b}\Big) \cong \sum_{i\in I}\sum_{j\colon A_i\to J}\prod_{a\in A_i}\prod_{b\in B_{ja}}(u_{i,a}\cdot v_{ja,b}).$$

Proposition 3.7. *There is a monoidal category* $(\Sigma\Pi\mathbb{C}, e, \triangleleft)$.

Proof. Routine, but we will describe the behavior of $\triangleleft$ on morphisms: given

$$\varphi\colon \sum_{i\in I}\prod_{a\in A_i} u_{i,a} \to \sum_{k\in K}\prod_{c\in C_k} w_{k,c} \quad \text{and} \quad \psi\colon \sum_{j\in J}\prod_{b\in B_j} v_{j,b} \to \sum_{\ell\in L}\prod_{d\in D_\ell} x_{\ell,d},$$

the morphism

$$\varphi \triangleleft \psi\colon \sum_{i\in I}\sum_{j\colon A_i\to J}\prod_{a\in A_i}\prod_{b\in B_{ja}}(u_{i,a}\cdot v_{ja,b}) \to \sum_{k\in K}\sum_{\ell\colon C_k\to L}\prod_{c\in C_k}\prod_{d\in D_{\ell c}}(w_{k,c}\cdot x_{\ell c,d})$$

(whose domain and codomain we have rewritten using Lemma 3.6) consists of the following data:

- an *on-positions function* $\varphi \triangleleft \psi \colon \sum_{i \in I} J^{A_i} \to \sum_{k \in K} L^{C_k}$ consisting of:
  - a function $I \to K$ given by φ;
  - for each $i \in I$, a function $J^{A_i} \to L^{C_{\varphi i}}$ given by precomposing $\varphi_i^\sharp \colon C_{\varphi i} \to A_i$ and postcomposing $\psi \colon J \to L$;
- for each $i \in I$ and $j \colon A_i \to J$, sent to $\varphi i \in K$ and $\psi j \varphi_i^\sharp \colon C_{\varphi i} \to L$ by the on-positions function, an *on-directions function* $(\varphi \triangleleft \psi)_{i,j}^\sharp \colon \sum_{c \in C_{\varphi i}} D_{\psi j \varphi_i^\sharp c} \to \sum_{a \in A_i} B_{ja}$ consisting of:
  - a function $C_{\varphi i} \to A_i$ given by $\varphi_i^\sharp$;
  - for each $c \in C_{\varphi i}$, a function $D_{\psi j \varphi_i^\sharp c} \to B_{j \varphi_i^\sharp c}$ given by $\psi^\sharp_{j \varphi_i^\sharp c}$;
- for each $i \in I, j \colon A_i \to J, c \in C_{\varphi i}$, and $d \in D_{\psi j \varphi_i^\sharp c}$, sent to $\varphi_i^\sharp \colon C_{\varphi i} \to A_i$ and $\psi^\sharp_{j \varphi_i^\sharp c} \colon D_{\psi j \varphi_i^\sharp c} \to B_{j \varphi_i^\sharp c}$ by the on-directions function, an *on-predicates map* $(\varphi \triangleleft \psi)_{i,j,c,d} \colon u_{i, \varphi_i^\sharp c} \cdot v_{j', \psi_{j'}^\sharp d} \to w_{\varphi i, c} \cdot x_{\psi j', d}$ (here $j' \coloneqq j \varphi_i^\sharp c$) given by $\varphi_{i,c} \cdot \psi_{j',d}$. □

4 Composition comonoids as enriched categories

Our main result concerns the category of comonoids in $(\Sigma\Pi\mathbb{C}, e, \triangleleft)$. We will show that it is equivalent to a category whose objects are enriched categories and whose morphisms are enriched cofunctors. While the former may be more familiar than the latter, we review both these definitions here.

Recall the definition of a *category enriched over a monoidal category* from Kelly [15]. We restate it here for the special case where the enriching category is $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$.

Definition 4.1. A ***small*** $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$***-enriched category*** $\mathcal{A}$, with $\odot$ defined as in Proposition 3.1, consists of the following data:

- a set $\operatorname{Ob}\mathcal{A}$ (or just $\mathcal{A}$) of **objects**;
- for each $x, y \in \operatorname{Ob}\mathcal{A}$, a **hom-family** $\sum_{f \colon x \to y} |f| \in \Sigma\mathbb{C}^{\mathsf{op}}$ consisting of:
  - a set $\mathcal{A}(x,y)$ of **morphisms**, i.e. a **hom-set**, with $f \in \mathcal{A}(x,y)$ denoted by $f \colon x \to y$;
  - for each morphism $f \colon x \to y$, a **weight** $|f| \in \mathbb{C}$;
- for each $x \in \operatorname{Ob}\mathcal{A}$, a morphism $e \to \sum_{f \colon x \to x} |f|$ in $\Sigma\mathbb{C}^{\mathsf{op}}$ consisting of:
  - an **identity morphism** $\mathrm{id}_x \colon x \to x$;
  - an **identity map** $\eta_x \colon |\mathrm{id}_x| \to e$ from $\mathbb{C}$;
- for each $x, y, z \in \operatorname{Ob}\mathcal{A}$, a morphism

$$\sum_{f \colon x \to y} \sum_{g \colon y \to z} (|f| \cdot |g|) \to \sum_{h \colon x \to z} |h|$$

  in $\Sigma\mathbb{C}^{\mathsf{op}}$ consisting of, for each $f \colon x \to y$ and $g \colon y \to z$:
  - a **composite morphism** $gf \colon x \to z$;
  - a **composite map** $\mu_{f,g} \colon |gf| \to |f| \cdot |g|$ from $\mathbb{C}$.

Here $w, x, y, z \in \operatorname{Ob}\mathcal{A}$ and $f \colon w \to x$, $g \colon x \to y$, and $h \colon y \to z$ must satisfy the following:

- **unitality**, that $f\,\mathrm{id}_w = f = \mathrm{id}_x f$ and the following diagram commutes in $\mathbb{C}$, up to unitors:

$$\begin{tikzcd} {|\mathrm{id}_w| \cdot |f|} \arrow[dr, "\eta_w \cdot |f|"'] & {|f|} \arrow[l, "\mu_{\mathrm{id}_w, f}"'] \arrow[r, "\mu_{f, \mathrm{id}_x}"] \arrow[d, equal] & {|f| \cdot |\mathrm{id}_x|} \arrow[dl, "|f| \cdot \eta_x"] \\ & {|f|} & \end{tikzcd}$$

- **associativity**, that $(hg)f = h(gf)$ and the following commutes in $\mathbb{C}$, up to associators:

$$\begin{tikzcd} {|hgf|} \arrow[r, "\mu_{f,hg}"] \arrow[d, "\mu_{gf,h}"'] & {|f| \cdot |hg|} \arrow[d, "|f| \cdot \mu_{g,h}"] \\ {|gf| \cdot |h|} \arrow[r, "\mu_{f,g} \cdot |h|"] & {|f| \cdot |g| \cdot |h|} \end{tikzcd}$$

◊

By Proposition 3.1, the monoidal category $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$ is distributive, so there exists a notion of a $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$*-enriched cofunctor* as introduced by Clarke and Di Meglio [6]. We restate the definition of an enriched cofunctor in this special case here.

Definition 4.2. A $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$***-enriched cofunctor*** $\Phi \colon \mathcal{A} \nrightarrow \mathcal{B}$ between small $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched categories $\mathcal{A}$ and $\mathcal{B}$ consists of the following data:

- a function $\Phi \colon \operatorname{Ob}\mathcal{A} \to \operatorname{Ob}\mathcal{B}$;
- for each $a \in \mathcal{A}, b \in \mathcal{B}$, and morphism $f \colon \Phi a \to b$ from $\mathcal{B}$:
  - a morphism $\Phi_a^\sharp f \colon a \to x$ from $\mathcal{A}$ with $\Phi x = b$;
  - a morphism $\Phi_{a,f} \colon |\Phi_a^\sharp f| \to |f|$ from $\mathbb{C}$.

Here $a, x \in \mathcal{A}$; $b, b' \in \mathcal{B}$; $f \colon \Phi a \to b$ with $\Phi_a^\sharp f \colon a \to x$; and $g \colon b \to b'$ must satisfy:

- **preservation of identities**, that $\Phi_a^\sharp(\mathrm{id}_{\Phi a}) = \mathrm{id}_a$ and the following commutes in $\mathbb{C}$:

$$\begin{tikzcd} {|\,\mathrm{id}_a|} \arrow[r, "\Phi_{a,\mathrm{id}_{\Phi a}}"] \arrow[dr, "\eta_a"'] & {|\,\mathrm{id}_{\Phi a}|} \arrow[d, "\eta_{\Phi a}"] \\ & e \end{tikzcd}$$

- **preservation of composites**, that $\Phi_a^\sharp(gf) = \Phi_x^\sharp(g)\Phi_a^\sharp(f)$ and the following commutes in $\mathbb{C}$:

$$\begin{tikzcd} {|\Phi_a^\sharp(gf)|} \arrow[r, "\mu_{\Phi_a^\sharp f, \Phi_x^\sharp g}"] \arrow[d, "\Phi_{a,gf}"'] & {|\Phi_a^\sharp f| \cdot |\Phi_x^\sharp g|} \arrow[d, "\Phi_{a,f} \cdot \Phi_{x,g}"] \\ {|gf|} \arrow[r, "\mu_{f,g}"] & {|f| \cdot |g|} \end{tikzcd}$$

◊

There is then a category whose objects are small $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched categories and whose morphisms are $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched cofunctors. While enriched cofunctors differ from enriched functors, it is nevertheless the case that an isomorphism in this category corresponds to our usual notion of isomorphism of enriched categories as defined by a pair of invertible enriched functors.

The following is a generalization of a result by Ahman and Uustalu [2, 3]: that the category of polynomial comonads is equivalent to the category of small categories and cofunctors, corresponding to the case where $\mathbb{C} = \mathbb{1}$ (the **Set**-enriched case, for $\Sigma\mathbb{1}^{\mathsf{op}} \approx$ **Set**) in the theorem below.

Theorem 4.3. *The category of comonoids in the monoidal category* $(\Sigma\Pi\mathbb{C}, e, \triangleleft)$ *is equivalent to the category of small* $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$*-enriched categories and enriched cofunctors.*

Proof. First, we describe how to construct a comonoid in $(\Sigma\Pi\mathbb{C}, e, \triangleleft)$ from each $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched category; the inverse construction will then be evident. Given a small $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched category $\mathcal{A}$, define a polynomial in $\mathbb{C}$ with positions $\mathrm{Ob}\,\mathcal{A}$, directions $A_i := \sum_{j\in\mathcal{A}} \mathcal{A}(i,j)$ for $i \in \mathcal{A}$, and predicate $|a| \in \mathbb{C}$ for $i \in \mathcal{A}$ and $(j, a\colon i \to j) \in A_i$. In other words: positions are objects, directions are morphisms of a given domain, and predicates are the morphisms' weights. We endow this polynomial $\sum_{i\in\mathcal{A}} \prod_{a\colon i\to\_} |a|$ (where $a\colon i \to \_$ denotes a morphism a in $\mathcal{A}$ with domain i and arbitrary codomain) with a comonoid structure as follows. Its counit

$$\varepsilon\colon \sum_{i\in\mathcal{A}} \prod_{a\colon i\to\_} |a| \to e$$

is trivial on positions, the assignment $i \mapsto \mathrm{id}_i$ on directions, and the identity map $\eta_i\colon |\mathrm{id}_i| \to e$ on predicates. Meanwhile its comultiplication

$$\delta\colon \sum_{i\in\mathcal{A}} \prod_{a\colon i\to\_} |a| \to \Big(\sum_{i\in\mathcal{A}} \prod_{b\colon i\to\_} |b|\Big) \triangleleft \Big(\sum_{j\in\mathcal{A}} \prod_{c\colon j\to\_} |c|\Big) \cong \sum_{i\in\mathcal{A}} \sum_{j\colon A_i\to\mathrm{Ob}\,\mathcal{A}} \prod_{b\colon i\to\_} \prod_{c\colon jb\to\_} |b|\cdot|c|$$

is the assignment $i \mapsto (i, \mathrm{cod})$ on positions, where $\mathrm{cod}\colon A_i \to \mathrm{Ob}\,\mathcal{A}$ sends each morphism $a\colon i \to j$ to its codomain j; morphism composition on directions, sending $b\colon i \to \_$ and $c\colon \mathrm{cod}(b) \to \_$ to $cb\colon i \to \_$; and the composite map $\mu_{b,c}\colon |cb| \to |b|\cdot|c|$ on predicates. The counitality and coassociativity of the comonoid follow from the unitality and associativity of the enriched category, as well as the equations $\mathrm{cod}(\mathrm{id}_i) = i$ and $\mathrm{cod}(cb) = \mathrm{cod}(c)$. Moreover, from any comonoid we can recover its corresponding enriched category up to isomorphism.

Next, we describe how to construct a morphism of comonoids in $(\Sigma\Pi\mathbb{C}, e, \triangleleft)$ from each $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched cofunctor; again the inverse construction will then be evident. Given a $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched cofunctor $\Phi\colon \mathcal{A} \nrightarrow \mathcal{B}$ between small $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched categories $\mathcal{A}$ and $\mathcal{B}$, we construct a structure-preserving morphism

$$\varphi\colon \sum_{i\in\mathcal{A}} \prod_{a\colon i\to\_} |a| \to \sum_{j\in\mathcal{B}} \prod_{b\colon j\to\_} |b|$$

in $\Sigma\Pi\mathbb{C}$ between the comonoids corresponding to $\mathcal{A}$ and $\mathcal{B}$ like so. On positions, set $\varphi i := \Phi i \in \mathcal{B}$ for $i \in \mathcal{A}$; on directions, set $\varphi_i^\sharp b := (\Phi_i^\sharp b\colon i \to \_)$ in $\mathcal{A}$ for $i \in \mathcal{A}$ and $b\colon \Phi i \to \_$ in $\mathcal{B}$; and on predicates, set $\varphi_{i,b} := (\Phi_{i,b}\colon |\Phi_i^\sharp b| \to |b|)$ in $\mathbb{C}$ for $i \in \mathcal{A}$ and $b\colon \Phi i \to \_$ in $\mathcal{B}$. That φ preserves counits and comultiplications follows from the fact that Φ preserves identities and composites and that $\Phi(\mathrm{cod}(\Phi_i^\sharp a)) = \mathrm{cod}(a)$. Moreover, from any comonoid morphism we can recover its corresponding enriched cofunctor. $\square$

5 Application: compositional bounds on dynamical systems

Here we give an example of how the structure of $\Sigma\Pi\mathbb{C}$ may be used to model open dynamical systems and their invariants. This case study is by no means comprehensive; we seek only to hint at the possibilities of how $\Sigma\Pi\mathbb{C}$ may be used.

Throughout, we let $(\mathbb{C}, e, \cdot) := ([0,\infty]_{\le}, 0, +)$, the poset of nonnegative extended reals ordered by $\le$ viewed as a category and endowed with the additive monoidal structure. We take the free coproduct completion of its opposite category and endow it with a monoidal structure $\oplus$ given by Day convolution. Then a $(\Sigma[0,\infty]_{\ge}, 0, \oplus)$-enriched category is an *additively weighted category* [12].

Definition 5.1. An ***additively weighted category*** (or ***weighted category***) $\mathcal{X}$ is a small $(\Sigma[0,\infty]_{\geq}, 0, \oplus)$-enriched category. It thus consists of the following data:

- a set $\mathrm{Ob}\,\mathcal{X}$ of **objects** or **points**;
- for each $x, y \in \mathrm{Ob}\,\mathcal{X}$, an object $\sum_{p\colon x\to y} |p| \in \Sigma[0,\infty]_{\geq}$ consisting of:
  - a set $\mathcal{X}(x,y)$ of **morphisms** or **paths**, with $p \in \mathcal{X}(x,y)$ denoted by $p\colon x \to y$;
  - for each path $p\colon x \to y$, a **weight** or **cost** $|p| \in [0,\infty]$;
- for each $x \in \mathrm{Ob}\,\mathcal{X}$, a morphism $0 \to \sum_{f\colon x\to x} |f|$ in $\Sigma[0,\infty]_{\geq}$ consisting of:
  - a **constant path** $\mathrm{id}_x\colon x \to x$,
  - satisfying **nonpositivity**: $|\mathrm{id}_x| \leq 0$, and thus $|\mathrm{id}_x| = 0$;
- for each $x, y, z \in \mathrm{Ob}\,\mathcal{X}$, a morphism

$$\sum_{f\colon x\to y}\sum_{g\colon y\to z}(|f|+|g|) \to \sum_{h\colon x\to z}|h|$$

  in $\Sigma[0,\infty]_{\geq}$ consisting of, for each $f\colon x \to y$ and $g\colon y \to z$:
  - a **composite path** $gf\colon x \to z$,
  - satisfying the **triangle inequality**: $|gf| \leq |f| + |g|$. ◊

A weighted category $\mathcal{X}$ with $|\mathcal{X}(x,y)| = 1$ for all $x, y \in \mathrm{Ob}\,\mathcal{X}$ is a Lawvere metric space [17].

By Theorem 4.3, a weighted category $\mathcal{X}$, defined above as an enriched category, is equivalently a comonoid in $(\Sigma\Pi[0,\infty]_{\leq}, 0, \triangleleft)$. Then we can define a discrete dynamical system on $\mathcal{X}$ in terms of the category $\Sigma\Pi[0,\infty]_{\leq}$ as follows.

Definition 5.2. A ***discrete dynamical system*** on a weighted category $\mathcal{X}$, viewed as a comonoid object $\mathcal{X} \in \Sigma\Pi[0,\infty]_{\leq}$, is a morphism $\varphi\colon \mathcal{X} \to \infty$ in $\Sigma\Pi[0,\infty]_{\leq}$. It thus consists of the following data:

- a trivial *on-positions function* $\varphi\colon\ \mathrm{Ob}\,\mathcal{X} \to 1$;
- for each point $x \in \mathrm{Ob}\,\mathcal{X}$, an *on-directions function* $\varphi^\sharp_x\colon 1 \to \sum_{y\in\mathcal{X}} \mathcal{X}(x,y)$ that picks out a path $\varphi^\sharp_x$ from x to some other point,
- satisfying the trivial inequality $|\varphi^\sharp_x| \leq \infty$. ◊

In other words, a discrete dynamical system on $\mathcal{X}$ assigns to each point x in $\mathcal{X}$ a path $\varphi^\sharp_x\colon x \to x_1$ out of that point. The intuition is that starting from x, the system moves to a new point x_1 along the path $\varphi^\sharp_x$ in one time step. We can "run" the system by taking the n-fold composition product $\varphi^{\triangleleft n}$ for $n \in \mathbb{N}$ and composing with the canonical n-ary comultiplication δ^{n-1} of $\mathcal{X}$ provided by its comonoid structure:[4]

$$\mathcal{X} \xrightarrow{\delta^{n-1}} \mathcal{X}^{\triangleleft n} \xrightarrow{\varphi^{\triangleleft n}} \infty^{\triangleleft n} \cong \infty + \cdots + \infty \cong \infty. \tag{3}$$

This is a new discrete dynamical system that assigns to each point x in $\mathcal{X}$ the n-fold composite of paths

$$x \xrightarrow{\varphi^\sharp_x} x_1 \xrightarrow{\varphi^\sharp_{x_1}} x_2 \xrightarrow{\varphi^\sharp_{x_2}} \cdots \xrightarrow{\varphi^\sharp_{x_{n-1}}} x_n \tag{4}$$

from $\mathcal{X}$, mapping out the evolution of the dynamical system after n time steps. Similarly, given another discrete dynamical system $\psi\colon \mathcal{X} \to \infty$, we can compose it with the first to obtain a third system that runs one before the other:

$$\mathcal{X} \xrightarrow{\delta} \mathcal{X} \triangleleft \mathcal{X} \xrightarrow{\varphi \triangleleft \psi} \infty \triangleleft \infty \cong \infty.$$

Furthermore, we could repackage the data of a discrete dynamical system as an enriched cofunctor by the following proposition, where $\mathcal{X}$ is a weighted category viewed as a comonoid object $\mathcal{X} \in \Sigma\Pi[0,\infty]_{\leq}$.

[4] We inductively define $\delta^1 := \delta$ and $\delta^n := (\mathrm{id}_{\mathcal{X}^{\triangleleft(n-1)}} \triangleleft \delta) \circ \delta^{n-1}$.

Proposition 5.3. *There is a natural correspondence between discrete dynamical systems $\varphi\colon \mathcal{X} \to \infty$ and enriched cofunctors $\Phi\colon \mathcal{X} \nrightarrow \prod_{n\in\mathbb{N}} \infty$, whose codomain is the one-object $(\Sigma[0,\infty]_{\geq}, 0, \oplus)$-enriched category with hom-set $\mathbb{N}$, addition as composition, and all weights infinite.*

Proof. Given φ, construct Φ by setting $\Phi_x^\sharp(n)$ to the composite path defined in (4) for $x \in \mathrm{Ob}\,\mathcal{X}$ and $n \in \mathbb{N}$ (with $\Phi_x^\sharp(0) := \mathrm{id}_x$); the cofunctor laws follow immediately. Conversely, given Φ, construct φ by setting $\varphi_x^\sharp := \Phi_x^\sharp(1)$. These constructions are natural and mutually inverse. □

Thus discrete dynamical systems on $\mathcal{X}$ are precisely enriched cofunctors $\mathcal{X} \nrightarrow \prod_{n\in\mathbb{N}} \infty$. We could generalize how these systems run by replacing $\mathbb{N}$ with some other monoid, or indeed by replacing the entire codomain by a different weighted category, which could in turn be acted on via an enriched cofunctor to another weighted category, and so forth—suggesting the versatility of comonoids in $\Sigma\Pi\mathbb{C}$ for modeling general interactions.

So far, the examples we have described could have been done in $\Sigma\Pi\mathbb{1} \approx$ **Poly** (indeed, the material so far is adapted from [20, 25]); we have yet to make use of the enriched structure. Now we will put finite weights in the codomains of our systems to bound their behavior.

Definition 5.4. A discrete dynamical system $\varphi\colon \mathcal{X} \to \infty$ is ***bounded (above) by*** $r \in [0,\infty]$ if φ factors through the morphism $r \to \infty$ in $[0,\infty]_{\leq} \subset \Sigma\Pi[0,\infty]_{\leq}$. Equivalently, for each point $x \in \mathrm{Ob}\,\mathcal{X}$, the path $\varphi_x^\sharp$ has cost at most r. ◊

Boundedness is well-behaved under composition: if $\varphi\colon \mathcal{X} \to \infty$ factors through r as $\overline{\varphi}\colon \mathcal{X} \to r$, then the n-fold composition product $\varphi^{\triangleleft n}\colon \mathcal{X}^{\triangleleft n} \to \infty^{\triangleleft n} \cong \infty$ factors through $r^{\triangleleft n} \cong nr$ as $\overline{\varphi}^{\triangleleft n}\colon \mathcal{X}^{\triangleleft n} \to r^{\triangleleft n}$. Hence the n-fold composite dynamical system $\varphi^{\triangleleft n} \circ \delta^{n-1}$ from (3) must factor through nr as well, so it is a discrete dynamical system bounded by nr. This coincides with our intuition: if the cost of every time step of a dynamical system is bounded above by r, then the cost of n successive time steps must be bounded above by nr. We thus have the following result, generalizing Proposition 5.3.

Proposition 5.5. *There is a natural correspondence between discrete dynamical systems $\varphi\colon \mathcal{X} \to \infty$ bounded above by $r \in [0,\infty]$ and enriched cofunctors $\Phi\colon \mathcal{X} \nrightarrow \prod_{n\in\mathbb{N}} nr$, whose codomain is the one-object $(\Sigma[0,\infty]_{\geq}, 0, \oplus)$-enriched category with hom-set $\mathbb{N}$, addition as composition, and weights $|n| := nr$.*

Proof. The construction mirrors the one in the proof of Proposition 5.3; we need only verify that the additional restrictions on costs are satisfied. Given φ bounded by r, the n-fold composite path from (4) has cost at most nr, ensuring $|\Phi_x^\sharp(n)| \leq nr$. Conversely, given Φ, we have $|\varphi_x^\sharp| = |\Phi_x^\sharp(1)| \leq r$. □

The preceding material is only a sample of how $\Sigma\Pi[0,\infty]_{\geq}$ and, by extension, $\Sigma\Pi\mathbb{C}$ may be used to model compositional behavioral patterns of dynamical systems. We could generalize the codomain of our discrete dynamical systems beyond one-position, one-direction polynomials in $\mathbb{C}$; we could generalize $\mathbb{C}$ beyond mere posets; and so forth. Indeed, Propositions 5.3 and 5.5 are special cases of a far more general result.

Theorem 5.6. *The forgetful functor from comonoids in $(\Sigma\Pi\mathbb{C}, e, \triangleleft)$ to their underlying polynomials has a right adjoint, yielding cofree $(\Sigma\mathbb{C}^{\mathsf{op}}, e, \odot)$-enriched categories on polynomials in $\mathbb{C}$.*

Sketch of proof. The construction follows the analogous result for cofree polynomial comonads as detailed in [20]. There the cofree category on a given polynomial has tuples of the polynomial's directions as morphisms; we then assign each tuple a weight in $\mathbb{C}$ equal to the monoidal product of the predicates of the directions in the tuple. □

6 Future directions

We close with future directions for research in addition to the potential applications already suggested.

Foundations of polynomial categories

Spivak surveys categorical properties and structures on **Poly** $\approx \Sigma\Pi\mathbb{1}$ in [26]; in addition to those we have already covered, it would be instructive to examine which of these properties and structures carry over to $\Sigma\Pi\mathbb{C}$, perhaps requiring various conditions on $\mathbb{C}$. Similarly, we could investigate how known structures on $\mathbf{Dial}(\mathbf{Set}) \approx \mathbf{Hmg}(\mathbb{2})$ carry over to $\mathbf{Hmg}(\mathbb{C})$.

Interaction between monoidal structures on polynomials

Spivak observed that $\triangleleft$ is duoidal over $\otimes$ in the case of $\mathbb{C} := \mathbb{1}$, i.e. there is a natural transformation $(-\triangleleft-)\otimes(-\triangleleft-) \to (-\otimes-)\triangleleft(-\otimes-)$ satisfying various coherence conditions [25]. Shapiro and Spivak go on to leverage this duoidality to model compositional dependence [22]. We hope to generalize their results to the parallel and compositional products on $\Sigma\Pi\mathbb{C}$.

Other monoidal structures on polynomials

Given a monoidal category $(\mathbb{C}, e, \cdot)$, there are at least two other monoidal structures on $\Sigma\Pi\mathbb{C}$ with unit e: one given by

$$\Big(\sum_{i\in I}\prod_{a\in A_i} u_{i,a}\Big)\bowtie\Big(\sum_{j\in J}\prod_{b\in B_j} v_{j,b}\Big) := \sum_{i\in I}\sum_{j\in J}\prod_{a\colon J\to A_i}\prod_{b\colon I\to B_j}(u_{i,aj}\cdot v_{j,bi})$$

and another given by

$$\Big(\sum_{i\in I}\prod_{a\in A_i} u_{i,a}\Big)\rtimes\Big(\sum_{j\in J}\prod_{b\in B_j} v_{j,b}\Big) := \sum_{i\in I}\sum_{j\in J}\prod_{a\colon J\to A_i}\prod_{b\in B_j}(u_{i,aj}\cdot v_{j,b}).$$

We would like to know if there are interpretations or applications for these monoidal products as there are for the parallel and composition products.

Recovering categories and cofunctors enriched over any distributive category

Theorem 4.3 recovers the category of small categories and cofunctors enriched over a free coproduct completion with Day convolution as the category of comonoids of a particular monoidal category. Yet Clarke and Di Meglio described how cofunctors may be enriched over any distributive monoidal category [6]. The free coproduct completion with Day convolution gives us a way to freely construct a distributive monoidal category from any monoidal category, but not every distributive monoidal category arises this way. We would like to know if our theorem may be generalized to recover categories of small categories and cofunctors enriched over any distributive monoidal category as some category of comonoids.

References

[1] Marcelo Aguiar (1997): *Internal categories and quantum groups*, PhD thesis. Cornell University. Available at `https://pi.math.cornell.edu/~maguiar/thesis2.pdf`.

[2] Danel Ahman & Tarmo Uustalu (2016): *Directed Containers as Categories*. *Electronic Proceedings in Theoretical Computer Science*. 207, pages 89–98, doi: `10.4204/eptcs.207.5`.

[3] Danel Ahman & Tarmo Uustalu (2017): *Taking Updates Seriously*. In *BX@ETAPS*. Available at `https://iris.rais.is/en/publications/taking-updates-seriously`.

[4] Robert Atkey, Bruno Gavranović, Neil Ghani, Clemens Kupke, Jérémy Ledent & Fredrik Nordvall Forsberg (2021): *Compositional Game Theory, Compositionally*. *Electronic Proceedings in Theoretical Computer Science*. 333, pages 198–214, doi: `10.4204/eptcs.333.14`.

[5] Matteo Capucci (2023): *Diegetic Representation of Feedback in Open Games*. *Electronic Proceedings in Theoretical Computer Science*. 380, pages 145–158, doi: `10.4204/eptcs.380.9`.

[6] Bryce Clarke & Matthew Di Meglio (2022): *An introduction to enriched cofunctors*, doi: `10.48550/arXiv.2209.01144`.

[7] Geoffrey SH Cruttwell, Bruno Gavranović, Neil Ghani, Paul Wilson & Fabio Zanasi (2022): *Categorical foundations of gradient-based learning*. In *Programming Languages and Systems: 31st European Symposium on Programming, ESOP 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2–7, 2022, Proceedings*, Springer International Publishing Cham, pages 1–28, doi: `10.1007/978-3-030-99336-8_1`.

[8] Brian Day (1970): *Construction of biclosed categories*, PhD thesis. University of New South Wales. doi: `10.26190/unsworks/8048`.

[9] Brendan Fong & Michael Johnson (2019): *Lenses and Learners*, doi: `10.48550/arXiv.1903.03671`.

[10] Brendan Fong, David Spivak & Rémy Tuyéras (2019): *Backprop as Functor: A compositional perspective on supervised learning*. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, IEEE, pages 1–13, doi: `10.1109/LICS.2019.8785665`, arXiv: `1711.10455`.

[11] Jeremy Gibbons & Michael Johnson (2012): *Relating Algebraic and Coalgebraic Descriptions of Lenses*. *Proc. of 1st Int. Wksh. on Bidirectional Transformations, BX 2012 (Tallinn, March 2012)*. 49, doi: `10.14279/tuj.eceasst.49.726`.

[12] Marco Grandis (2006): *Categories, norms and weights*, doi: `10.48550/arXiv.math/0603298`.

[13] Jules Hedges (2017): *Coherence for lenses and open games*, doi: `10.48550/arXiv.1704.02230`.

[14] Michael Johnson, Robert Rosebrugh & R. J. Wood (2012): *Lenses, fibrations and universal translations*. *Mathematical Structures in Computer Science*. 22(1), pages 25–42, doi: `10/bn76vr`.

[15] G. Max Kelly (1982): *Basic concepts of enriched category theory*. *Lecture Notes in Mathematics*, Cambridge University Press. Available at `http://www.tac.mta.ca/tac/reprints/articles/10/tr10abs.html`. Reprinted in *Reprints in Theory and Applications of Categories*.

[16] Elena Di Lavore, Wilmer Leal & Valeria de Paiva (2022): *Dialectica Petri nets*, doi: `10.48550/arXiv.2105.12801`.

[17] F. William Lawvere (1973): *Metric spaces, generalized logic, and closed categories*. *Rendiconti del seminario matématico e fisico di Milano*. 43, pages 135–166, doi: `10.1007/BF02924844`. Available at `http://www.tac.mta.ca/tac/reprints/articles/1/tr1abs.html`. Reprinted in *Reprints in Theory and Applications of Categories*.

[18] Sean Moss (2022): *Dependent products of polynomials*. Workshop on Polynomial Functors. Available at `https://youtu.be/tw08TmOORRo&t=7190s`.

[19] David Jaz Myers (2021): *Double Categories of Open Dynamical Systems (Extended Abstract)*. *Electronic Proceedings in Theoretical Computer Science*. 333, pages 154–167, doi: `10.4204/eptcs.333.11`.

[20] Nelson Niu & David I. Spivak (2023): *Polynomial Functors: A General Theory of Interaction*. Available at `https://topos.site/poly-book.pdf`. In preparation.

[21] Valeria Correa Vaz de Paiva (1991): *The Dialectica categories*, technical report UCAM-CL-TR-213. University of Cambridge, Computer Laboratory. doi: `10.48456/tr-213`.

[22] Brandon T. Shapiro & David I. Spivak (2022): *Duoidal Structures for Compositional Dependence*, doi: `10.48550/arXiv.2210.01962`.

[23] Brandon T. Shapiro & David I. Spivak (2023): *Dynamic Operads, Dynamic Categories: From Deep Learning to Prediction Markets*. *Electronic Proceedings in Theoretical Computer Science*. 380, pages 183–202, doi: `10.4204/eptcs.380.11`.

[24] Toby St. Clere Smithe (2023): *Open Dynamical Systems as Coalgebras for Polynomial Functors, with Application to Predictive Processing*. *Electronic Proceedings in Theoretical Computer Science*. 380, pages 307–330, doi: `10.4204/eptcs.380.18`.

[25] David I. Spivak (2020): *Poly: An abundant categorical setting for mode-dependent dynamics*, doi: `10.48550/arXiv.2005.01894`.

[26] David I. Spivak (2023): *A reference for categorical structures on* **Poly**, doi: `10.48550/arXiv.2202.00534`.

[27] Davide Trotta, Matteo Spadetto & Valeria de Paiva (2022): *Dialectica logical principles: not only rules*. *Journal of Logic and Computation*. 32(8), pages 1855–1875, doi: `10.1093/logcom/exac079`.

Syntax Monads for the Working Formal Metatheorist

Lawrence Dunn
University of Pennsylvania
Philadelphia, USA
dunnla@seas.upenn.edu

Val Tannen
University of Pennsylvania
Philadelphia, USA
val@cis.upenn.edu

Steve Zdancewic
University of Pennsylvania
Philadelphia, USA
stevez@cis.upenn.edu

Formally verifying the properties of formal systems using a proof assistant requires justifying numerous minor lemmas about capture-avoiding substitution. Despite work on category-theoretic accounts of syntax and variable binding, *raw, first-order* representations of syntax, the kind considered by many practitioners and compiler frontends, have received relatively little attention. Therefore applications miss out on the benefits of category theory, most notably the promise of reusing formalized infrastructural lemmas between implementations of different systems. Our Coq framework Tealeaves provides libraries of reusable infrastructure for a raw, locally nameless representation and can be extended to other representations in a modular fashion. In this paper we give a string-diagrammatic account of *decorated traversable monads* (DTMs), the key abstraction implemented by Tealeaves. We define DTMs as monoids of structured endofunctors before proving a representation theorem à la Kleisli, yielding a recursion combinator for finitary tree-like datatypes.

1 Introduction

Machine-certified proofs of the properties of programming languages, type theories, and other formal systems are increasingly critical for establishing confidence in the design and implementation of computer systems. Much of this reasoning is overtly concerned with the manipulation of syntactical structures, especially variable-binding constructs, making the representation of these structures a key issue in formal metatheory [6]. As implementations scale in complexity to realistic formalizations of compilers [39] and programming languages [24], often with many kinds of variables, the bookkeeping required to manipulate variables correctly becomes nearly prohibitive.

Category-theoretic accounts of syntax with variable binding (e.g. [8, 16, 17, 18, 2]) offer the tantalizing benefit of formalizing tedious syntax "infrastructure" once and for all over an abstract choice of signature, instead of repeating this effort for the particular syntax of each new system. However, the kind of syntax usually considered by theorists—often intrinsically well-typed with well-scoped de Bruijn indices—is different from what many working semanticists and compilers actually implement. Consequently, the benefits of a principled categorical framework are not yet available to many applications. This work lays the foundations of a category-theoretic account of variable binding as it often looks in practice, with the aim of building certified libraries of generic syntax infrastructure that can be used (and reused) in real-world applications.

Contributions. This manuscript makes two contributions.

- We introduce the strict monoidal category $\mathbf{DecTrav}_W$ of decorated-traversable endofunctors on **Set** for some monoid W (Definition 3.16) and define decorated-traversable monads (DTMs) as monoids in this category (Definition 3.17). Examples of decorated-traversable functors come from the signature functors of languages with variable binding; the free monads they generate are DTMs. These structures admit a string-diagram calculus, which aids in equational reasoning.

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 98–117, doi:10.4204/EPTCS.397.7

- We prove an equivalence (Theorem 4.2) between monoids in $\mathbf{DecTrav}_W$ and a Kleisli-style presentation (Definition 4.1) that describes a structured recursion combinator for abstract syntax trees.

As with ordinary (strong) monads [29], the Kleisli presentation is of more immediate utility from a functional programming or formal metatheory perspective, in part because the definition requires checking fewer axioms. In a previous, tool-oriented paper [15] we introduced Kleisli-presented DTMs and used them to derive generic syntax infrastructure for first-order representations of variable binding in Coq. However, that paper did not explain why the seemingly ad-hoc equational axioms should be considered "correct." This paper justifies the robustness of the axioms by proving their equivalence with a more clearly principled, string-diagrammatic set of axioms. The results in this paper have been formalized in Coq and are available in our GitHub repository.[1]

Layout. The rest is laid out as follows. Section 2 contains background on first-order representations of variable binding. We recall that abstract syntax trees, parameterized by the data in the leaves, naturally give rise to a (free) monad. For such monads, the Kleisli axiomatization provides a theory of naïve substitution, but this is not expressive enough to define the *capture-avoiding* substitution operations considered by different representations of variables. Section 3 introduces the endofunctor categories $\mathbf{Dec}_W$, $\mathbf{Trav}$, and $\mathbf{DecTrav}_W$. Section 4 derives a Kleisli-style characterization of monoids in $\mathbf{DecTrav}_W$ and explains why this abstraction solves the problems identified in Section 2. Section 5 contrasts our approach with related work. Section 6 concludes.

Functors in this paper have type $\mathbf{Set} \to \mathbf{Set}$ and typically represent parameterized container types like lists, binary trees, and abstract syntax trees. We recall that $\mathbf{End}_{\mathrm{Set}}$ is the strict monoidal category whose objects are endofunctors on $\mathbf{Set}$, whose arrows are natural transformations, and whose tensor product is given by composition of functors.

2 First-order Representations of Variable Binding

The modern formal metatheorist has many options for representing and manipulating terms with variable binding in a proof assistant. The first choice is whether to employ a *first-order* or *higher-order* approach. Higher-order strategies represent variable-binding constructors in the object language as higher-order functions in the metatheory; this sidesteps thorny issues like variable capture but does not shed much light on syntax as defined in, say, a compiler. We are interested in things like verified compilers, so we consider a first-order approach. This style is also simple, intuitive, and well supported by general-purpose proof assistants like Coq [36]. Theoretically it lends itself to the theory of initial algebras, the category-theoretic take on structural recursion [11].

A more or less orthogonal question is whether to consider an *intrinsic* or *extrinsic* (also called *raw*) representation. For instance, intrinsically well-scoped terms exist in some context Γ and can only mention free variables declared in Γ, while instrinsically well-typed terms essentially carry around their own typing judgment. The raw approach posits that a single set of terms simply exists, including ones that are ill-formed and untypable in the formal system. Properties like being well-scoped in Γ are then defined post-hoc as predicates on terms by structural recursion. We consider an extrinsic representation, though in future work we could investigate an intrinsic approach.

Finally, one has a choice about how to represent free and bound variables, i.e. the datatype stored in the *leaves* of syntax trees. Encoding strategies go by names like *fully named*, *de Bruijn indices*, *de Bruijn levels*, *locally named*, *locally nameless*, and variations. DTMs capture what is tree-like about syntax

[1] `https://github.com/dunnl/tealeaves`

without saying anything about the type of data in the leaves, and for now we shall remain agnostic about this choice. Figure 1 displays a first-order definition of the set of raw lambda terms. The only unusual part of this definition is that we parameterize the set of terms by a representation of variables V and binder annotations B. These parameters will be fixed by a variable encoding strategy in Section 2.1.

```
Inductive term (B V : Set) : Set :=
  | Var: V -> term B V
  | App: term B V -> term B V -> term B V
  | Lam: B -> term B V -> term B V.
```

Figure 1: Syntax of the lambda calculus in Coq

$$\begin{aligned}
\mathrm{bind}\, f\, (\mathtt{Var}\, v) &= f\, v\\
\mathrm{bind}\, f\, (\mathtt{App}\, t_1\, t_2) &= \mathtt{App}\, (\mathrm{bind}\, f\, t_1)\, (\mathrm{bind}\, f\, t_2)\\
\mathrm{bind}\, f\, (\mathtt{Lam}\, b\, t) &= \mathtt{Lam}\, b\, (\mathrm{bind}\, f\, t)
\end{aligned}$$

Figure 2: bind instance for `term`

To concentrate on `term` as functor in V, we shall typeset B as a subscript. Associated to the lambda calculus is a signature functor

$$\Sigma^\lambda_B X \stackrel{def}{=} X \times X + B \times X$$

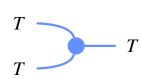
encoding the domain of the two constructors of `term` besides `Var`. $\mathtt{term}_B V$ is defined as the least fixpoint $\mu X. \left(V + \Sigma^\lambda_B X\right)$, i.e. as the smallest solution to the following equation:

$$\mathtt{term}_B V \simeq V + \mathtt{term}_B V \times \mathtt{term}_B V + B \times \mathtt{term}_B V.$$

It is well known that, by its least fixed point construction, a datatype like $\mathtt{term}_B$ (for any B) naturally forms a monad. We present monads string-diagrammatically alongside a conventional equational presentation. A general introduction to string diagrams is outside the scope of this paper, but the interested reader may consult [19, 22]. In this paper our calculus depicts a monad T with a blue wire.

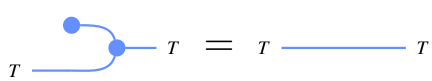
Definition 2.1. *A* monad T *is a functor equipped with two natural transformations*

$\mathrm{ret}^{\mathrm{T}} : \forall (A : \mathbf{Set}), A \to TA$

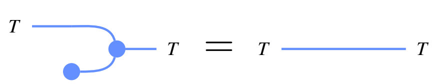


$\mathrm{join}^{\mathrm{T}} : \forall (A : \mathbf{Set}), T(TA) \to TA$

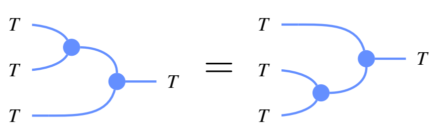
subject to the following laws.



$$\mathrm{join}^{\mathrm{T}} \cdot \mathrm{ret}^{\mathrm{T}} = \mathrm{id} \tag{2.1}$$



$$\mathrm{join}^{\mathrm{T}} \cdot \mathrm{map}^{\mathrm{T}} \left(\mathrm{ret}^{\mathrm{T}}\right) = \mathrm{id} \tag{2.2}$$



$$\mathrm{join}^{\mathrm{T}} \cdot \mathrm{join}^{\mathrm{T}} = \mathrm{join}^{\mathrm{T}} \cdot \mathrm{map}^{\mathrm{T}} \left(\mathrm{join}^{\mathrm{T}}\right) \tag{2.3}$$

The $\mathrm{ret}^{\mathrm{T}}$ operation constructs a tree from a single leaf—for `term` this is the `Var` constructor. $\mathrm{join}^{\mathrm{T}}$ flattens a tree-of-trees into a tree by grafting the layers together. $\mathrm{map}^{\mathrm{T}}$ applies a function to each of the leaves. This presentation is visually pleasing, but fairly abstruse for our purposes. For applications, the following definition is more pragmatic.

Definition 2.2. *A* Kleisli-presented monad $T : \mathbf{Set} \to \mathbf{Set}$ *is a set-forming operation equipped with two polymorphic operations*

$$\begin{aligned}
\mathrm{ret} &: \forall (A : \mathbf{Set}),\ A \to TA\\
\mathrm{bind} &: \forall (A\ B : \mathbf{Set}),\ (A \to TB) \to TA \to TB
\end{aligned}$$

subject to the following three laws (implicitly universally quantified over all relevant variables).

$$\text{bind ret} = \text{id} \qquad (2.4) \qquad\qquad \text{bind } g \cdot \text{bind } f = \text{bind } (\text{bind } g \cdot f) \qquad (2.6)$$

$$\text{bind } f \cdot \text{ret} = f \qquad (2.5)$$

The equivalence of these definitions is well-known [26].

Lemma 2.3 (Manes, 1976)**.** *Definitions 2.1 and 2.2 are equivalent.*

Figure 2 gives the bind instance for `term`. We note that bind $f\ t$ merely applies f to each variable occurrence in t, replacing it with a subterm. We call this simple replacement operation a *naïve* substitution. (2.4) stipulates that replacing all variables with themselves yields the original t. (2.5) is the definition of bind on `Var`. (2.6) governs the composition of multiple substitutions. The limitations of this naïve notion of substitution become apparent when we turn our attention to situations involving both free and bound variables.

2.1 Variable Encodings

We discuss two exemplary techniques for representing variables.

Fully named A fully named approach assigns names, represented as atoms $a \in \mathbb{A}$, to both free and bound variables, hence $V = \mathbb{A}$. Variable-binding constructs are labeled with the names they introduce, so $B = \mathbb{A}$. The set $\mathtt{term}_{\mathbb{A}}\mathbb{A}$ corresponds to the following pen-and-paper syntax of lambda terms:

$$t ::= a|tt|\lambda a.t$$

Consider the main axiom of lambda calculus, the beta conversion rule $(\lambda x.t_1)\,t_2 =_\beta t_1\{t_2/x\}$, where $t_1\{t_2/x\}$ stands for the capture-avoiding substitution of t_2 in place of free occurrences of x in t_1. For instance:

$$(\lambda x.xz)\,\{z/x\} =_\beta \lambda x.xz \quad (\lambda y.xz)\,\{z/x\} =_\beta \lambda y.zz \quad (\lambda z.xz)\,\{z/x\} =_\beta \lambda y.zy$$

In the first case, x occurs bound and is not replaced, while in the second and third cases it occurs free and is replaced with z. In the last case, z also happens to be the name of the distinct entity introduced by the λ, so a naïve substitution would incorrectly result in the term $\lambda z.zz$. Therefore we rename this entity, and all variables bound to it, to a non-conflicting name, say y. Renaming variables like this complicates a fully named representation, and it also complicates the theory of DTMs. Therefore this manuscript focuses on representations that do not require binder renaming, but see future work in Section 6.

Locally nameless The locally nameless strategy represents free variables as atoms, as before, but represents bound variables as de Bruijn indices [13], natural numbers that describe the "distance" from the occurrence to the abstraction that introduced it, indexing from 0. For example, $\lambda x.\lambda y.xyz$ becomes $\lambda\lambda 10z$. Thus V is the (tagged) union $\mathbb{A} + \mathbb{N}$. For clarity, we use `fvar` and `bvar` as the names of the left and right injections (respectively) into V.

Because the representation of a bound variable is canonical, there is no need to give arbitrary names to bound variables, hence no need to rename them to avoid conflicts. Lambda abstractions do not need to be annotated with names either, which we formally represent by annotating them with type $B = \mathbf{1} = \{\star\}$, the singleton. This gives the set $\mathtt{term}_{\mathbf{1}}\,(\mathbb{A} + \mathbb{N})$, corresponding to the following grammar:

$$t ::= a|n|tt|\lambda t$$

A benefit of locally nameless is that substitution of free variables is particularly simple: a variable is free exactly when it is an atom, so it can never be mistaken for a bound variable. Due to this special simplicity,

$$\mathrm{subst}\, x\, u\; (\mathtt{Var}\, v) = \begin{cases} u & \text{if } v = \mathtt{fvar}\, x \\ \mathtt{Var}\, v & \text{else} \end{cases}$$
$$\mathrm{subst}\, x\, u\; (\mathtt{App}\, t_1\, t_2) = \mathtt{App}\; (\mathrm{subst}\, x\, u\, t_1)\; (\mathrm{subst}\, x\, u\, t_2)$$
$$\mathrm{subst}\, x\, u\; (\mathtt{Lam} \star t) = \mathtt{Lam} \star (\mathrm{subst}\, x\, u\, t)$$

(a) Structurally recursive definition

$$\mathrm{subst}\; x\; u\; t = \mathrm{bind}\;\; (\mathrm{subst}_{\mathrm{loc}}\; x\; u)\;\; t$$
$$\mathrm{subst}_{\mathrm{loc}}\; x\; u\; v = \begin{cases} u & \text{if } v = \mathtt{fvar}\; x \\ \mathrm{ret}\; v & \text{else} \end{cases}$$

(b) Definition abstract over a choice of monad

Figure 3: Substitution of atoms in a locally nameless representation

the "correct" notion of substitution for free variables, subst (Figure 3a), happens to be expressible using bind. This operation has the following type, where subst $x\ u\ t$ replaces x in t with u:

$$\mathrm{subst} : \mathbb{A} \to \mathtt{term_1}\,(\mathbb{A}+\mathbb{N}) \to \mathtt{term_1}(\mathbb{A}+\mathbb{N}) \to \mathtt{term_1}\,(\mathbb{A}+\mathbb{N})$$

Figure 3b defines subst in terms of bind and a function $\mathrm{subst}_{\mathrm{loc}}$ that prescribes the "local" effect of substitution on individual occurrences. Decomposing subst like this practical value because $\mathrm{subst}_{\mathrm{loc}}$ does not depend on the particulars of `term`, so this definition is given abstractly over a monad T. This also means we can employ the monad laws to reason about it, exemplified in the following lemma.

Lemma 2.4. *Let T be any monad, let $x \neq y$ be atoms, and let $t[x \mapsto u]$ denote* subst $x\ u\ t$*, defined abstractly in T. Substitution has the following properties:*[2]

$$x[x \mapsto t] = t \quad t[x \mapsto x] = t \quad t[x \mapsto u_1][y \mapsto u_2] = t\big[x \mapsto u_1[y \mapsto u_2]; y \mapsto u_2\big]$$

Lemma 2.4 is easily proven abstractly over T by appealing to equations (2.4)–(2.6). On the other hand, here is a lemma that cannot even be stated, much less proven, abstractly over T:

Lemma 2.5 (fresh-subst)**.** *If an atom x does not occur in t, then $t[x \mapsto u] = t$.*

Lemma 2.5 cannot be formulated abstractly because we lack a mechanism for defining what it means for an atom to *occur* in a term—occurrence is a predicate, and bind does not provide a mechanism for defining predicates. We can of course prove the lemma for `term` in particular by structural recursion, but this is no longer generic over a choice of T and cannot be shared by users formalizing a different syntax. In order to reason about syntax as a container (of occurrences of variables) like this, we define traversable monads in Section 3.2. This definition admits a generic proof of Lemma 2.5.

However, subst is not the main operation of locally nameless. That distinction belongs instead to an operation called *opening*, defined in Figure 4a. This operation is used to define β-reduction, with the β-conversion rule taking the form $(\lambda t)u =_\beta t^u$. Here, t^u stands for the opening of t by u, defined by replacing all indices in t previously bound to the outermost λ with u. (Note that the replaced variables are actually de Bruijn indices rather than free variables, hence this is not a substitution of atoms.) Unlike with atoms, the replaced indices do not have to share a common representation, as the representation of an index bound to the outer lambda depends on how many other abstractions are in scope at the occurrence—both 0 and 1 in $\lambda(0\lambda 1)$ point to the outermost λ, for instance. Therefore open is defined with an auxiliary function that maintains a count of how many binders we have gone under during

[2] Where x is used as a term, it is understood as the atomic term ret $(\mathtt{fvar}\ x)$. In the third equation, the right side mentions the *parallel* substitution that simultaneously replaces all x with $u_1[y \mapsto u_2]$ and y with u_2.

$$\text{open} : \texttt{term}_{\mathbf{1}}(\mathbb{A}+\mathbb{N}) \to \texttt{term}_{\mathbf{1}}(\mathbb{A}+\mathbb{N}) \to \texttt{term}_{\mathbf{1}}(\mathbb{A}+\mathbb{N})$$
$$\text{open}\ u\ t = \text{open}_0\ u\ t$$

$$\text{open}_n\, u\ (\texttt{Var}\, v) = \begin{cases} u & \text{if } v = \texttt{bvar}\, n \\ \texttt{Var}\, v & \text{else} \end{cases}$$
$$\text{open}_n\, u\ (\texttt{App}\, t_1\, t_2) = \texttt{App}\ (\text{open}_n\, u\, t_1)\ (\text{open}_n\, u\, t_2)$$
$$\text{open}_n\, u\ (\texttt{Lam} \star t) = \texttt{Lam} \star \left(\text{open}_{n+1}\, u\, t\right)$$

(a) Opening a lambda term by u

$$\text{LC} : \texttt{term}_{\mathbf{1}}(\mathbb{A}+\mathbb{N}) \to \mathbf{2}$$
$$\text{LC}\ t = \text{LC}_0\ t$$

$$\text{LC}_n\,(\texttt{Var}\ v) = \begin{cases} \bot & \text{if } v = \texttt{bvar}\ m \text{ and } n \leq m \\ \top & \text{else} \end{cases}$$
$$\text{LC}_n\,(\texttt{App}\ t_1\ t_2) = \text{LC}_n\ t_1\ \wedge\ \text{LC}_n\ t_2$$
$$\text{LC}_n\,(\texttt{Lam}\ \star\ t) = \text{LC}_{n+1}\ t$$

(b) Testing for local closure

Figure 4: Operations on locally nameless terms

recursion. In order to define operations that maintain an "accumulator" argument like this, we introduce decorated monads in Section 3.1.

As a final example, some locally nameless terms, e.g. $\lambda(01)$, do not correspond to ordinary lambda terms because they have indices (in this example, the 1) that do not "point" to any abstraction. Therefore one restricts attention to terms that are *locally closed*, defined in Figure 4b. Like open, LC is defined with a helper function that counts the number of binders gone under during recursion. Unlike open, LC computes a boolean ($\mathbf{2} = \{\top, \bot\}$) instead of a term. To define and reason about LC, one must integrate both concepts above to define decorated-traversable functors and DTMs. Σ^{λ}_{B} is an example of a decorated-traversable functor, and $\texttt{term}_B$ is a DTM. As we have shown with Tealeaves [15], this abstraction suffices to prove a large suite of infrastructural lemmas about the operations above.

3 Decorated Traversable Functors

We introduce decorated and traversable monads separately before incorporating both to form DTMs. We present definitions type-theoretically alongside a diagrammatic calculus. For ease of reading, the different sorts of wires in our graphical calculus, which play different roles, are typeset with high-contrast colors.

3.1 Decorations

The category $\mathbf{Dec}_W$ (Definition 3.4) of decorated functors is parameterized by some monoid W, which we take as given. In Tealeaves, W is typically the free monoid `list` B, representing the list of the binders in scope at some occurrence. In brief, decorated functors arise from the elementary fact that any monoid W in **Set** forms a unique *bi*monoid—a coherent combination of a monoid and a comonoid on the same set. The "product-with" embedding,

$$X \mapsto (X \times -) : \mathbf{Set} \to \mathbf{End}_{\text{Set}}$$

is strong monoidal,[3] meaning it preserves monoids, comonoids, and indeed bimonoids, making $(\mathrm{W} \times -)$ a *bimonad*. Decorated functors are precisely the right comodules of this bimonad, which, by adapting

[3] As opposed to merely lax or oplax monoidal, not to be confused with tensorial strength.

a construction from abstract algebra (see Section 4.1 of [7]), form a monoidal category. This means we can consider monoids of decorated functors, or decorated monads. Now we step throw this slowly.

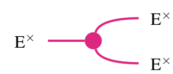
As a first step, consider any set E. It is an exercise in definitions to verify that E is the carrier of exactly one comonoid, the duplication comonoid over E. This structure captures aspects of classical information and its fundamental operations of duplication and deletion.

Definition 3.1. *The* duplication comonoid over E : Set *is given by the following operations.*

$$\begin{array}{rcl} \mathrm{del} & : & E \to \mathbf{1} \\ \Delta & : & E \to E \times E \end{array} \qquad \begin{array}{rcl} \mathrm{del}\, e & = & \star \\ \Delta\, e & = & (e,e) \end{array}$$

The duplication comonoid induces a comonad on $(E \times -)$ known to functional programmers as the environment or reader comonad. In this paper, these wires, which we think of as carrying "contextual" information, are drawn in red.

Definition 3.2. *The* environment comonad over E : Set *is given by the product functor* $(E \times -)$ *equipped with the following operations of* extraction *and* duplication.



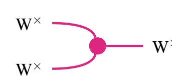
$$\mathrm{extr}^{\mathrm{E}\times} : \forall (A : \mathbf{Set})\,,\ E \times A \to A$$
$$\mathrm{extr}^{\mathrm{E}\times}_A\,(e,a) = a \tag{3.1}$$

$$\mathrm{dup}^{\mathrm{E}\times} : \forall (A : \mathbf{Set})\,,\ E \times A \to E \times (E \times A)$$
$$\mathrm{dup}^{\mathrm{E}\times}_A\,(e,a) = (e,(e,a)) \tag{3.2}$$

The co-Kleisli arrows of the environment comonad have the form $E \times A \to B$. In functional programming, this comonad captures computations $A \to B$ that additionally can read, but not modify, an environment of type E, such as a user-supplied configuration file. This is a classic example of the general intuition that while monads can be used to structure computations with "effects", comonads represent notions of computation that depend on a "context" [37].

Now consider our monoid $W = \langle W, \cdot, 1_W \rangle$. The duplication comonoid exists on the underlying set of W, so in particular $(W \times -)$ is an instance of the reader comonad. Additionally, mirroring the comonoid structure, the monoid on W gives rise to a monad structure on $(W \times -)$ known variously as the writer or logger monad.

Definition 3.3. *The* writer monad *over* W : *Set is given by the product functor* $(W \times -)$ *equipped with the following operations.*

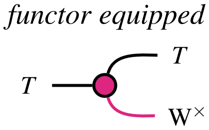


$$\mathrm{ret}^{\mathrm{W}\times} : \forall (A : \mathbf{Set})\,,\ A \to W \times A$$
$$\mathrm{ret}^{\mathrm{W}\times}_A\, a = (1_W, a) \tag{3.3}$$

$$\mathrm{join}^{\mathrm{W}\times} : \forall (A : \mathbf{Set})\,,\ W \times (W \times A) \to W \times A$$
$$\mathrm{join}^{\mathrm{W}\times}_A(w_1,(w_2,a)) = (w_1 \cdot w_2, a) \tag{3.4}$$

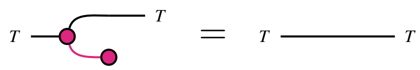
If one thinks about functors as functional data structures, then "decorated" funtors are ones whose elements each occur in a context of type W.

Definition 3.4. *A* decorated functor T : $\mathbf{Set} \to \mathbf{Set}$ *is a right coalgebra of the writer bimonad* $(\mathrm{W} \times -)$. *Explicitly, it is a functor equipped with a natural transformation*

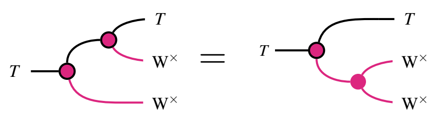


$$\mathrm{dec}^{\mathrm{T}} : \forall (A : \mathbf{Set})\,, TA \to T\,(W \times A)$$

subject to the following two laws:



$$\mathrm{map^T extr^{W\times}} \cdot \mathrm{dec^T} = \mathrm{id} \tag{3.5}$$



$$\mathrm{dec^T} \cdot \mathrm{dec^T} = \mathrm{map^T dup^{W\times}} \cdot \mathrm{dec^T} \tag{3.6}$$

Intuitively, (3.5) states that computing the context of every element and immediately deleting it is the same as doing nothing. (3.6) states that computing each context once and making a copy of it is the same as computing each context twice.

Example 3.5. *The functor Σ^λ_B is decorated by* `list` *B, the free monoid over B. The operation is defined as follows (where by abuse of notation we give constructors of Σ^λ the same name as corresponding constructors of* `term`*):*

$$\mathrm{dec}_X : \Sigma^\lambda_B X \to \Sigma^\lambda_B(\texttt{list}\, B \times X)$$

$$\begin{aligned} \mathrm{dec}\,(\texttt{App}\, x_1\ x_2) &= \texttt{App}\ ([], x_1)\ ([], x_2) \\ \mathrm{dec}\,(\texttt{Lam}\, b\ x) &= \texttt{Lam}\, b\ ([b], x) \end{aligned}$$

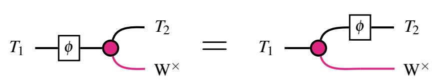
Notation*: $[]$ is the empty list, while $[b]$ is a singleton.*

The decoration in Example 3.5 encodes the policy determining which constructors act as binders in which arguments. The policy states that an abstraction $\lambda b.x$ adds b to the binding context of all occurrences in its body, but applications contribute nothing to the binding context of variables.

Technically, we have not yet used the monoid structure assumed of W. A related fact is that we have only defined decorated *functors*, but our term functor T is a monad. How should these structures be related to each other? The answer comes from the recognition that decorated functors form a monoidal category much like $\mathbf{End}_{\mathrm{Set}}$.

Lemma 3.6. *The category $\mathbf{Dec}_W$ of decorated functors is given by the following data:*

- *Objects are endofuntors $T : \mathbf{Set} \to \mathbf{Set}$ paired with a decoration*
- *Morphisms are natural transformations $T_1 \Rightarrow T_2$ that commute with the decorations of T_1 and T_2*

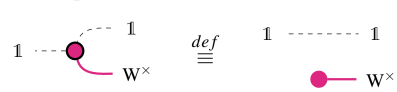


$$\mathrm{dec}^{T_2} \cdot \phi = \phi \cdot \mathrm{dec}^{T_1} \tag{3.7}$$

That this constitutes a category is clear. Slightly less obvious is that $\mathbf{Dec}_W$ is a strict monoidal category. Like $\mathbf{End}_{\mathrm{Set}}$, the tensor operation is composition of functors, but we must explain how to decorate the composition. Likewise, the tensor unit is the identity functor, whose decoration must also be defined.

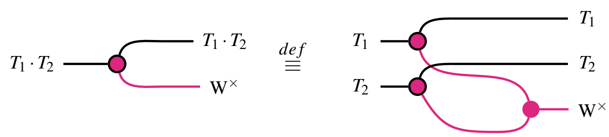
Lemma 3.7. $\mathbf{Dec}_W$ *is a monoidal category by the following data:*

- *The tensor unit is the identity functor paired with the "null" decoration*



$$\mathrm{dec}^{\mathbb{1}} = \mathrm{ret}^{W\times} \tag{3.8}$$

- *Tensor product is given by composition of functors, with decorations added monoidally*

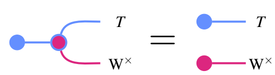


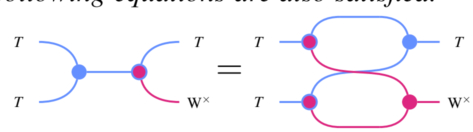
$$\mathrm{dec}^{\mathrm{T}_1 \cdot \mathrm{T}_2} = \mathrm{map}^{\mathrm{T}_1} \left(\mathrm{map}^{\mathrm{T}_2} \left(\mathrm{join}^{\mathrm{W}^\times} \right) \cdot \mathrm{st}^{\mathrm{T}_2}_{\mathrm{W}} \right) \cdot \mathrm{dec}^{\mathrm{T}_1} \cdot \mathrm{map}^{\mathrm{T}_1} \mathrm{dec}^{\mathrm{T}_2} \tag{3.9}$$

Above, $\mathrm{st}^{\mathrm{T}_2}_{\mathrm{W}} : \forall\, (A : \mathbf{Set})\,, W \times T_2\, A \to T_2\, (W \times A)$ is the tensorial strength operation, depicted as crossing a red wire over a functor. That (3.8) and (3.9) satisfy axioms (3.5)—(3.6) is easily verified, as are the laws governing the tensor operation.

Since $\mathbf{Dec}_W$ is a monoidal category, it makes sense to consider monoids in this category. Such a structure must be both an ordinary monad and a decorated functor. The new detail is that the monad operations must also satisfy (3.7), given the operations defined in Lemma 3.7. This yields two additional equations.

Definition 3.8. *A* decorated monad *is a monoid in* $\mathbf{Dec}_W$. *Explicitly, it is equipped with the structures of both a decorated functor and a monad such that the following equations are also satisfied.*





$$\mathrm{dec}^{\mathrm{T}} \cdot \mathrm{ret}^{\mathrm{T}} = \mathrm{ret}^{\mathrm{T}} \cdot \mathrm{ret}^{\mathrm{W}^\times} \tag{3.10}$$

$$\mathrm{dec}^{\mathrm{T}} \cdot \mathrm{join}^{\mathrm{T}} = \mathrm{join}^{\mathrm{T} \cdot \mathrm{W}^\times} \cdot \mathrm{dec}^{\mathrm{T}} \cdot \mathrm{map}^{\mathrm{T}} \left(\mathrm{dec}^{\mathrm{T}} \right) \tag{3.11}$$

In (3.11), $\mathrm{join}^{\mathrm{T} \cdot \mathrm{W}^\times}$ is an abbreviation for

$$\mathrm{join}^{\mathrm{T}} \cdot \mathrm{map}^{\mathrm{T}} \left(\mathrm{map}^{\mathrm{T}} \left(\mathrm{join}^{\mathrm{W}^\times} \right) \cdot \mathrm{st}^{\mathrm{T}}_{\mathrm{W}} \right) : \forall (A : \mathbf{Set}), T\, (W \times T\, (W \times A)) \to T\, (W \times A)$$

Indeed, this operation is part of a monad instance on $T \cdot (W \times -)$.

In the context of syntax metatheory, (3.10) states that an atomic term (some `Var` x) has no binders—the context of x is the monoid unit, typically the empty list or the natural number 0. (3.11) governs how decoration behaves when we compose constructors to form complex syntax trees. It states that the context of each variable instance is the concatenation of the context contributed by each constructor. That is, binders accumulate as one recurses down a syntax tree, as in the recursive operations from Figure 4.

Example 3.9. *The monad* $\mathtt{term}_B$ *is decorated by* $\mathtt{list}\, B$. *The operation annotates each variable with the list of* B *values encountered on the unique path from root of the syntax tree to the variable occurrence. We show examples using fully named and locally nameless variables:*

$$\begin{array}{cc}
\mathrm{dec} : \mathtt{term}_{\mathbb{A}}\, \mathbb{A} \to \mathtt{term}_{\mathbb{A}}\, (\mathtt{list}\, \mathbb{A} \times \mathbb{A}) & \mathrm{dec} : \mathtt{term}_{\mathbf{1}}\, (\mathbb{A} + \mathbb{N}) \to \mathtt{term}_{\mathbf{1}}\, (\mathbb{N} \times (\mathbb{A} + \mathbb{N})) \\
\lambda x. \lambda y. yx \mapsto \lambda x. \lambda y. ([x,y], y)([x,y], x) & \lambda\lambda 01 \mapsto \lambda\lambda(2,0)(2,1) \\
(\lambda x. y \lambda y. z) \mapsto (\lambda x. ([x], y) \lambda y. ([x,y], z)) & (\lambda 0)\, (\lambda\lambda 1) \mapsto (\lambda\, (1,0))\, (\lambda\lambda(2,1))
\end{array}$$

Note that in the locally nameless example we make the implicit identification $\mathtt{list}\, \mathbf{1} \simeq \mathbb{N}$.

The payoff of this definition will be explained after we consider the separate issue of traversability.

3.2 Traversals

Intuitively, a traversable data structure is a finitary container we can "iterate" [21] over, such as a `list` or `tree` type. McBride and Paterson [28] defined traversable functors as those equipped with a distributive law over applicative functors (i.e. lax monoidal endofunctors on **Set**). Subsequent work [21, 23] refined the notion by supplying an appropriate set of axioms for this operation.

Definition 3.10. *An* applicative functor *is a set-forming operation* $F : \mathbf{Set} \to \mathbf{Set}$ *with operations*

$$\mathrm{pure}^{\mathrm{F}} : \forall\,(A : \mathbf{Set})\,,\ A \to FA$$
$$(\circledast)^{\mathrm{F}} : \forall\,(A\,B : \mathbf{Set})\,,\ F\,(A \to B) \to FA \to FB$$

subject to the following equations (note that $\circledast$ *is left-associative).*

$$\mathrm{pure}\ \mathrm{id} \circledast a = a \qquad (3.12)$$
$$\mathrm{pure}\ f \circledast \mathrm{pure}\ a = \mathrm{pure}\ \ (f a) \qquad (3.13)$$
$$g \circledast (f \circledast a) = \mathrm{pure}\ \ (\cdot) \circledast g \circledast f \circledast a \qquad (3.14)$$
$$f \circledast \mathrm{pure}\ a = \mathrm{pure}\ \ (f \mapsto f a) \circledast f \qquad (3.15)$$

This class includes the identity functor $\mathbb{1}$ and is closed under composition. An important special case are constant applicatives: these must map all sets to some monoid M, with the operations and axioms coinciding with those of monoids.

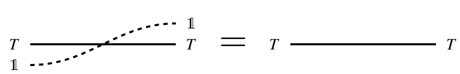
Definition 3.11. *An* applicative morphism $\phi : F \Rightarrow G$ *is a natural transformation between applicative functors that commutes with* pure *and* $(\circledast)$ *in an obvious way.*

Traversable functors are those that distribute over any choice of applicative functor in a well-behaved way.

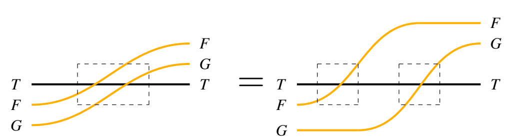
Definition 3.12. *A traversable functor is equipped with an operation*

$$\mathrm{dist}^{\mathrm{T}} : \forall\,(F : \mathrm{Applicative})\,(A : \mathbf{Set})\,, T\,(FA) \to F\,(TA)$$

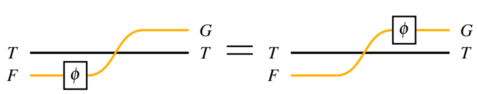
subject to the following axioms (ϕ *ranging over applicative morphisms).*



$$\mathrm{dist}^{\mathrm{T}}_{\mathbb{1}} = \mathrm{id} \qquad (3.16)$$



$$\mathrm{dist}^{\mathrm{T}}_{\mathrm{F \cdot G}} = \mathrm{map}^{\mathrm{F}}\left(\mathrm{dist}^{\mathrm{T}}_{\mathrm{G}}\right) \cdot \mathrm{dist}^{\mathrm{T}}_{\mathrm{F}} \qquad (3.17)$$



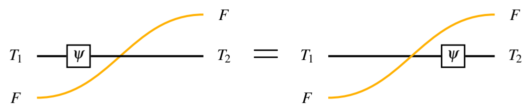
$$\mathrm{dist}^{\mathrm{T}}_{\mathrm{G}} \cdot \mathrm{map}^{\mathrm{T}}\,(\phi_{\mathrm{A}}) = \phi_{\mathrm{A}} \cdot \mathrm{dist}^{\mathrm{T}}_{\mathrm{F}} \qquad (3.18)$$

The connection between traversability and container-like properties is best exemplified by choosing F to be a constant functor over a monoid M. Then, the type of dist reduces to $TM \to M$. Intuitively, T contains a finite number of elements, so that when all elements have type M, we can combine them together using multiplication in M. Gibbons and Oliveira [21] pointed out that (3.16) forbids this operation from "skipping" any elements in T, while Jaskelioff and Rypacek [23] pointed out that (3.17) forbids this operation from "double counting" any elements.

Waern [38] defined the monoidal category of traversable functors. An arrow in this category is a natural transformation between traversable functors that commutes with $\mathrm{dist}_{,}$ in an obvious way.

Lemma 3.13 (Category **Trav**). *The category* **Trav** *of traversable functors is given by the following data:*

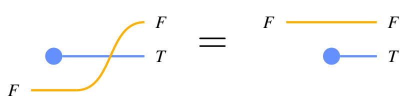
- *Objects are endofunctors T :* **Set** $\to$ **Set** *paired with a distributive law over applicative functors*
- *Morphisms are natural transformations $\psi : T_1 \Rightarrow T_2$ that commute with distribution.*



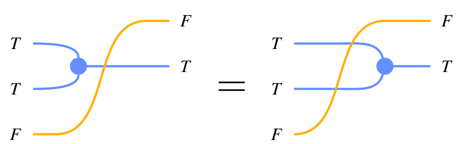
$$\mathrm{dist}_{\mathrm{F}}^{\mathrm{T_2}} \cdot \psi = \mathrm{map}^{\mathrm{F}}(\psi) \cdot \mathrm{dist}_{\mathrm{F}}^{\mathrm{T_1}} \tag{3.19}$$

The identity functor is trivially traversable, and the composition of traversables is traversable just by composing the distributions. Hence, traversable functors **Trav** forms a monoidal category. As before, we can consider monoids in this category. These are monads that are also traversable and whose monad operations satisfy (3.19).

Definition 3.14. *A* traversable monad *T is a monoid in* **Trav**. *Explicitly, T has the structures of both a traversable functor and a monad and satisfies the following equations:*



$$\mathrm{dist}_{\mathrm{F}}^{\mathrm{T}} \cdot \mathrm{ret}^{\mathrm{T}} = \mathrm{map}^{\mathrm{F}}\left(\mathrm{ret}^{\mathrm{T}}\right) \tag{3.20}$$



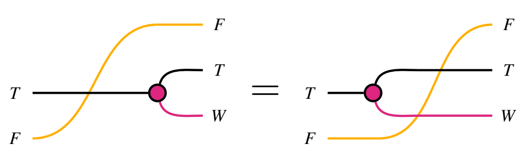
$$\mathrm{dist}_{\mathrm{F}}^{\mathrm{T}} \cdot \mathrm{join}^{\mathrm{T}} = \mathrm{map}^{\mathrm{F}}\left(\mathrm{join}^{\mathrm{T}}\right) \cdot \mathrm{dist}_{\mathrm{F}}^{\mathrm{T}} \cdot \mathrm{map}^{\mathrm{T}}\left(\mathrm{dist}_{\mathrm{F}}^{\mathrm{T}}\right) \tag{3.21}$$

Though the laws appear opaque, for syntax metatheory, (3.20) states that a term formed from ret/`Var` contains only a single variable. (3.21) implies that substituting a subterm u for x in t adds the occurrences in u to the set of occurrences of t. This concept is more thoroughly examined in [15].

3.3 Decorated Traversable Functors

For functors that are both traversable and decorated, it is necessary to impose one more condition relating the decoration and distribution operations. For the following definition, we note that $(\mathrm{W} \times -)$ is uniquely traversable.

Definition 3.15. *A* decorated-traversable *functor is equipped with the structure of both a decorated and traversable functor (Definitions 3.4 and 3.12), subject to the following extra condition:*



$$\mathrm{map}^{\mathrm{F}}\left(\mathrm{dec}^{\mathrm{T}}\right) \cdot \mathrm{dist}_{\mathrm{F}}^{\mathrm{T}} = \mathrm{dist}_{\mathrm{F}}^{\mathrm{T}} \cdot \mathrm{map}^{\mathrm{T}}\left(\mathrm{dist}_{\mathrm{F}}^{\mathrm{W}\times}\right) \cdot \mathrm{dec}^{\mathrm{T}} \tag{3.22}$$

Lemma 3.16 (Category $\mathbf{DecTrav}_W$)**.** *The strict monoidal category* $\mathbf{DecTrav}_W$ *of decorated-traversable functors is given by the following data:*

- *Objects are decorated traversable functors*
- *Morphisms are natural transformations satisfying both* (3.7) *and* (3.19).
- *The tensor product is given by composition of decorated-traversable functors, with the identity functor serving as the tensor unit.*

Definition 3.17. *A* decorated traversable monad *(DTM) is a monoid in* $\mathbf{DecTrav}_W$.

The force of Definition 3.17 is that a DTM is simultaneously an instance of Definitions 3.8, 3.14, and 3.16. A self-contained summary of the axioms can be found in the appendix.

4 Kleisli Representation for DTMs

Definition 3.17 is phrased in terms of principled categorical abstractions, but this is not the most convenient presentation when working in a theorem prover. Just proving that a syntax forms a DTM is tedious, requiring five operations and 19 equations. The following Kleisli-style definition, mirroring Definition 2.2, is more economical and more useful to program with.

Definition 4.1 (DTMs, Kleisli-style)**.** *A* Kleisli-presented DTM *is a set-forming operation T equipped with two operations of the following types*

$$\begin{array}{rcl} \mathrm{ret} & : & \forall (A : \mathbf{Set}),\ A \to TA \\ \mathrm{binddt} & : & \forall (F : \mathrm{Applicative})\,(A : \mathbf{Set}),\ (W \times A \to F(TB)) \to TA \to F\,(TB) \end{array}$$

subject to the following laws (where ϕ is quantified over applicative morphisms $\phi : F \Rightarrow G$)

$$\mathrm{binddt}_{\mathbb{1}}\ \left(\mathrm{ret}^{\mathrm{T}} \cdot \mathrm{extr}^{\mathrm{W}\times}\right) = \mathrm{id} \tag{4.1}$$

$$\mathrm{binddt}_{\mathrm{F}}\, f \cdot \mathrm{ret}^{\mathrm{T}} = f \cdot \mathrm{ret}^{\mathrm{W}\times} \tag{4.2}$$

$$\mathrm{map}^{\mathrm{F}}\ (\mathrm{binddt}_{\mathrm{G}}\ g) \cdot (\mathrm{binddt}_{\mathrm{F}}\ f) = \mathrm{binddt}_{\mathrm{F}\cdot\mathrm{G}}\ \left(\lambda(w,a).\mathrm{map}^{\mathrm{F}}\ (\mathrm{binddt}_{\mathrm{G}}\ (g \odot w))\, f(w,a)\right) \tag{4.3}$$

$$\phi \cdot \mathrm{binddt}_{\mathrm{F}}\, f = \mathrm{binddt}_{\mathrm{G}}\,(\phi \cdot f) \tag{4.4}$$

In (4.3), $(\odot)$ *is defined* $(g \odot w_1)\,(w_2, b) \stackrel{def}{=} g\ (w_1 \cdot w_2, b)$.

The following theorem speaks to the robustness of Definition 4.1.

Theorem 4.2. *Definitions 3.17 and 4.1 are equivalent.*

Proof. The ret operation is the same for both presentations. Given map, join, dec, and dist, we define binddt as follows:

$$\mathrm{binddt}_{\mathrm{F}} f \stackrel{def}{=} \mathrm{map}^{\mathrm{F}}\left(\mathrm{join}^{\mathrm{T}}\right) \cdot \mathrm{dist}^{\mathrm{T}}_{\mathrm{F}} \cdot \mathrm{map}^{\mathrm{T}}\ f \cdot \mathrm{dec}. \tag{4.5}$$

Given ret and binddt, we define the operations of DTMs thus:

$$\begin{array}{rcl@{\qquad}rcl} \mathrm{map} f & \stackrel{def}{=} & \mathrm{binddt}_{\mathbb{1}}\left(\mathrm{ret}^{\mathrm{T}} \cdot f \cdot \mathrm{extr}^{\mathrm{W}\times}\right) & \mathrm{dec} & \stackrel{def}{=} & \mathrm{binddt}_{\mathbb{1}}\left(\mathrm{ret}^{\mathrm{T}}\right) \\ \mathrm{join} & \stackrel{def}{=} & \mathrm{binddt}_{\mathbb{1}}\left(\mathrm{extr}^{\mathrm{W}\times}\right) & \mathrm{dist} & \stackrel{def}{=} & \mathrm{binddt}_{\mathrm{F}}\left(\mathrm{ret}^{\mathrm{T}} \cdot \mathrm{extr}^{\mathrm{W}\times}\right) \end{array}$$

Besides verifying these definitions satisfy the appropriate equations, that starting with either representation and completing a roundtrip returns the original set of operations. A full proof of this fact can be found in our GitHub repository. The appendix contains a string-diagrammatic derivation of (4.1)—(4.4) (Lemma A.1). ☐

Example 4.3. *The* binddt *operation for* $\texttt{term}_B$ *is defined as follows (for any* $f : W \times A \to F(TB)$*):*

$$\begin{array}{lcl} \mathrm{binddt_F}\ f\ (\texttt{Var}\ v) & = & f([],v) \\ \mathrm{binddt_F}\ f\ (\texttt{App}\ t_1\ t_2) & = & \mathrm{pure^F}\ \texttt{App} \circledast \mathrm{binddt_F}\ f\ t_1 \circledast \mathrm{binddt_F}\ f\ t_2 \\ \mathrm{binddt_F}\ f\ (\texttt{Lam}\ b\ t) & = & \mathrm{pure^F}\ (\texttt{Lam}\ b) \circledast \mathrm{binddt_F}\ (f \odot [b])\ t \end{array}$$

Like bind, binddt can be seen as a template for defining structurally recursive operations on abstract syntax trees. However, it is appreciably more expressive, introducing two new features. First, the first argument of f is now a list of binders in scope at each variable. Second, the output of f is wrapped in an applicative functor, and all function application is replaced with "idiomatic" application ($\circledast$). Incorporating these aspects greatly expands the range of operations we can define generically.

4.1 Substitution Metatheory

Figure 6 contains generic versions of the opening operation and local closure, relating to Figure 4 as Figure 3b does to 3a. The definition of LC in particular requires full use of the expressiveness of $\mathrm{binddt}_,$. Here, **2** stands for the constant applicative functor over the monoid $\langle \mathbf{2}, \wedge, \top \rangle$, which provides a form of universal quantification over variables. As instances of binddt, we can reason about these operations axiomatically.

$$\mathrm{open_{loc}} : T\,(\mathbb{A}+\mathbb{N}) \to \mathbb{N}\times(\mathbb{A}+\mathbb{N}) \to \mathtt{T}\,(\mathbb{A}+\mathbb{N})$$
$$\mathrm{open_{loc}}\,u\;(n,\texttt{fvar}\,a) = \mathrm{ret}\,(\texttt{fvar}\,a)$$
$$\mathrm{open_{loc}}\,u\;(n,\texttt{bvar}\,m) = \begin{cases} u & \text{if } n=m \\ \mathrm{ret^T}\,(\texttt{bvar}\,\mathrm{m}) & \text{else} \end{cases}$$
$$\mathrm{open}\,u = \mathrm{binddt}^{\mathrm{T}}_{\mathbb{1}}\;(\mathrm{open_{loc}}\,u)$$

$$\mathrm{LC_{loc}} : \mathbb{N}\times(\mathbb{A}+\mathbb{N}) \to \mathbf{2}$$
$$\mathrm{LC_{loc}}\,(n,\texttt{fvar}\,a) = \top$$
$$\mathrm{LC_{loc}}\,(n,\texttt{bvar}\,m) = \begin{cases} \bot & \text{if } n \le m \\ \top & \text{else} \end{cases}$$
$$\mathrm{LC} = \mathrm{binddt}^{\mathrm{T}}_{\mathbf{2}}\;\mathrm{LC_{loc}}$$

Figure 6: Generic locally nameless operations for a DTM T

The adequacy of Definition 4.1 for the needs of working metatheorists is an empirical question demonstrated by formalizing generic syntax metatheory with it. For comparison, Weirich and Aydemir previously introduced LNgen [5], a code generator that accepts a grammar and synthesizes files containing locally nameless infrastructure for it in Coq. Using Tealeaves, we were able to formalize all of the infrastructure lemmas defined in [5], as well as others, statically and generically over a choice of arbitrary DTM. We have not found any lemmas of the locally nameless representation that we cannot prove in this fashion. The advantage of Tealeaves over LNgen is that our lemmas are proven once and for all, while LNgen generates proofs specific to a given signature. Because it relies on heuristics and Ltac [14] (Coq's incompletely specified proof automation language), the authors have reported in private correspondence that LNgen can fail to prove some lemmas. Additionally they have reported long compile times which must be re-endured after any changes to the user's syntax. These downsides do not apply to Tealeaves because it is a static Coq library rather than a program. The cost of entry is to furnish a proof of (4.1)–(4.4), which we hope to automate in future work.

We have also developed a generalization of DTMs for languages with multiple sorts of variables, and re-derived the same locally nameless infrastructure, now extended to reason about operations affecting different sorts of variables.

5 Related Work

Bellegarde and Hook [8] first considered term monads in the context of formal metatheory. They defined substitution for a de Bruijn encoding in terms of a combinator `Ewp` ("extend with policy") which is similar in spirit to, but strictly less expressive than, $\text{binddt}_{,}$. Lacking axioms comparable to (4.1)–(4.4), they were unable to reason about substitution generically.

Subsequent work has generally considered intrinsically well-scoped [4] and well-typed [10, 27, 3] representations using heterogeneous datatypes [9]. Leveraging the metatheory's type system to constrain object terms will tend to lead to a more dependently-typed style of programming where operations and their correctness properties are woven together. Building on this line of work, Ahrens et al. [2] have recently proposed an intrinisically typed language formalization framework in Coq. The goal of Tealeaves is to support raw syntax, which involves defining operations first and reasoning about them post factum.

Fiore and collaborators [16, 17] have developed a presheaf-theoretic account of syntax. Subsequent work by Power and Tanaka axiomatized and expanded the presheaf-theoretic approach [31, 32]. The basic idea is that intrinsically scoped terms are stratified by a context—the set of all contexts is then used as the indexing category for the presheaves. In our development, syntax is parameterized by types V and B for representations of variables and binder annotations. These are fixed by a particular representation strategy (e.g. locally nameless) and one is left with a single set of terms rather than a presheaf. Fiore and Szamozvancev have proposed a intrinsically well-scoped, well-typed, syntax formalization framework in Agda [18] which takes inspiration from the presheaf approach.

Approaches that differ more dramatically from ours include strategies based on nominal sets [20] and variations of higher-order abstract syntax [30, 12].

Besides LNgen, utilities similar in spirit to Tealeaves include GMeta [25] and Autosubst [34, 35]. GMeta is a Coq framework for generic raw, first-order syntax. Like Tealeaves, it is parameterized by a variable encoding strategy. GMeta resorts to proofs by induction on a universe of representable types, while Tealeaves is based on a principled equational theory. Autosubst is an equational framework for reasoning about de Bruijn indices in Coq based on explicit substitution calculi [1, 33]. Our $\text{binddt}_{,}$ can express de Bruijn substitution; it may be enlightening to consider DTMs vis-à-vis these calculi.

6 Conclusion and Future Work

We have presented decorated traversable monads, an enrichment of monads on the category of sets that can be used to reason equationally about raw, first-order representations of variable binding.

As presented, DTMs are not equipped with a binder-renaming operation necessary to implement a fully named binding strategy. A first step in this direction is to recognize that `term` is also a functor in B besides V, yielding an operation

$$\text{bmap} : \forall (V \ \mathrm{B}_1 \ \mathrm{B}_2 : \mathbf{Set}), (B_1 \to B_2) \to \mathtt{term}_{B_1} V \to \mathtt{term}_{B_2} V$$

We are investigating an extension of DTMs that incorporates the functor instance in B. One intended application is to provide a certified generic translation between a named and locally nameless representation, which could be used as part of a certified compiler, for example.

Imposing a distributive law over all applicative functors imposes an order on variable occurrences, which may be unnecessarily strong. Some process calculi, for example, feature a notion of parallel composition $|$ such that formulas $p_1|p_2$ and $p_2|p_1$ should be taken as syntactically identical. To support quotiented syntax, one might require a distributive law only over commutative applicative functors.

References

[1] M. Abadi, L. Cardelli, P.-L. Curien & J.-J. Lévy (1991): *Explicit substitutions*. Journal of Functional Programming 1(4), p. 375–416, doi:10.1017/S0956796800000186.

[2] Benedikt Ahrens, Ralph Matthes & Anders Mörtberg (2022): *Implementing a Category-Theoretic Framework for Typed Abstract Syntax*. In: Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2022, Association for Computing Machinery, New York, NY, USA, p. 307–323, doi:10.1145/3497775.3503678.

[3] Guillaume Allais, James Chapman, Conor McBride & James McKinna (2017): *Type-and-Scope Safe Programs and Their Proofs*. In: Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Association for Computing Machinery, New York, NY, USA, p. 195–207, doi:10.1145/3018610.3018613.

[4] Thorsten Altenkirch & Bernhard Reus (1999): *Monadic Presentations of Lambda Terms Using Generalized Inductive Types*. In: Proceedings of the 13th International Workshop and 8th Annual Conference of the EACSL on Computer Science Logic, CSL '99, Springer-Verlag, Berlin, Heidelberg, p. 453–468, doi:10.1007/3-540-48168-0_32.

[5] Brian Aydemir & Stephanie Weirich (2010): *LNgen: Tool Support for Locally Nameless Representations*. Technical Report, University of Pennsylvania, Department of Computer and Information Science. Available at https://repository.upenn.edu/handle/20.500.14332/7902.

[6] Brian E. Aydemir, Aaron Bohannon, Matthew Fairbairn, J. Nathan Foster, Benjamin C. Pierce, Peter Sewell, Dimitrios Vytiniotis, Geoffrey Washburn, Stephanie Weirich & Steve Zdancewic (2005): *Mechanized Metatheory for the Masses: The PoplMark Challenge*. In: Proceedings of the 18th International Conference on Theorem Proving in Higher Order Logics, TPHOLs'05, Springer-Verlag, Berlin, Heidelberg, p. 50–65, doi:10.1007/11541868_4.

[7] Tørris Koløen Bakke (2007): *Hopf Algebras and Monoidal Categories*. Master's thesis, University of Tromsø. Available at https://munin.uit.no/bitstream/handle/10037/1084/finalthesis.pdf.

[8] Françoise Bellegarde & James Hook (1994): *Substitution: A Formal Methods Case Study Using Monads and Transformations*. Sci. Comput. Program. 23(2–3), p. 287–311, doi:10.1016/0167-6423(94)00022-0.

[9] Richard Bird & Lambert Meertens (1998): *Nested datatypes*. In: In MPC'98, volume 1422 of LNCS, Springer-Verlag, pp. 52–67, doi:10.1007/BFb0054285.

[10] Richard Bird & Ross Paterson (1999): *de Bruijn notation as a nested datatype*. Journal of Functional Programming 9, pp. 77 – 91, doi:10.1017/S0956796899003366.

[11] R. M. Burstall (1969): *Proving Properties of Programs by Structural Induction*. The Computer Journal 12(1), pp. 41–48, doi:10.1093/comjnl/12.1.41.

[12] Adam Chlipala (2008): *Parametric Higher-Order Abstract Syntax for Mechanized Semantics*. In: Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming, ICFP '08, Association for Computing Machinery, New York, NY, USA, p. 143–156, doi:10.1145/1411204.1411226.

[13] N.G de Bruijn (1972): *Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem*. Indagationes Mathematicae (Proceedings) 75(5), pp. 381–392, doi:10.1016/1385-7258(72)90034-0.

[14] David Delahaye (2000): *A Tactic Language for the System Coq*. In Michel Parigot & Andrei Voronkov, editors: Logic for Programming and Automated Reasoning, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 85–95, doi:10.1007/3-540-44404-1_7.

[15] Lawrence Dunn, Val Tannen & Steve Zdancewic (2023): *Tealeaves: Structured Monads for Generic First-Order Abstract Syntax Infrastructure*. In Adam Naumowicz & René Thiemann, editors: 14th International Conference on Interactive Theorem Proving (ITP 2023), Leibniz International Proceedings in Informatics (LIPIcs) 268, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 14:1–14:20, doi:10.4230/LIPIcs.ITP.2023.14.

[16] M. Fiore, G. Plotkin & D. Turi (1999): *Abstract syntax and variable binding*. In: *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*, pp. 193–202, doi:10.1109/LICS.1999.782615.

[17] Marcelo Fiore (2008): *Second-Order and Dependently-Sorted Abstract Syntax*. In: *Proceedings of the 2008 23rd Annual IEEE Symposium on Logic in Computer Science*, LICS '08, IEEE Computer Society, USA, p. 57–68, doi:10.1109/LICS.2008.38.

[18] Marcelo Fiore & Dmitrij Szamozvancev (2022): *Formal Metatheory of Second-Order Abstract Syntax*. *Proc. ACM Program. Lang.* 6(POPL), doi:10.1145/3498715.

[19] Brendan Fong & David I Spivak (2018): *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*. doi:10.48550/arXiv.1803.05316.

[20] Murdoch J. Gabbay & Andrew M. Pitts (2002): *A New Approach to Abstract Syntax with Variable Binding*. *Form. Asp. Comput.* 13(3–5), p. 341–363, doi:10.1007/s001650200016.

[21] Jeremy Gibbons & Bruno Oliveira (2009): *The essence of the Iterator pattern*. *J. Funct. Program.* 19, pp. 377–402, doi:10.1017/S0956796809007291.

[22] Ralf Hinze & Dan Marsden (2023): *Introducing String Diagrams: The Art of Category Theory*. Cambridge University Press, doi:10.1017/9781009317825.

[23] Mauro Jaskelioff & Ondrej Rypacek (2012): *An Investigation of the Laws of Traversals*. *Electronic Proceedings in Theoretical Computer Science* 76, doi:10.4204/EPTCS.76.5.

[24] Ramana Kumar, Magnus O. Myreen, Michael Norrish & Scott Owens (2014): *CakeML: a verified implementation of ML*. In: *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pp. 179–192, doi:10.1145/2535838.2535841.

[25] Gyesik Lee, Bruno C. D. S. Oliveira, Sungkeun Cho & Kwangkeun Yi (2012): *GMeta: A Generic Formal Metatheory Framework for First-Order Representations*. In Helmut Seidl, editor: *Programming Languages and Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 436–455, doi:10.1007/978-3-642-28869-2_22.

[26] Ernest G. Manes (1976): *Algebraic Theories*. Springer New York, doi:10.1007/978-1-4612-9860-1.

[27] Conor McBride (2005): *Type-Preserving Renaming and Substitution*. Unpublished note.

[28] Conor McBride & Ross Paterson (2008): *Applicative Programming with Effects*. *J. Funct. Program.* 18(1), p. 1–13, doi:10.1017/S0956796807006326.

[29] Eugenio Moggi (1988): *Computational Lambda-Calculus and Monads*. IEEE Computer Society Press, pp. 14–23, doi:10.1109/LICS.1989.39155.

[30] Frank Pfenning & Conal Elliott (1988): *Higher-Order Abstract Syntax*. 23, pp. 199–208, doi:10.1145/960116.54010.

[31] John Power (2003): *A Unified Category Theoretic Approach to Variable Binding*. In: *Proceedings of the 2003 ACM SIGPLAN Workshop on Mechanized Reasoning about Languages with Variable Binding*, MERLIN '03, Association for Computing Machinery, New York, NY, USA, p. 1–9, doi:10.1145/976571.976578.

[32] John Power & Miki Tanaka (2008): *Category Theoretic Semantics for Typed Binding Signatures with Recursion*. *Fundam. Informaticae* 84, pp. 221–240. Available at http://content.iospress.com/articles/fundamenta-informaticae/fi84-2-05.

[33] Steven Schäfer, Gert Smolka & Tobias Tebbi (2015): *Completeness and Decidability of de Bruijn Substitution Algebra in Coq*. In: *Proceedings of the 2015 Conference on Certified Programs and Proofs*, CPP '15, Association for Computing Machinery, New York, NY, USA, p. 67–73, doi:10.1145/2676724.2693163.

[34] Steven Schäfer, Tobias Tebbi & Gert Smolka (2015): *Autosubst: Reasoning with de Bruijn Terms and Parallel Substitutions*. In Xingyuan Zhang & Christian Urban, editors: *Interactive Theorem Proving - 6th International Conference, ITP 2015, Nanjing, China, August 24-27, 2015*, LNAI, Springer-Verlag, doi:10.1007/978-3-319-22102-1_24.

[35] Kathrin Stark, Steven Schäfer & Jonas Kaiser (2019): *Autosubst 2: Reasoning with Multi-Sorted de Bruijn Terms and Vector Substitutions*. In: *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2019, Association for Computing Machinery, New York, NY, USA, p. 166–180, doi:`10.1145/3293880.3294101`.

[36] The Coq Development Team (2023): *The Coq Proof Assistant*, doi:`10.5281/zenodo.8161141`.

[37] Tarmo Uustalu & Varmo Vene (2008): *Comonadic Notions of Computation*. *Electronic Notes in Theoretical Computer Science* 203(5), pp. 263–284, doi:`10.1016/j.entcs.2008.05.029`. Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science (CMCS 2008).

[38] Love Waern (2019): *Cofree Traversable Functors*. Bachelor's thesis. Available at `https://uu.diva-portal.org/smash/record.jsf?pid=diva2:1369180`.

[39] Jianzhou Zhao, Santosh Nagarakatte, Milo M.K. Martin & Steve Zdancewic (2012): *Formalizing the LLVM Intermediate Representation for Verified Program Transformations*. In: *Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '12, Association for Computing Machinery, New York, NY, USA, p. 427–440, doi:`10.1145/2103656.2103709`.

A Appendix

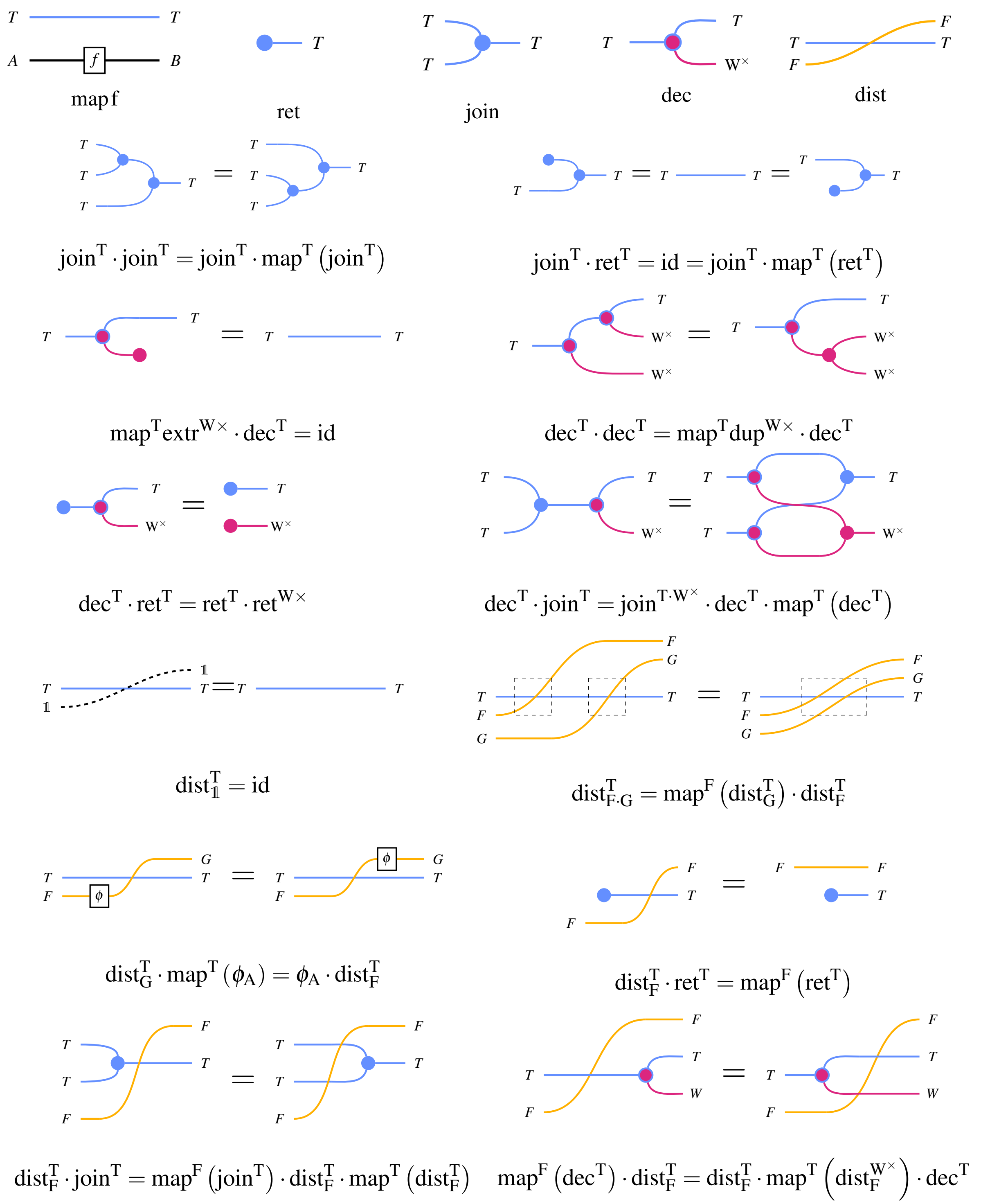
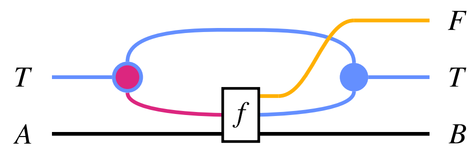


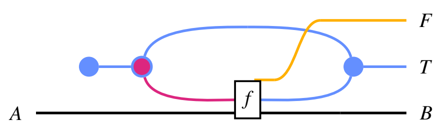
Figure 7: String diagrammatic presentation of DTMs

Lemma A.1. *Every DTM gives rise to a Kleisli-presented DTM according to the following definition of* binddt.

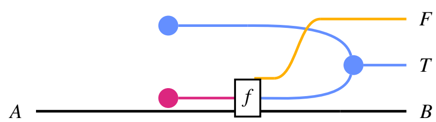


$$\mathrm{binddt}_{\mathrm{F}}\, f = \mathrm{map}^{\mathrm{F}}\left(\mathrm{join}^{\mathrm{T}}\right) \cdot \mathrm{dist}^{\mathrm{T}}_{\mathrm{F}} \cdot \mathrm{map}^{\mathrm{T}}\, f \cdot \mathrm{dec}^{\mathrm{T}}$$

Proof. Proof of Equation (4.1):

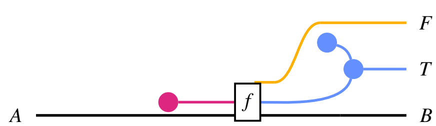


=



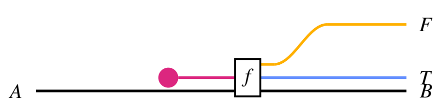
Apply the decoration cup law (3.10).

=



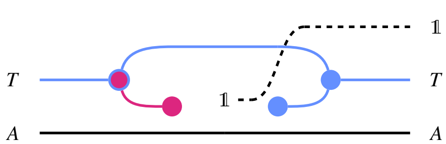
Pull the unit across F (3.20).

=

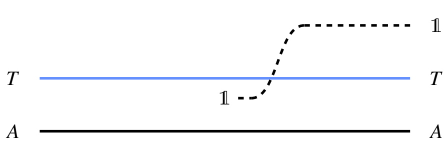


Apply the left monad unit law (2.1).

Proof of Equation (4.2):

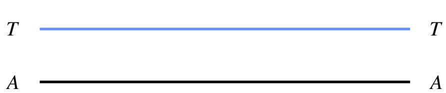


=



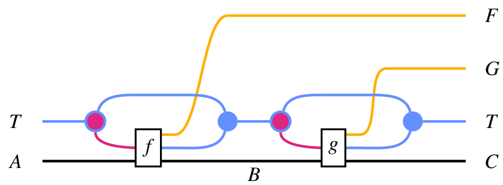
Apply unit and counit laws (3.5) (2.2).

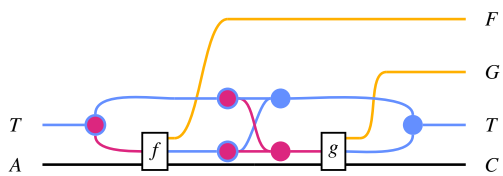
=



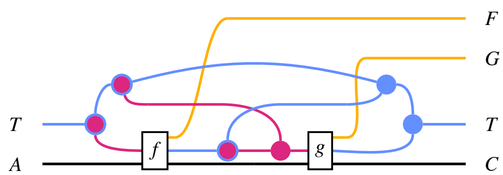
Apply traversal unitary law (3.16).

Proof of Equation (4.3):

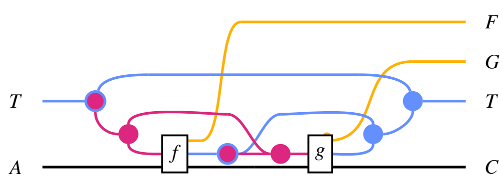


=


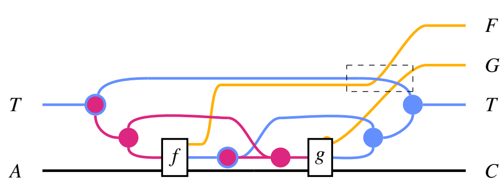
Apply the butterfly law (3.11).

=


Drag operations past distributions (3.21) (3.22).

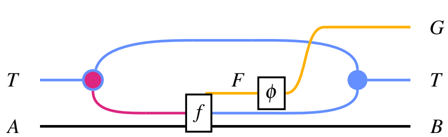
=


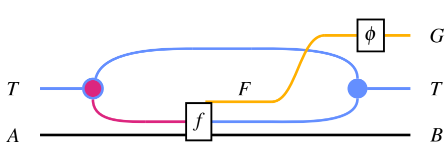
Apply (co)associativity (2.3) (3.6).

=


Apply traversal composition law (3.17).

Proof of Equation (4.4):



=


Slide the applicative morphism (3.18).

□

A Categorical Model for Classical and Quantum Block Designs

Paulina L. A. Goedicke
Institute for Theoretical Physics
University of Cologne
goedicke@thp.uni-koeln.de

Jamie Vicary
Department of Computer Science
University of Cambridge
jamie.vicary@cl.cam.ac.uk

Classical block designs are important combinatorial structures with a wide range of applications in Computer Science and Statistics. Here we give a new abstract description of block designs based on the arrow category construction. We show that models of this structure in the category of matrices and natural numbers recover the traditional classical combinatorial objects, while models in the category of completely positive maps yield a new definition of quantum designs. We show that this generalizes both a previous notion of quantum designs given by Zauner and the traditional description of block designs. Furthermore, we demonstrate that there exists a functor which relates every categorical block design to a quantum one.

1 Introduction

Combinatorial design theory has a variety of applications in computer science and statistics. Its main focus lies on finite discrete structures, such as (hyper)graphs, finite projective and affine planes, block designs, orthogonal arrays and Latin squares that fulfil certain constraints on the arrangement of elements [15]. With the development of quantum computing, it is natural to ask how combinatorial objects fit into quantum information theory. Many interesting intersections between these fields are already known; for example, it has been shown that the problem of constructing mutually orthogonal Latin squares of order d is equivalent to constructing a 2-uniform state of N qudits of d levels having d^2 positive terms [5].

Some combinatorial objects have an analogous quantum form, which can lead to rich insights. For instance, Vicary et al. have introduced the notion of *quantum Latin squares* (QLS), a quantum analogue of Latin squares with which one can build a new construction scheme for unitary error bases [10]. Based on this, Goyeneche et al. introduced *quantum orthogonal arrays* and showed that they are related to QLSs, in the same way that orthogonal arrays are related to Latin squares [6], [4]. Quantum Latin squares also play a role in the classification of other biunitary constructions such as Hadamard matrices [12], quantum teleportation and error correction [10], and also give a new construction scheme for mutually unbiased bases (MUBs) as shown by Musto [9]. Only recently, Życzkowski et al. have found a quantum solution to the famous Euler's problem of thirty six officers, i.e. classically no two orthogonal Latin squares of order six exist, by constructing two orthogonal QLS of order six [11]. Other examples of the connection between classical and quantum combinatorics include the work by Wocjan and Beth on mutually unbiased basis construction from orthogonal Latin squares [18], and Wooter's description of affine plane constructions of mutually unbiased bases [19]. However, a unified perspective on the relationship between classical and quantum combinatorics remains elusive.

A prominent example of combinatorial designs are *balanced incomplete block designs* (BIBDs) [15], combinatorial designs with special balance constraints on the arrangement of elements, and Zauner has extended this to give a more general notion of quantum design [20]. In this paper we develop a category-theoretical model for both classical and quantum designs, using the language of arrow categories. We

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 118–136, doi:10.4204/EPTCS.397.8

will start by reviewing classical design theory and Zauner's notion of quantum designs. We then develop a categorical framework based on arrow categories, that transfers the intrinsic properties of block designs to a pointed monoidal dagger category. Applied to the categories $\mathbf{Mat}(\mathbb{N})$ and $\mathbf{CP}[\mathbf{FHilb}]$, this framework yields a categorical model for both block designs and quantum designs. This not only leads to a more general description of both classical and quantum designs via completely positive maps, but also allows us to relate classical to quantum designs via a functor. Moreover, we will use these techniques to define a category of mutually-unbiased bases.

1.1 Structure of the paper

This paper is organised as follows. In Section 2 and Section 3 we will summarise the central mathematical concepts that are used in the paper, covering elementary combinatorics, quantum designs as they were defined by Zauner, and elements from category theory. Moreover, we give the definition of a category of BIBDs, namely **Block** and a category of general designs, namely **Design**. In Section 4 we will develop the *design construction* that describes the design structures discussed in Section 2.1 via an abstract operation on a pointed monoidal dagger category. By applying this construction to $\mathbf{Mat}(\mathbb{N})$, we then show in the beginning of Section 5 that this recovers the category theoretical description of classical designs, namely $\mathbf{Design}[\mathbf{Mat}(\mathbb{N})]$, and we show that there exists a functor from the subcategory $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N})]$ to **Block**. Furthermore, we define a category **QDesign** of quantum designs by applying the design construction to $\mathbf{CP}[\mathbf{FHilb}]$, and show that it contains a subcategory where objects are uniform and regular quantum designs of degree 1. This also allows us to define a category of MUBs. Finally, we construct a functor between $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ and **QDesign**, yielding a relation between categorical block designs and quantum design in accordance with earlier results given by Zauner [20].

Throughout the paper we will only consider finite dimensional Hilbert spaces.

1.2 Open questions

It would be interesting to know if the more general quantum designs that we define have an application in quantum computing, and in particular if the trace-preserving examples are interesting to use as quantum channels. The same holds for the CP-maps representing classical designs, which can be interpreted in terms of statistical mechanics.[1] Moreover, a category theoretical perspective might lead to new insights on the problem of finding the maximal number of MUBs in arbitrary dimension, a problem which has already been approached by Musto using categorical techniques [9]. Especially in this context, it would be relevant to know how one can embed Latin squares into our model, which may also give new insights on how QLS relate to quantum designs. Finally, one could ask how one can embed mutually unbiased measurements (MUMs) into this framework.

1.3 Acknowledgements.

The first author has been supported by Germany's Excellence Strategy Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 390534769. The second author acknowledges support from the Royal Society. We are grateful to the reviewers for their excellent and insightful comments which improved our paper considerably.

[1]This is because they are morphisms of $\mathbf{CP_c}[\mathbf{FHilb}]$ (see Section 3).

2 Design Theory

In this section we recall the definition of balanced incomplete block designs, following Stinson [15] and Buekenhout [1]. We then review Zauner's definition of quantum designs [20].

2.1 Classical designs

We begin with the definition of designs.

Definition 2.1. A *design* is given by a set $V = \{1,..,v\}$ of *points*, and a set $B = \{1,\dots,b\}$ of *blocks* and an *incidence relation* I between them.

Designs can also be viewed as bipartite graphs on the partitioned set given by the disjoint union of the blocks and points. More concretely, we can represent a design as an *incidence matrix*, a $v \times b$ matrix χ with $\chi_{i,j} = 1$ if and only if $(i,j) \in I$, and $\chi_{i,j} = 0$ otherwise. Throughout the paper we will mainly use the incidence matrix representations of designs.

Definition 2.2. A design $\chi : b \longrightarrow v$ is called

- *k-uniform*, if every block contains exactly k points:

$$\sum_{i=1}^{v} \chi_{i,j} = k, \quad \text{for all} \quad j = 1,..,b$$

- *r-regular*, if every point appears in exactly r blocks:

$$\sum_{j=1}^{b} \chi_{i,j} = r \quad \text{for all} \quad i = 1,..,v$$

Definition 2.3. A k-uniform and r-regular designs is called λ*-balanced*, if any two points are contained in exactly λ blocks. We then have:

$$\chi \cdot \chi^T = \lambda \left(E_{v\times v} - \mathbb{I}_{v\times v} \right) + r\mathbb{I}_{v\times v}$$

Here χ^T is the transpose incidence matrix, $E_{v\times v}$ denotes the $v \times v$-matrix in which every entry is equal to 1, and $\mathbb{I}_{v\times v}$ denotes the $v \times v$ identity matrix.

The last expression from above means that by multiplying χ with its transpose one obtains a matrix where every off-diagonal entry is equal to λ and every diagonal entry is equal to r. These properties combine to give the important notion of block design.

Definition 2.4. A *block design*, or a (v,k,r,b,λ)*-design*, is a design $\chi : b \longrightarrow v$ which is k-uniform, r-regular and λ-balanced.[2]

By simple counting arguments, one can easily derive the following equational properties ([15], p. 4-5):

Lemma 2.5. *For a* (v,k,r,b,λ)*-design, the following equations hold:*

$$b \cdot k = r \cdot v \tag{1}$$

$$\lambda(v-1) = r(k-1) \tag{2}$$

[2] What we define here is actually known as a balanced incomplete block design (BIBD) in the literature, whereas block designs define a more general concept. For the sake of simplicity we will call BIBD's block designs.

Definition 2.6. A (v,k,r,b,λ)-design is *symmetric* when $v = b$; that is, when there are as many points as blocks. Lemma 2.5 then implies that $r = k$.

Example 2.7 ([15], p.27)**.** Consider a finite projective plane of order d. We then have $v = d^2 + d + 1$ points and $b = d^2 + d + 1$ lines such that there are $k = d + 1$ points on each line and each point appears on $r = d + 1$ lines. Moreover, every pair of lines intersect in exactly one point. Hence we have a symmetric block design with parameters $v = b = d^2 + d + 1$, $r = k = d + 1$ and $\lambda = 1$.

We now consider the appropriate notion of homomorphism of block design.

Definition 2.8. Consider two designs $\chi : b \longrightarrow v$ and $\chi' : b' \longrightarrow v'$. A *design homomorphism* $f : \chi \to \chi'$ is a pair of functions $f_v : v \to v'$ and $f_b : b \to b'$ such that the following diagram commutes:

$$\begin{array}{ccc} b & \xrightarrow{f_b} & b' \\ \downarrow{\chi} & & \downarrow{\chi'} \\ v & \xrightarrow{f_v} & v' \end{array}$$

We can use this to obtain categories of designs and block designs, as follows.

Definition 2.9. The category **Design** has designs as objects, and design homomorphisms as morphisms. The category **Block** is the full subcategory on the block designs.

This definition could alternatively be given in terms of points and blocks, but we define it in this abstract way to better relate the categorical machinery to follow. A related definition was given by Dörfler and Waller [2] who explored categories of hypergraphs. Since hypergraphs generalise relations, designs are instances of hypergraphs. In their definition they use the power-set functor to assign to each edge (block) the set of vertices (points) it is incident with. For our notion of a block design we do not make explicit use of the power-set functor; however, this could be an interesting approach since the category of relations is the Kleisli category of the power-set functor, and we would like to consider it in future work.

2.2 Quantum designs

A notion of quantum design has been presented by Zauner in his PhD thesis [20]. Here we recall that definition, adapting the terminology slightly for consistency.

Definition 2.10. A *quantum* (v,b)*-design* is a set $D = \{p_1, ..., p_v\}$ of complex orthogonal $b \times b$ projection matrices p_i on a b-dimensional Hilbert space $\mathbb{C}^b$, i.e. $p_i = p_i^\dagger = p_i^2$ for all $i \in \{1, \ldots, v\}$.

As for classical designs above, we introduce certain properties for quantum designs.

Definition 2.11. A quantum (v,b)-design is called

- *r-regular* if there exists some $r \in \mathbb{N}$ with $\mathrm{Tr}(p_i) = r$ for all $i \in \{1, \ldots, v\}$;
- *k-uniform* if there exists some $k \in \mathbb{R}$ with $\sum_{i=1}^{v} p_i = k \cdot \mathbb{I}_{b\times b}$.

Definition 2.12. Given a quantum (v,b)-design, its *degree* is the cardinality of the set $\{\mathrm{Tr}(p_i p_j) | i, j \in \{1, \ldots, v\}, i \neq j\}$.

It follows that a quantum design has degree 1 just when there exists some $\lambda \in \mathbb{R}$ such that

$$\mathrm{Tr}(p_i p_j) = \lambda \qquad \forall i, j = 1, ..., v \qquad \text{with} \qquad i \neq j. \tag{3}$$

We will call such a quantum design λ*-balanced*. That λ is real in this case follows from a simple argument: $\lambda = \mathrm{Tr}(p_i p_j) = \mathrm{Tr}((p_i p_j)^\dagger)^* = \mathrm{Tr}(p_j^\dagger p_i^\dagger)^* = \mathrm{Tr}(p_j p_i)^* = \mathrm{Tr}(p_i p_j)^* = \lambda^*$.

The following lemma can then be established analogous to Lemma 2.5 for classical designs.

Lemma 2.13. *For a k-uniform, r-regular and λ-balanced quantum design $D = \{p_1, ..., p_v\}$ with $p_i \in \mathbb{C}^b$ the following equations hold:*

$$b \cdot k = v \cdot r \tag{4}$$

$$\lambda(v-1) = r(k-1) \tag{5}$$

The proof for this lemma can be found in Appendix B.

Definition 2.14. A quantum design is *commutative* when all projection matrices pairwise commute.

Theorem 2.15 ([20], Theorem 1.10)**.** *A commutative quantum design is equivalent to a classical block design.*[3]

A prominent example of a uniform and regular quantum design with degree 2 are *mutually unbiased bases* (MUBs). These objects play a significant role in quantum information theory.

Example 2.16 (MUBs.)**.** Mutually unbiased bases are a pair of bases $\{|a_i\rangle\}_{i=0,...,d-1}, \{|b_i\rangle\}_{i=0,...,d-1}$ for a d-dimensional Hilbert space H, such that the inner products $\langle a_i|b_j\rangle$ are equal for all $i, j = 0, ..., d-1$. A uniform and regular quantum design of degree 2 with parameters $r = 1$, $b = d$, $v = d \cdot k$ and $\Lambda = \{\frac{1}{d}, 0\}$ defines a set of k MUB's in a d-dimensional Hilbert space H. To see that, note that the $d \cdot k$ projectors all have trace one and satisfy the following condition, where a labels the different orthogonal classes, and i labels the projectors within an orthogonal class:

$$\sum_{a=1}^{k} \sum_{i=1}^{d} p_i^a = k \cdot \mathbb{I} \tag{6}$$

Moreover, the following holds:

$$\mathrm{tr}(p_i^a p_j^b) = \frac{1}{d}(1 - \delta_{ab}) + \delta_{ij}\delta_{ab}$$

It is easy to see that we get a *complete* set of MUBs if v equals $d\,(d+1)$, as we then have $k = d+1$.[4]

3 Category Theory

In this section we give the definition of an arrow category, and explain the CP-construction. We will assume familiarity with basic concepts in category theory and the graphical calculus, and refer to Maclane [7] and Heunen and Vicary [16] for further background.

The categories we will mostly use in this paper are the category of matrices and natural numbers, **Mat**($\mathbb{N}$), and the category of finite dimensional Hilbert spaces and bounded linear maps, **FHilb**.

Example 3.1. (i) ([16], p. 16) The category **FHilb** has as objects finite dimensional Hilbert spaces and as morphisms bounded linear maps between Hilbert spaces. Composition is the composition of linear maps as ordinary functions and the identity morphisms are given by identity linear maps. The monoidal product is given by the tensor product on Hilbert spaces and the unit object is the one-dimensional Hilbert space $\mathbb{C}$.

[3]This holds because every commutative design is unitarily equivalent to a design comprised of diagonal matrices; as the projections are idempotent, the diagonal entries must therefore be 0 or 1 [20].

[4]Complete means that we have $d+1$ MUBs in a d-dimensional Hilbert space.

(ii) ([13], p. 4) The category **Mat**($\mathbb{N}$) of matrices over $\mathbb{N}$ has objects given by natural numbers. For $m,n \in \mathbb{N}$, the Hom-set $\mathrm{Hom}_{\mathbf{Mat}(\mathbb{N})}(m,n)$ is the set of all $n \times m$-matrices over $\mathbb{N}$, composition being matrix multiplication. The monoidal product on objects is given by the multiplication of numbers and on morphisms by the Kronecker product of matrices. The monoidal unit is the natural number 1.

Definition 3.2 (See [13], p. 23-24)**.** For a category $\mathscr{C}$, its *arrow category* $\mathbf{Arr}[\mathscr{C}]$ is defined as follows:

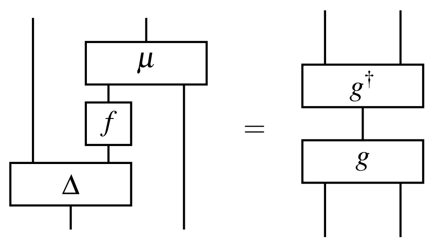
- objects are triples (A,B,h) with $h : A \to B$ in $\mathscr{C}$;
- morphisms $\phi : (A,B,h) \to (A',B',h')$ are pairs of morphisms $\phi_A : A \to A'$ and $\phi_B : B \to B'$ in $\mathscr{C}$ such that the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, "\phi_A"] \arrow[d, "h"'] & A' \arrow[d, "h'"] \\ B \arrow[r, "\phi_B"] & B' \end{tikzcd}$$

3.1 The CP construction

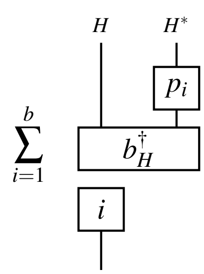
The concept of completely positive maps is well-established [8]. Here we use Selinger's categorical description of completely positive maps [14], as follows, exploiting the notion of dagger Frobenius structure, which is standard in the categorical quantum mechanics literature [16].

Definition 3.3. In a monoidal dagger category, let (A,μ_A,η_A) and (B,μ_B,η_B) be dagger Frobenius structures.[5] A morphism $f : A \to B$ satisfies the *CP-condition* if there exists some object X and some morphism $g : A \otimes B \to X$ such that the following equation holds:



One can show that, in a symmetric monoidal dagger category, a morphism that satisfies this condition constitutes a CP-map [16].

Example 3.4. In **FHilb**, consider a POVM consisting of b projections $p_i : H \to H$. One can define a completely positive map $\varphi : \mathbb{C}^b \to H \otimes H^*$ that sends the computational basis vector $|i\rangle$ to p_i. Graphically, we can represent φ as follows, where $b_H : \mathbb{C} \to H^* \otimes H$ is the evaluation map:



[5] A dagger Frobenius structure is a monoid structure which, together with its dagger, satisfies a Frobenius condition. For an introduction to dagger Frobenius structures we refer to Heunen and Vicary [16].

Proposition 3.5. *Let $(\mathscr{C}, \otimes_{\mathscr{C}}, \mathbb{I}_{\mathscr{C}})$ be a monoidal dagger category. There is a category* $\mathbf{CP}[\mathscr{C}]$ *in which*

- *objects are special symmetric dagger Frobenius structures in $\mathscr{C}$*
- *morphisms are morphisms of $\mathscr{C}$ that satisfy the CP-condition.*

Example 3.6. In $\mathbf{CP}[\mathbf{FHilb}]$ objects are finite dimensional H^*-algebras, i. e. an algebra A that is also a Hilbert space with an anti-linear involution $\dagger : A \to A$ satisfying $\langle ab|c\rangle = \langle b|a^\dagger c\rangle = \langle a|cb^\dagger\rangle$, and morphisms are completely positive maps.

Proposition 3.7. *The category* $\mathbf{CP_c}[\mathscr{C}]$ *with* classical structures*, i.e. special commutative dagger Frobenius structures in $\mathscr{C}$, as objects and completely positive maps between these structures as morphisms, is a subcategory of* $\mathbf{CP}[\mathscr{C}]$.

Proposition 3.8 ([16], p. 241)**.** *The category* $\mathbf{CP_c}[\mathbf{FHilb}]$ *is monoidally equivalent to* $\mathbf{Mat}(\mathbb{N})$.

An interpretation of these constructions is the following: $\mathscr{C}$ models pure state quantum mechanics, $\mathbf{CP}[\mathscr{C}]$ models mixed state quantum mechanics, while $\mathbf{CP_c}[\mathscr{C}]$ describes statistical mechanics [16].

The final piece of structure we require is that of pointed monoidal category.

Definition 3.9. A *pointed* monoidal category is a monoidal category for which every object A is equipped with a canonical morphism $p_A : \mathbb{I} \to A$.

Example 3.10. We obtain examples as follows from the categories we have been considering:

(i) The category $\mathbf{CP}[\mathscr{C}]$ has a pointed structure given by the adjoint of the trace map $V \otimes V^* \to \mathbb{I}$.

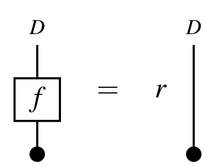
(ii) In $\mathbf{Mat}(\mathbb{N})$ a pointed structure is given by a column matrix with a 1 at every entry: $p_n : 1 \to n$.

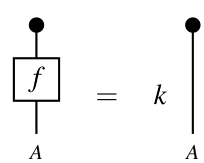
4 Categorical Block Designs

In this section we will develop a construction that gives an abstract notion of the uniformity-, regularity- and λ-balanced condition from Section 2.1 in an arbitrary rigid monoidal category. We will call this the *design construction*.

Definition 4.1 (Design construction)**.** Let $F : \mathscr{D} \hookrightarrow \mathscr{C}$ be a faithful monoidal functor between pointed monoidal dagger categories. The category $\mathbf{Design}[\mathscr{C}, \mathscr{D}]$ is the subcategory of $\mathbf{Arr}[\mathscr{C}]$ where the morphisms are given by pairs of morphisms of $\mathscr{C}$ which are in the image of the functor F; we omit F from the notation, ensuring it is clear from the context. Where $F = \mathrm{id}$, we simply write $\mathbf{Design}[\mathscr{C}]$.

Definition 4.2. The category $\mathbf{RUDesign}[\mathscr{C}, \mathscr{D}]$ is the subcategory of $\mathbf{Design}[\mathscr{C}, \mathscr{D}]$ where objects $f : A \to D$ are r-regular and k-uniform, for scalars $r, k \in \mathrm{Hom}(\mathbb{I}_{\mathscr{C}}, \mathbb{I}_{\mathscr{C}})$, with the pointed structure and its dagger represented by a black dot:





Lemma 4.3. *In* $\mathbf{RUDesign}[\mathscr{C}, \mathscr{D}]$ *for any k-uniform, r-regular object $f : A \to D$, the following equations hold:*

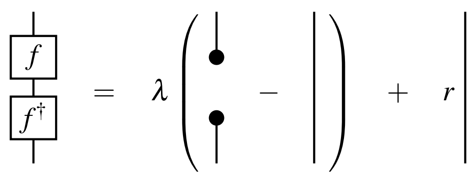
$$k \cdot \dim(D) = r \cdot \dim(A) \tag{7}$$

where $\dim(A) = p_A^\dagger \circ p_A$ *for $A \in \mathrm{obj}(\mathscr{C})$. Here $p_A : 1 \to A$ is the pointed structure of $\mathscr{C}$.*

Proof. Via composition with $p_D^\dagger$ and p_A respectively, the regularity and uniformity condition become:

Hence Eq. 7 holds. $\square$

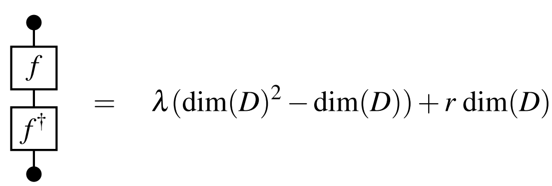
Definition 4.4. The category **BDesign**$[\mathscr{C}, \mathscr{D}]$ is the subcategory of **RUDesign**$[\mathscr{C}, \mathscr{D}]$ where all k-uniform and r-regular objects $f : A \to D$ are λ-balanced for scalars $\lambda \in \mathrm{Hom}(\mathbb{I}_{\mathscr{C}}, \mathbb{I}_{\mathscr{C}})$:



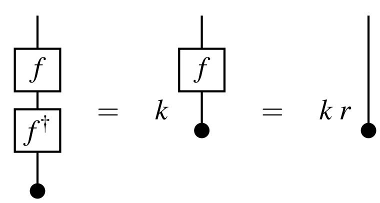
Lemma 4.5. *In* **BDesign**$[\mathscr{C}, \mathscr{D}]$ *for any k-uniform, r-regular object* $f : A \to D$*, the following equation holds, where* $\dim(D) = p_D^\dagger \circ p_D$ *for* $D \in \mathrm{obj}(\mathscr{C})$*:*

$$\lambda \cdot (\dim(D) - 1) = k \cdot (r - 1) \tag{8}$$

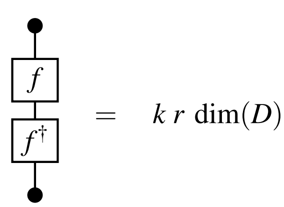
Proof. To prove Eq. 8, we concatenate the λ-condition with both p_A and $p_A^\dagger$ which gives:



On the other hand we have:



If we now concatenate with $p_D^\dagger$, we get:



From this we can easily deduce Eq. 8. $\square$

5 Classical and Quantum Models

In this section we will apply the design-constructions from Section 4 to our model categories $\mathbf{Mat}(\mathbb{N})$ and $\mathbf{CP}[\mathbf{FHilb}]$ and show that this gives us a categorical model of both classical quantum designs.

5.1 The Category of Block Designs

Writing **FSet** for the category of finite sets and functions, there is a faithful functor $\mathbf{FSet} \hookrightarrow \mathbf{Mat}(\mathbb{N})$ which takes every set to the natural number given by its cardinality. Moreover, we have that $\mathbf{FSet} \hookrightarrow \mathbf{Mat}(\mathbb{N}) \cong \mathbf{CP_c}[\mathbf{FHilb}]$ (see Theorem 3.8). In the following we will prove that there exists a functor from the category $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ to the category **Block**. Moreover, we will show that $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ is equivalent to $\mathbf{BDesign}[\mathbf{CP_c}[\mathbf{FHilb}], \mathbf{FSet}]$.

Theorem 5.1. *There exists a functor* $G : \mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}] \longrightarrow \mathbf{Block}$.

Proof. We first note that the morphisms in $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ are given by pairs of functions. The functor sends each object in $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ to an incidence matrix in **Block** by sending each matrix entry greater than 0 to 1. The uniformity, regularity and λ-balance conditions of the design construction ensure that the incidence matrix we obtain that way, represents a uniform, regular and λ-balanced design. On morphisms the functor acts as the identity. □

Similarly, one can argue that the following holds.

Theorem 5.2. *There exists a functor* $G : \mathbf{Design}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}] \longrightarrow \mathbf{Design}$.

Note that this indicates that the categories $\mathbf{Design}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ and $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ actually define a more general concept of (block)design. We will refer to it as *categorical (block)designs*.

Lemma 5.3. *The category* $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ *is equivalent to* $\mathbf{BDesign}[\mathbf{CP_c}[\mathbf{FHilb}], \mathbf{FSet}]$.

Proof. According to Proposition 3.8 the categories $\mathbf{CP_c}[\mathbf{FHilb}]$ and $\mathbf{Mat}(\mathbb{N})$ are equivalent. Using Theorem A.5 from Appendix A, this gives rise to an equivalence between their arrow categories. □

5.2 The Category of Quantum Designs

In this section we will define a category of quantum designs by applying the design construction to the category $\mathbf{CP}[\mathbf{FHilb}]$. Moreover, we will show that this category contains two important subcategories: $\mathbf{QDesign}_{\mathrm{B}}$ which has objects that are uniform and regular quantum designs of degree 1, and $\mathbf{QDesign}_{\mathrm{RU}}$ that has uniform and regular quantum designs as objects. We will demonstrate that the latter actually contains a subcategory **MUB**, with objects that are sets of mutually unbiased bases.

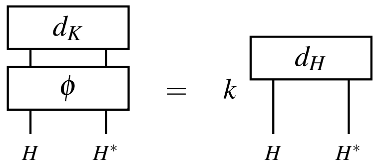
Recall from Section 3 that $\mathbf{CP}[\mathbf{FHilb}]$ is comprised of finite dimensional H^*-algebras and completely positive maps. Applying the design construction using the identity functor, we get a category $\mathbf{Design}[\mathbf{CP}[\mathbf{FHilb}]]$, with objects that are CP-maps between finite dimensional H^*-algebras, and morphisms that are pairs of CP-maps.

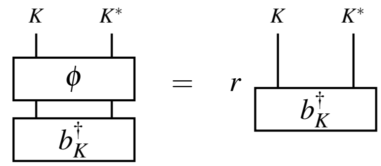
Definition 5.4. The category **QDesign** is defined to be the category $\mathbf{Design}[\mathbf{CP}[\mathbf{FHilb}]]$.

Definition 5.5. The subcategory $\mathbf{RUDesign}[\mathbf{CP}[\mathbf{FHilb}]]$ of **QDesign** is called $\mathbf{QDesign}_{\mathrm{RU}}$. Its objects are uniform and regular quantum designs.

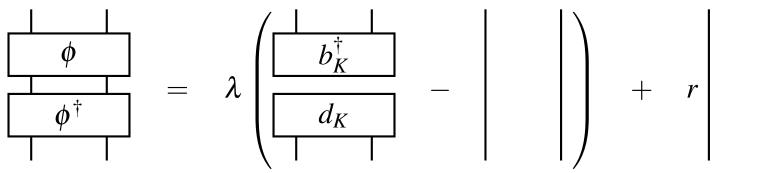
Definition 5.6. The subcategory $\mathbf{BDesign}[\mathbf{CP}[\mathbf{FHilb}]]$ of $\mathbf{QDesign}_{\mathrm{RU}}$ is called $\mathbf{QDesign}_{\mathrm{B}}$. Its objects are uniform and regular quantum designs with degree 1.

Example 5.7. Consider the subcategory of $\mathbf{QDesign}_\mathrm{B}$ where all objects are CP-maps between matrix algebras: $\phi : H \otimes H^* \to K \otimes K^*$ where $\dim(H) = b$ and $\dim(K) = v$. Because H is a special Frobenius algebra, we get $\dim(H) = \mathrm{tr}(\mathrm{id}_H)$. We then have the following uniformity and regularity conditions:





The λ-condition is given by:



Let $H = \mathbb{C}^2 = K$ and consider the CP-map $\varphi : \mathbb{C}^2 \otimes \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2$ with matrix representation:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

This map represents a quantum design with parameters $\lambda = k = r = 2 = v = b$.

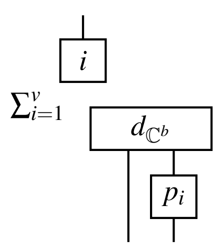
Theorem 5.8. *Every* 1*-uniform, r-regular and* λ*-balanced quantum design of the form* $S : H \otimes H^* \to K \otimes K^*$ *defines a superoperator.*

Proof. By definition, S is completely positive. Applying the uniformity condition to $S(\rho)$, where ρ is a an arbitrary state in $H \otimes H^*$ shows that S is also trace-preserving. □

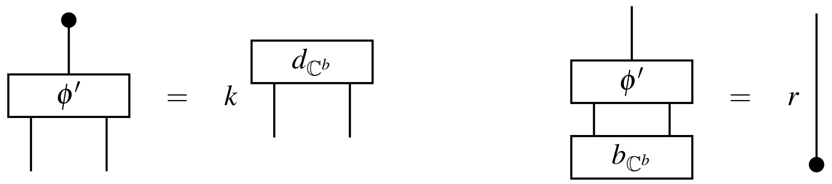
In the previous example we have considered completely positive maps from a non-commutative algebra to a non-commutative algebra in **FHilb**. We can also consider CP-maps from a commutative algebra to a non-commutative algebra, i. e. maps of the form: $\varphi : H \to K^* \otimes K$. In fact, we can encode uniform and regular quantum designs of degree 1 according to Zauner's notion via these maps.

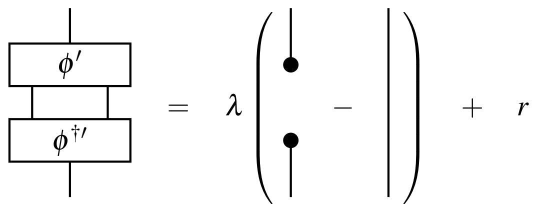
Theorem 5.9. *There exists a subcategory of* $\mathbf{QDesign}_\mathrm{B}$ *that has objects that represent uniform and regular quantum designs of degree 1 according to Zauner's notion.*

Proof. Consider a uniform, regular and λ-balanced quantum design $D = \{p_1, ..., p_v\}$, where each p_i is a $b \times b$ projection matrix in a Hilbert space $\mathbb{C}^b$. Following Example 3.4, these projections $p_i : \mathbb{C}^b \to \mathbb{C}^b$ then give rise to a completely positive map $\phi : \mathbb{C}^v \to \mathbb{C}^b \otimes \mathbb{C}^b$ in **FHilb**. This is valid because imposing uniformity, regularity and being λ-balanced on the projector has no impact on the CP-condition. Now take $\phi' = \phi^\dagger : \mathbb{C}^b \otimes \mathbb{C}^b \to \mathbb{C}^v$, i. e.



Then we find:

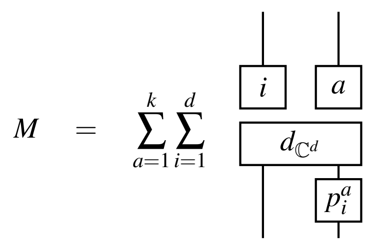




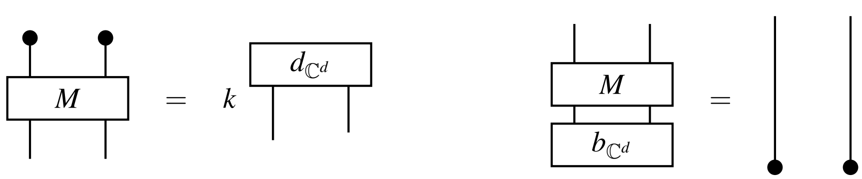
These coincide with the conditions given by the design construction applied to **FHilb**. Moreover, we can recover Eq. 4 and Eq. 5 from Lemma 2.13 as $\dim(\mathbb{C}^v) = v$ and $\dim(\mathbb{C}^b) = b$ and Eq. 7 and Eq. 8 hold. $\square$

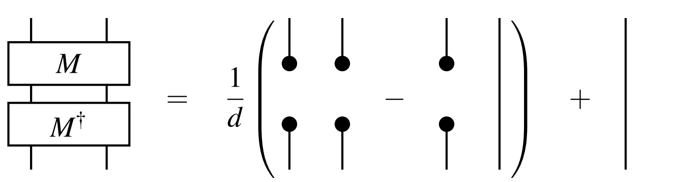
Theorem 5.10. *There exists a subcategory* **MUB** *of* $\mathbf{QDesign}_{\mathrm{RU}}$ *with objects that are collections of MUBs and morphisms that are pairs of functions.*

Proof. Consider the CP-map $M : \mathbb{C}^{k\cdot d} \cong \mathbb{C}^d \otimes \mathbb{C}^k \to \mathbb{C}^d \otimes \mathbb{C}^d$:



This map satisfies the following equations:





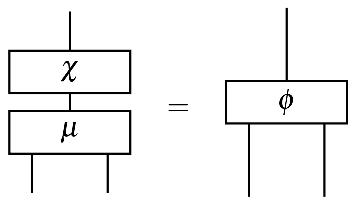
Here the last equation can be understood as a generalised λ-equation. Restricting $\mathbf{QDesign}_{RU}$ to objects of this form, we get a category that has objects that are 1-uniform and k-regular quantum designs of degree 2 where $\Lambda = \{\frac{1}{d}, 0\}$, i. e. k MUBs in dimension d. $\square$

5.3 Relating BDesign to QDesign

In this section we will construct a functor between $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ and $\mathbf{QDesign}_B$.

Proposition 5.11. *There exists a functor $Q : \mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}] \to \mathbf{QDesign}_B$ that relates a generalized balanced incomplete block designs to uniform, regular and λ-balanced quantum designs.*

Proof. According to Lemma 5.3 the categories $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ and $\mathbf{BDesign}[\mathbf{CP}_c[\mathbf{FHilb}],\mathbf{FSet}]$ are equivalent. So we can actually represent an arbitrary object $\chi : b \to v$ in $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ via a uniform, regular and λ-balanced CP-map $\chi : \mathbb{C}^b \to \mathbb{C}^v$. The functor Q acts on objects by sending each object $\chi : \mathbb{C}^b \to \mathbb{C}^v$ in $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ with parameters k, r and λ to the map $\phi = \chi \circ L : \mathbb{C}^b \otimes \mathbb{C}^b \to \mathbb{C}^b \to \mathbb{C}^v$, where the map $L : \mathbb{C}^b \otimes \mathbb{C}^b \to \mathbb{C}^b$ is the so-called Cayley embedding, which in our case simply becomes the multiplication $\mu : \mathbb{C}^b \otimes \mathbb{C}^b \to \mathbb{C}^b$ as we have that $A = \mathbb{C}^b \cong (\mathbb{C}^b)^* = A^*$. Its conjugate $L^\dagger$ is just the comultiplication $\Delta : \mathbb{C}^b \to \mathbb{C}^b \otimes \mathbb{C}^b$. The resulting map ϕ is as concatenation of completely positive maps also completely positive. We depict this via the following string diagram:



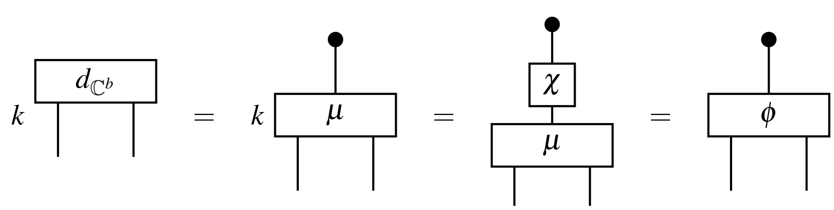
Via concatenation, each morphism in $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$

$$\begin{array}{ccc} \mathbb{C}^b & \xrightarrow{\xi'} & \mathbb{C}^{b'} \\ \downarrow{\scriptstyle \chi} & & \downarrow{\scriptstyle \chi'} \\ \mathbb{C}^v & \xrightarrow{\xi} & \mathbb{C}^{v'} \end{array}$$

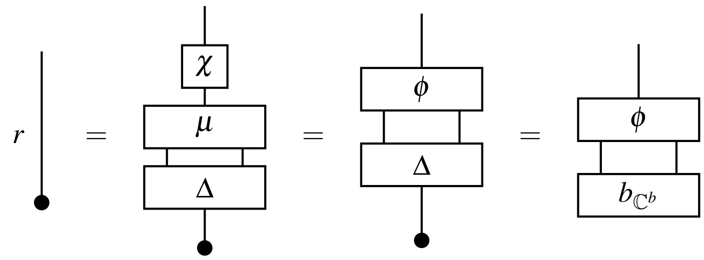
gets mapped to a morphism in $\mathbf{QDesign}_B$, as follows:

$$\begin{array}{ccc} \mathbb{C}^b \otimes \mathbb{C}^b & \xrightarrow{\xi' \otimes \xi'} & \mathbb{C}^{b'} \otimes \mathbb{C}^{b'} \\ \downarrow{\scriptstyle \mu} & & \downarrow{\scriptstyle \mu} \\ \mathbb{C}^b & \xrightarrow{\xi'} & \mathbb{C}^{b'} \\ \downarrow{\scriptstyle \chi} & & \downarrow{\scriptstyle \chi'} \\ \mathbb{C}^v & \xrightarrow{\xi} & \mathbb{C}^{v'} \end{array}$$

This diagram commutes, because ξ' can be extended to a morphism of monoids as ξ' is a function. It is easy to verify that this functor respects composition and sends the identity morphism in $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$, i. e. $\mathrm{id}_\psi = (\mathrm{id},\mathrm{id})$, to the identity morphism $\mathrm{id}_{Q(\psi)} = (\mathrm{id}\otimes\mathrm{id},\mathrm{id}\otimes\mathrm{id})$ in $\mathbf{QDesign}_{\mathrm{B}}$. The regularity condition then becomes:



which is exactly the regularity condition in $\mathbf{QDesign}_{\mathrm{B}}$. Note that we have used the fact that $\mathbb{C}^b$ is a special Frobenius algebra in the first step. in the second step. For uniformity we find:



which is precisely the uniformity condition in $\mathbf{QDesign}_{\mathrm{B}}$. In a similar way one can verify that the λ-condition in $\mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}]$ gets mapped to the λ-condition in $\mathbf{QDesign}_{\mathrm{B}}$. $\square$

In this construction every classical design gives rise to a uniform and regular quantum design of degree 1, analogously to Theorem 2.15. However, it is straightforward to verify that the functor Q does not yield an equivalence of categories, as it is not essentially surjective.

A widely-discussed topic is the existence of MUBs in non-primepower dimensions. One can ask if it is possible to extend the functor Q to a functor $\widetilde{Q} : \mathbf{BDesign}[\mathbf{Mat}(\mathbb{N}), \mathbf{FSet}] \to \mathbf{MUB}$ that maps a classical design to a set of MUBs. We conjecture that there does not exist a classical design that gets sent to a **MUB** via Q.

References

[1] F. Buekenhout. *Handbook of Incidence Geometry. Buildings and Foundations*. North-Holland, 1995.

[2] Willibald Dörfler and D. A. Waller. A category-theoretical approach to hypergraphs. *Archiv der Mathematik*, 34:185–192, 1980.

[3] Paulina L. A. Goedicke and Jamie Vicary. On structures in arrow categories, 2023. `arXiv:2309.15544`.

[4] D. Goyeneche and Karol Życzkowski. Genuinely multipartite entangled states and orthogonal arrays. *Physical Review A*, 90, 04 2014. `doi:10.1103/PhysRevA.90.022316`.

[5] Dardo Goyeneche, Daniel Alsina, José I. Latorre, Arnau Riera, and Karol Życzkowski. Absolutely maximally entangled states, combinatorial designs, and multiunitary matrices. *Phys. Rev. A*, 92:032316, Sep 2015. URL: `https://link.aps.org/doi/10.1103/PhysRevA.92.032316`, `doi:10.1103/PhysRevA.92.032316`.

[6] Dardo Goyeneche, Zahra Raissi, Sara Di Martino, and Karol Życzkowski. Entanglement and quantum combinatorial designs. *Physical Review A*, 97(6), Jun 2018. `doi:10.1103/physreva.97.062326`.

[7] Saunders Mac Lane. *Categories for the Working Mathematician. Graduate Texts in Mathematics. Vol. 5*. Springer, 2013.

[8] Gerald Murphy. *C star Algebras and Operator Theory*. Academic Press, 1990.

[9] Benjamin Musto. Constructing mutually unbiased bases from quantum Latin squares. *Electronic Proceedings in Theoretical Computer Science*, 236:108–126. `doi:10.4204/eptcs.236.8`.

[10] Benjamin Musto and Jamie Vicary. Quantum Latin squares and unitary error bases, 2016. `arXiv:1504.02715`.

[11] Suhail Ahmad Rather, Adam Burchardt, Wojciech Bruzda, Grzegorz Rajchel-Mieldzioć, Arul Lakshminarayan, and Karol Życzkowski. Thirty-six entangled officers of Euler: Quantum solution to a classically impossible problem. *Physical Review Letters*, 128(8), 2022. `doi:10.1103/physrevlett.128.080507`.

[12] David J. Reutter and Jamie Vicary. Biunitary constructions in quantum information, 2019. `arXiv:1609.07775`.

[13] Steven Roman. *An Introduction to the Language of Category Theory*. Birkhäuser, 2017.

[14] Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical Computer Science*, 170:139–163, 03 2007. `doi:10.1016/j.entcs.2006.12.018`.

[15] Douglas R. Stinson. *Combinatorial Designs: Construction and Analysis*. Springer, 2013.

[16] Jamie Vicary and Chris Heunen. *Categories for Quantum Theory*. Oxford University Press, 2020.

[17] David White and Donald Yau. Arrow categories of monoidal model categories, 2018. `arXiv:1703.05359`.

[18] Pawel Wocjan and Thomas Beth. New construction of mutually unbiased bases in square dimensions. *Quantum Information and Computation*, 5, 08 2004.

[19] William Wootters. Quantum measurements and finite geometry. *Foundations of Physics*, 36:112–126, 02 2006. `doi:10.1007/s10701-005-9008-x`.

[20] G. Zauner. Quantum designs. foundations of a non-commutative design theory. *University of Vienna*, 1999.

A Arrow categories

In this section we will review the concept of arrow categories and derive some facts about arrow categories which we believe to be new. We will focus only on content relevant for the overall purpose of this paper. However, there are more results on that topic, covering Hopf algebras, Frobenius structures and topological field theories in arrow categories which can be found in [3].

Definition A.1 (See [13], pages 23-24)**.** For a category $\mathscr{C}$, its arrow category $\mathbf{Arr}[\mathscr{C}]$ is defined as follows:

- objects are triples (A,B,h) with $h : A \to B$ in $\mathscr{C}$;
- morphisms $\phi : (A,B,h) \to (A',B',h')$ are pairs of morphisms $\phi_A : A \to A'$ and $\phi_B : B \to B'$ in $\mathscr{C}$ such that the following diagram commutes in $\mathscr{C}$:

$$\begin{array}{ccc} A & \xrightarrow{\phi_A} & A' \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle h'} \\ B & \xrightarrow{\phi_B} & B' \end{array}$$

In the following we will show that an arrow category inherits certain structures from their underlying category. This includes functors, natural transformations and the monoidal product.

Proposition A.2. *Given a functor $F : \mathscr{C} \to \mathscr{D}$, we apply the arrow construction to obtain a functor $\widetilde{F} : \mathbf{Arr}[\mathscr{C}] \to \mathbf{Arr}[\mathscr{D}]$.*

Proof. Given a functor $F : \mathscr{C} \to \mathscr{D}$, we can define a functor $\widetilde{F} : \mathbf{Arr}[\mathscr{C}] \to \mathbf{Arr}[\mathscr{D}]$ as follows. On objects, we map $f : A \to B$ in $\mathbf{Arr}[\mathscr{C}]$ to an object $F(f) : F(A) \to F(B)$ in $\mathbf{Arr}[\mathscr{D}]$. On morphisms, we map $(\phi, \psi) : f \to f'$ in $\mathbf{Arr}[\mathscr{C}]$ to a morphism $\tilde{F}(\phi, \psi) = (F(\phi), F(\psi)) : F(f) \to F(f')$ in $\mathbf{Arr}[\mathscr{D}]$. This is valid because the diagram

$$\begin{array}{ccc} F(A) & \xrightarrow{F(\phi)} & F(A') \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle F(f')} \\ F(B) & \xrightarrow{F(\psi)} & F(B') \end{array}$$

commutes due to functoriality of F. Moreover, we have

$$\widetilde{F}(\mathrm{id}_A, \mathrm{id}_B) = (F(\mathrm{id}_A), F(\mathrm{id}_B)) = (\mathrm{id}_{F(A)}, \mathrm{id}_{F(B)}), \tag{9}$$

where $(\mathrm{id}_A, \mathrm{id}_B)$ is the identity morphism in $\mathbf{Arr}[\mathscr{C}]$. Due to functoriality of F and because the concatenation of two commuting diagrams yields again a commuting diagram, $\tilde{F}$ also preserves composition. □

Similarly, a contravariant functor $F : \mathscr{C} \to \mathscr{D}$ gives rise to a contravariant functor $\widetilde{F} : \mathbf{Arr}[\mathscr{C}] \to \mathbf{Arr}[\mathscr{D}]$.

Proposition A.3. *Let $F, G : \mathscr{C} \to \mathscr{D}$ be two functors between two categories $\mathscr{C}$ and $\mathscr{D}$, and let $\widetilde{F}, \widetilde{G} : \mathbf{Arr}[\mathscr{C}] \to \mathbf{Arr}[\mathscr{D}]$ be the induced functors on the arrow categories. A natural transformation $\eta : F \Rightarrow G$ induces a natural transformation $\tilde{\eta} : \tilde{F} \Rightarrow \tilde{G}$.*

Proof. Let $\eta : F \Rightarrow G$ be a natural transformation that assigns to every object A in $\mathscr{C}$ a morphism $\eta_A : F(A) \to G(A)$, such that for any morphism $f : A \to B$ in $\mathscr{C}$ the following diagram (naturality condition) commutes:

$$\begin{tikzcd} F(A) \arrow[r, "\eta_A"] \arrow[d, "F(f)"'] & G(A) \arrow[d, "G(f)"] \\ F(B) \arrow[r, "\eta_B"] & G(B) \end{tikzcd}$$

One can use the naturality of η to define a natural transformation $\widetilde{\eta} : \widetilde{F} \Rightarrow \widetilde{G}$ that assigns to every object $f : A \to B$ in $\mathbf{Arr}[\mathscr{C}]$ a morphism $\widetilde{\eta}_f = (\eta_A, \eta_B) : \widetilde{F}(f) \to \widetilde{G}(f)$ via the commutative diagram from above, such that for any morphism $(\phi, \psi) : f \to f'$ in $\mathbf{Arr}[\mathscr{C}]$:

$$\begin{tikzcd} A \arrow[r, "\phi"] \arrow[d, "f"'] & A' \arrow[d, "f'"] \\ B \arrow[r, "\psi"] & B' \end{tikzcd}$$

the following diagram (naturality condition in the arrow category) commutes:

$$\begin{tikzcd} & F(A) \arrow[rr, "\eta_A"] \arrow[dl, "F(\phi)"'] \arrow[dd, dashed, "F(f)" near end] & & G(A) \arrow[dl, "G(\phi)"'] \arrow[dd, "G(f)"] \\ F(A') \arrow[rr, "\eta_{A'}" near end] \arrow[dd, "F(f')"] & & G(A') \arrow[dd, "G(f')" near start] & \\ & F(B) \arrow[rr, dashed, "\eta_B" near start] \arrow[dl, dashed, "F(\psi)"] & & G(B) \arrow[dl, "G(\psi)"] \\ F(B') \arrow[rr, "\eta_{B'}"'] & & G(B') & \end{tikzcd}$$

Here the the top, the back, the front and the bottom face commute due to naturality of η and the two side faces commute by definition. Hence the whole diagram commutes and we have defined a natural transformation $\widetilde{\eta} : \widetilde{F} \Rightarrow \widetilde{G}$. $\square$

Proposition A.4. *If $\eta : F \Rightarrow G$ is a natural isomorphism, then so is $\tilde{\eta} : \widetilde{F} \Rightarrow \widetilde{G}$.*

Theorem A.5. *Let $\mathscr{C}$ and $\mathscr{D}$ be equivalent categories; that is, there exist functors $F : \mathscr{C} \to \mathscr{D}$ and $G : \mathscr{D} \to \mathscr{C}$ and natural isomorphisms $F \circ G \cong \mathrm{id}_{\mathscr{D}}$ and $G \circ F \cong \mathrm{id}_{\mathscr{C}}$. Then Arr($\mathscr{C}$) and Arr($\mathscr{D}$) are also equivalent.*

Proof. By Proposition A.2 the functors $F : \mathscr{C} \to \mathscr{D}$ and $G : \mathscr{D} \to \mathscr{C}$ give rise to functors $\widetilde{F} : \mathbf{Arr}[\mathscr{C}] \to \mathbf{Arr}[\mathscr{D}]$ and $\widetilde{G} : \mathbf{Arr}[\mathscr{D}] \to \mathbf{Arr}[\mathscr{C}]$. From Proposition A.4 we know that the natural isomorphisms $F \circ G \cong \mathrm{id}_{\mathscr{D}}$ and $G \circ F \cong \mathrm{id}_{\mathscr{C}}$ give rise to natural isomorphisms $\widetilde{F} \circ \widetilde{G} \cong \mathrm{id}_{\mathbf{Arr}[\mathscr{D}]}$ and $\widetilde{G} \circ \widetilde{F} \cong \mathrm{id}_{\mathbf{Arr}[\mathscr{C}]}$. Hence we have an equivalence. $\square$

One can show that the same theorems apply to monoidal functors and monoidal natural transformations [3]. Moreover, one can define a monoidal product in the arrow category of a monoidal category as the following proposition will show. However, this result is not necessarily new and can be in fact found in a similar notion in [17].

Proposition A.6. *For a monoidal category $\mathscr{C}$, we can define a monoidal product on* $\mathbf{Arr}[\mathscr{C}]$*, written* $\boxtimes$*, as follows:*

- *on objects,* $f \boxtimes g := f \otimes g$*;*
- *on morphisms,* $(p,q) \boxtimes (p',q') := (p \otimes p', q \otimes q')$.

Proof. We will show that the pentagon and the triangle axiom are satisfied. The pentagon axiom holds due to the following diagram, where the front and the back face commute because α satisfies the ordinary pentagon axiom. The two side faces commute due to the definition of the monoidal product and naturality of the associator, and the top and bottom faces commute due to naturality of the associator:

$$
\begin{tikzcd}
 & A_1\otimes(A_2\otimes(A_3\otimes A_4)) \arrow[r, "\alpha"] \arrow[dl, "f_1\otimes(f_2\otimes(f_3\otimes f_4))"'] \arrow[dd, dashed, "\mathrm{id}_{A_1}\otimes\alpha"] & (A_1\otimes A_2)\otimes(A_3\otimes A_4) \arrow[r, "\alpha"] \arrow[dl, "(f_1\otimes f_2)\otimes(f_3\otimes f_4)"] & ((A_1\otimes A_2)\otimes A_3)\otimes A_4 \arrow[dl, "((f_1\otimes f_2)\otimes f_3)\otimes f_4"] \arrow[dd, "\alpha\otimes\mathrm{id}_{A_4}"] \\
B_1\otimes(B_2\otimes(B_3\otimes B_4)) \arrow[r, "\alpha"] \arrow[dd, "\mathrm{id}_{B_1}\otimes\alpha"] & (B_1\otimes B_2)\otimes(B_3\otimes B_4) \arrow[r, "\alpha"] & ((B_1\otimes B_2)\otimes B_3)\otimes B_4 \arrow[dd, "\alpha"] & \\
 & A_1\otimes((A_2\otimes A_3)\otimes A_4) \arrow[rr, dashed, "\alpha"] \arrow[dl, dashed, "f_1\otimes((f_2\otimes f_3)\otimes f_4)"] & & (A_1\otimes(A_2\otimes A_3))\otimes A_4 \arrow[dl, "(f_1\otimes(f_2\otimes f_3))\otimes f_4"] \\
B_1\otimes((B_2\otimes B_3)\otimes B_4) \arrow[rr, "\alpha"'] & & (B_1\otimes(B_2\otimes B_3))\otimes B_4 &
\end{tikzcd}
$$

The triangle axiom for $\mathbf{Arr}[\mathscr{C}]$ is given by the following diagram:

$$
\begin{tikzcd}
(A\otimes\mathbb{I})\otimes A' \arrow[rr, "\alpha"] \arrow[dr, "\rho\otimes\mathrm{id}_{A'}"] \arrow[dd, "(f\otimes\mathrm{id}_{\mathbb{I}})\otimes f'"] & & A\otimes(\mathbb{I}\otimes A') \arrow[dl, "\mathrm{id}_A\otimes\lambda"'] \arrow[dd, "f\otimes(\mathrm{id}_{\mathbb{I}}\otimes f')"] \\
 & A\otimes A \arrow[dd, "f\otimes f'"] & \\
(B\otimes\mathbb{I})\otimes B' \arrow[rr, dashed, "\alpha"] \arrow[dr, "\rho\otimes\mathrm{id}_{B'}"] & & B\otimes(\mathbb{I}\otimes B') \arrow[dl, "\mathrm{id}_B\otimes\lambda"] \\
 & B\otimes B' &
\end{tikzcd}
$$

Here the top and the bottom faces commute due to the the triangle identity and the two side faces commute due to the definition of the monoidal product in $\mathbf{Arr}[\mathscr{C}]$ and due to naturality of the left and right unitors in $\mathscr{C}$. Finally, the back face commutes because of the naturality of the associator. $\square$

B Proof for Lemma 2.13

Proof. Consider an r-uniform, k-regular and λ-balanced quantum design $D = \{p_1, \ldots, p_v\}$ with $p_i \in \mathbb{C}^b$. By applying the trace function to the regularity condition, we get:

$$\mathrm{Tr}\Big(\sum_{i=0}^{v} p_i\Big) = \sum_{i=0}^{v} \mathrm{Tr}(p_i) = k \cdot \mathrm{Tr}(\mathbb{I}_{b\times b}). \tag{10}$$

Using the uniformity condition this expression becomes:

$$\sum_{i=0}^{v} \mathrm{Tr}(p_i) = \sum_{i=0}^{v} r = v \cdot r = k \cdot \mathrm{Tr}(\mathbb{I}_{b\times b}) = k \cdot b. \tag{11}$$

This proves Eq. 4.

In order to prove Eq. 5, we start with the following expression:

$$b = \mathrm{Tr}(\mathbb{I}_{b\times b}) = \mathrm{Tr}(\mathbb{I}^2_{b\times b}). \tag{12}$$

Using the regularity condition, we get:

$$\mathrm{Tr}(\mathbb{I}^2_{b\times b}) = \frac{1}{k^2}\mathrm{Tr}\Big(\sum_{i=0}^{v} p_i \sum_{j=0}^{v} p_i\Big) = \frac{1}{k^2}\sum_{i,j=0}^{v} \mathrm{Tr}(p_i p_j) = \frac{1}{k^2}\sum_{i,j=0,j\neq i}^{v} \lambda + \frac{1}{k^2}\sum_{i}^{v} r = \frac{1}{k^2}(\lambda v(v-1) + vr) \tag{13}$$

Here we have used the uniformity and the λ-condition in the third step. Hence we have:

$$b = \frac{1}{k^2}(\lambda v(v-1) + vr) \Leftrightarrow \tag{14}$$

$$\frac{b \cdot k}{v} \cdot k = \lambda(v-1) + r \tag{15}$$

Using Eq. 4, we obtain:

$$r \cdot k = \lambda(v-1) + r \tag{16}$$

This is equivalent to Eq. 5. □

C Structures in RUDesign and QDesign

In the following we will discuss some structures of the categories **RUDesign** and **QDesign**.

Theorem C.1. *The category* $\mathbf{RUDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ *is a monoidal category with monoidal product given by the Kronecker product between matrices. In particular, for objects* $\chi : b \longrightarrow v$ *and* $\chi' : b' \longrightarrow v'$ *with parameters* k *resp.* k' *and* r *resp.* r' *we have that* $\chi \otimes \chi' : b \otimes b' \longrightarrow v \otimes v'$ *is an object in* $\mathbf{RUDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ *with parameters* $k \cdot k'$ *and* $r \cdot r'$.

Proof. Since $\mathbf{Mat}(\mathbb{N})$ is a monoidal category, we can apply Prop. A.6 from Appendix A to get a monoidal product on $\mathbf{Arr}[\mathbf{Mat}(\mathbb{N})]$. If we now restrict to matrices representing uniform and regular designs, every $\chi \otimes \chi' : b \cdot b' \longrightarrow v \cdot v'$, where $\chi : b \longrightarrow v$ and $\chi' : b' \longrightarrow v'$ are objects in $\mathbf{Arr}[\mathbf{Mat}(\mathbb{N})]$ with parameters k resp. k' and r resp. r', fulfils the uniformity and regularity conditions with parameters $k \cdot k'$ and $r \cdot r'$. □

Similarly, one can argue that the following can be derived from Prop. A.6:

Theorem C.2. *The category* $\mathbf{Design}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ *is a monoidal category, with monoidal product given by the Kronecker product between matrices.*

As the Kronecker product of two matrices not necessarily fulfils the λ-condition, the category **BDesign** is not monoidal in general.

Proposition C.3. *There exists a functor* $\widetilde{D} : \mathbf{RUDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}] \longrightarrow \mathbf{RUDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$ *that maps each k-uniform and r-regular design* χ *to its dual* χ^T *which is a r-uniform and k-regular design and each pair of functions* $(N_V, N_B) : \chi \longrightarrow \chi'$ *to its transpose* $(N_V^T, N_B^T) : \chi' \longrightarrow \chi$.

Proof. We can define a functor $D : \mathbf{Mat}(\mathbb{N}) \longrightarrow \mathbf{Mat}(\mathbb{N})$ that sends each natural number to itself and each matrix to its transpose. By remark A.2 this functor then gives rise to a contravariant functor $\tilde{D}$: $\mathrm{Arr}[\mathbf{Mat}(\mathbb{N})] \longrightarrow \mathrm{Arr}[\mathbf{Mat}(\mathbb{N})]$ that maps each object, i.e. a matrix, to its transpose and each pair of morphisms, i. e. a pair of matrices, to its transpose. If we now restrict to the subcategory where each object represents a uniform and regular design and all morphisms are pairs of functions, we get a functor: $\widetilde{D} : \mathbf{RUDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}] \longrightarrow \mathbf{RUDesign}[\mathbf{Mat}(\mathbb{N}),\mathbf{FSet}]$. □

Just as in the classical case, **QDesign** is also equipped with some structure.

Theorem C.4. *The category* $\mathbf{QDesign}_{\mathrm{RU}}$ *is a monoidal category.*

Proof. The category $\mathbf{CP}[\mathbf{FHilb}]$ is monoidal [16]. According to Prop. A.6 this gives rise to a monoidal product in $\mathbf{Arr}[\mathbf{CP}[\mathbf{FHilb}]]$. If we now restrict to the case where the objects in $\mathbf{Arr}[\mathbf{CP}[\mathbf{FHilb}]]$ encode uniform and regular quantum designs and the morphisms are pairs of functions, i. e. to the category $\mathbf{QDesign}_{RU}$, it is straightforward to verify that the tensor product of two uniform and regular CP-maps with parameters k, r and k', r' respectively, again fulfils the uniformity and regularity condition with parameters $k \cdot k'$ and $r \cdot r'$. □

Similarly, one can argue that the following has to hold:

Theorem C.5. *The category* **QDesign** *is a monoidal category.*

The monoidal product of two CP-maps satisfying the λ-condition does not satisfy the λ-condition in general and hence one cannot define a monoidal product in $\mathbf{QDesign}_B$ in general.

Constructor Theory as Process Theory

Stefano Gogioso
Hashberg Ltd
University of Oxford
stefano.gogioso@cs.ox.ac.uk

Vincent Wang-Maścianica
Quantinuum Ltd
University of Oxford
vincent.wang@cambridgequantum.com

Muhammad Hamza Waseem
Quantinuum Ltd
University of Oxford
hamza.waseem@physics.ox.ac.uk.

Carlo Maria Scandolo
University of Calgary
carlomaria.scandolo@ucalgary.ca

Bob Coecke
Quantinuum Ltd
bob.coecke@quantinuum.com

Constructor theory is a meta-theoretic approach that seeks to characterise concrete theories of physics in terms of the (im)possibility to implement certain abstract "tasks" by means of physical processes. Process theory, on the other hand, pursues analogous characterisation goals in terms of the compositional structure of said processes, concretely presented through the lens of (symmetric monoidal) category theory. In this work, we show how to formulate fundamental notions of constructor theory within the canvas of process theory. Specifically, we exploit the functorial interplay between the symmetric monoidal structure of the category of sets and relations, where the abstract tasks live, and that of symmetric monoidal categories from physics, where concrete processes can be found to implement said tasks. Through this, we answer the question of how constructor theory relates to the broader body of process-theoretic literature, and provide the impetus for future collaborative work between the fields.

1 Introduction

Constructor theory [17, 18, 30] is a metatheoretic approach that seeks to characterise concrete theories of physics and information in terms of the *possibility* and *impossibility* of *tasks*, which are *transformations* between *systems*. Transformations may require auxiliary inputs other than the system to be transformed: the task of turning black shoes into white shoes may require a stock of white paint as an auxiliary input in addition to the black shoes themselves. Tasks transform *states* of systems into other states, and *attributes* of systems—such as the blackness of a shoe—into other attributes. In our universe, we will eventually run out of white paint for this task, but if we had a mathematically ideal paintbrush with infinite white paint, we could reuse it for as many instances of the task as we'd like; such non-exhaustible auxiliary catalysts for tasks are called *constructors*. A task is *possible* when it is partnered with a constructor that allows the task to be performed arbitrarily many times, and the task is *impossible* otherwise. Though constructors and tasks are abstract, they provide explanatory value; constructor theory seeks to characterise physical theories in terms of what tasks are possible. As a metatheory, constructor theory is implementation-agnostic, and one can choose whatever formal system of mathematics they like as a concrete language to interpret the italic terms above.

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 137–151, doi:10.4204/EPTCS.397.9

Process theories provide one such mathematically formal language, one particularly well-suited to describe the composition of processes in space-time. Moreover, process theories are expressed in terms of string diagrams, which are an aesthetic, intuitive, flexible, and rigorous metalinguistic syntax, empowering the modeller by allowing them to operate at a level of abstraction of their choice. This means that the same abstract diagrams provide a common syntactic foundation for fields as disparate as linear and affine algebra [6, 5], first order logic [24], electrical circuits [4], digital circuits [21], database operations [26, 35], spatial relations [34], game theory [25], petri nets [3], hypergraphs [2], probability theory [8, 20], causal reasoning [28], machine learning [16], and quantum theory [15, 14], to name just a few.

In this short paper, we provide a formal interpretation of constructor-theoretic terminology and ideas within the string-diagrammatic setting of process theories, with the intent to build a bridge between the two communities. We caution against the view that constructor theory is "just" a class of process theories, in the same sense as it would be misguided to claim that prime numbers are "just" integers. Process theory merely provides a rigorous mathematical language for constructor theorists to tell their stories.

For the process theorists in our audience, we wish to stress that the pedagogical mathematical presentation of this paper is for the sake of constructor theorists who might be approaching our field for the first time. Regardless, we offer you a Rosetta stone for constructor theory within what we understand to be the *de rigueur* mathematics of the field, transliterated into diagrams with as few embellishments and interpretational choices as possible. For the constructor theorists in our audience, we extend a warm invitation to join the process-theoretic community: to the best of our knowledge, this is the most attractive and general formal arena available within which to explore the ramifications of constructor theory.

2 Conceivable Tasks

Constructor theory is concerned with the study of physical theories in terms of the question "which *tasks* are performable within this physical theory?": there is an abstract notion of *conceivable* tasks and a concrete notion of *possible* tasks. In seminal work by Deutsch [17], it was remarked that, in full generality, the only real requirement on conceivable tasks is arbitrary composability in sequence and in parallel, i.e. that they form a symmetric monoidal category (SMC). [1] Back then, however, the same author made a specific choice to model tasks as relations between sets: constructor theory literature has stuck by this choice ever since, and so will we.

Remark 2.1. In this work, we take all monoidal categories to be *strict*, and in particular we assume that objects $\mathrm{obj}(\mathbf{D})$ in a monoidal category $\mathbf{D}$ form a strict monoid. In the case of the SMCs **Rel** and **Set**, considered in Definition 2.2 below, this implies a choice of singleton set $1 := \{*\}$ to act as a strict unit for the Cartesian product:

$$X \times 1 = X = 1 \times X$$

This also affords us the freedom to write triples (and other tuples) without having to care about nesting:

$$X \times Y \times Z = \{(x,y,z) \mid x \in X, y \in Y, z \in Z\}$$

[1] It is possible that Deutsch meant for substrates to have an individual identity as physical systems, rather than just a "type": that is, it is possible that Deutsch would prefer for "this qubit" and "that qubit" to be modelled by different—albeit isomorphic—objects in a process theory. In this case, it would make no sense to consider parallel compositions of tasks involving the "same" physical system, and partially-monoidal categories as defined in [22] would be preferable as a process-theoretical universe.

Note that strictness does not extend to symmetry isomorphisms: we have that $X \times Y \cong Y \times X$, but this doesn't mean that $X \times Y = Y \times X$. As a consequence, the monoid formed by objects in a strict SMC is not generally commutative.

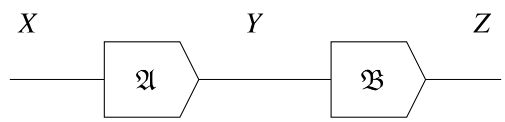
We take the theory of *conceivable tasks* to be **Rel**, the †-SMC of sets and relations. We write $\mathfrak{A} : X \to Y$ for a task/relation $\mathfrak{A} \subseteq X \times Y$, where the set X labels legitimate input states for the task and the set Y labels legitimate output states. To help distinguish between pairs/tuples of elements in a Cartesian product and pairs of domain/codomain elements in a relation, we reserve pair/tuple notation for the former and adopt *maplet* notation for the latter:

$$x \mapsto y \;\; :\equiv \;\; (x,y) \qquad\qquad x \overset{\mathfrak{A}}{\mapsto} y \;\; :\equiv \;\; (x,y) \in \mathfrak{A}$$

We omit $\mathfrak{A}$ from $\overset{\mathfrak{A}}{\mapsto}$ when clear from context. The *sequential composition* $\mathfrak{B} \circ \mathfrak{A} : X \to Z$ of task $\mathfrak{B} : Y \to Z$ after $\mathfrak{A} : X \to Y$ is defined as follows:

$$\mathfrak{B} \circ \mathfrak{A} := \left\{ x \mapsto z \;\middle|\; \exists y \in Y . x \overset{\mathfrak{A}}{\mapsto} y \text{ and } y \overset{\mathfrak{B}}{\mapsto} z \right\}$$

Sequential composition in diagrammatic language:



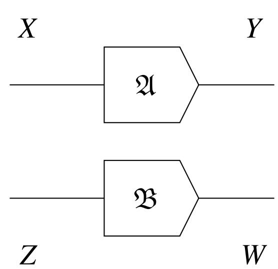
Composite sets of states are obtained by Cartesian product $X \times Y$:

$$X \times Y := \{(x,y) \mid x \in X \text{ and } y \in Y\}$$

The **parallel composition** $\mathfrak{A} \times \mathfrak{B} : X \times Z \to Y \times W$ of tasks $\mathfrak{A} : X \to Y$ and $\mathfrak{B} : Z \to W$ is defined as follows:

$$\mathfrak{A} \times \mathfrak{B} := \left\{ (x,z) \mapsto (y,w) \;\middle|\; x \overset{\mathfrak{A}}{\mapsto} y \text{ and } z \overset{\mathfrak{B}}{\mapsto} w \right\}$$

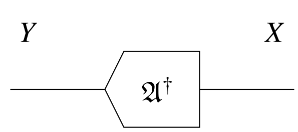
Parallel composition in diagrammatic language:



The **transpose** $\mathfrak{A}^\dagger : Y \to X$ of a task $\mathfrak{A} : X \to Y$ is defined as follows:

$$\mathfrak{A}^\dagger := \left\{ y \mapsto x \;\middle|\; x \overset{\mathfrak{A}}{\mapsto} y \right\}$$

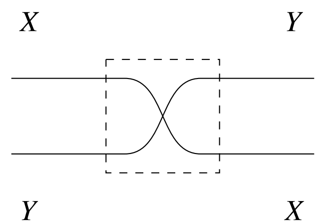
Transposition in diagrammatic language:



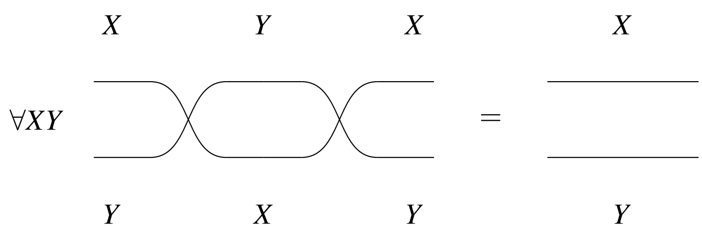
Finally, there are **symmetry isomorphisms** (aka **swaps**) $\sigma_{X,Y} : X \times Y \xrightarrow{\cong} Y \times X$:

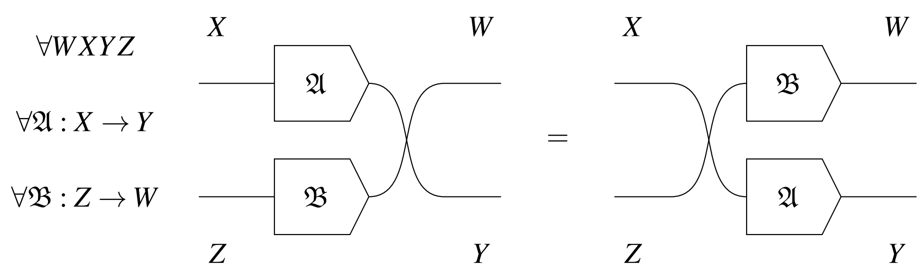
$$\sigma_{X,Y} := \{(x,y) \mapsto (y,x) \mid x \in X \text{ and } y \in Y\}$$

Symmetry isomorphisms in diagrammatic language:



The symmetry isomorphisms are a structural feature of the category, making it possible to compose relations into acyclic networks, where outputs of relations can be connected to inputs of other relations. This is made possible by the following properties of the symmetry isomorphisms:



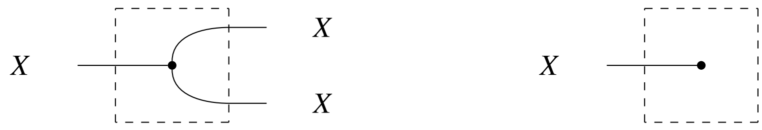


Remark 2.2. The sets X and Y are allowed to be distinct, for sake of generality. Asking that they are always equal is equivalent to restricting the theory of conceivable tasks to be the †-SMC **EndoRel** of sets and endo-relations $R : X \to X$, which is a sub-†-SMC of **Rel**.

If we restrict our attention to the *total deterministic* relations in **Rel**, we obtain the sub-SMC **Set** of sets and functions between them. Functions are closed under acyclic network composition (sequential and parallel, including the usage of symmetry isomorphisms), but not under transpose. Important examples of functions are the **copy map** $\delta_X : X \to X \times X$ and **discarding map** $\varepsilon_X : X \mapsto 1$ on a set X:

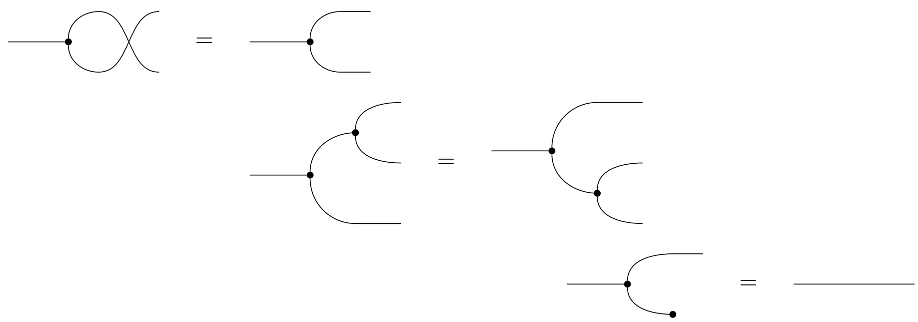
$$\delta_X := \{x \mapsto (x,x) \mid x \in X\}$$
$$\varepsilon_X := \{x \mapsto * \mid x \in X\}$$

Copy and delete maps in diagrammatic language:



δ_X ε_X

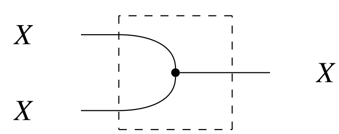
Copies are indistinguishable under swaps and repeated copies, and deleting a copy results in the identity:



The transposes of the copy and discarding map are not functions. The transpose $\delta_X^\dagger : X \times X \to X$ is the **match map**, a partial function which returns the common value of its inputs when they're equal and is otherwise undefined:

$$\delta_X^\dagger := \{(x,x) \mapsto x \mid x \in X\}$$

Match map in diagrammatic language:

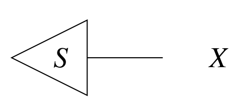


$\delta_X^\dagger$

Relations $S : 1 \to X$, such as the transpose $\varepsilon_X^\dagger : 1 \to X$ of the discarding map, can be identified with all possible **attributes** of states in X, i.e. with all possible subsets $S \subseteq X$:

$$S \cong \{* \mapsto x \mid x \in S\}$$

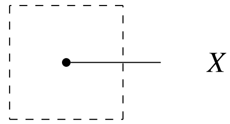
States and attributes have the same notation in diagrammatic language, since states $x \in X$ can be identified with singleton subsets $\{x\} \subseteq X$:



The transpose $\varepsilon_X^\dagger : 1 \to X$ of the discarding map is the **trivial attribute**, corresponding to subset $X \subseteq X$:

$$\eta_X := \varepsilon_X^\dagger = \{* \mapsto x \mid x \in X\}$$

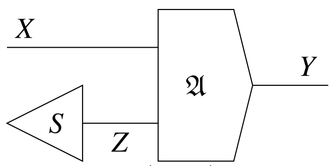
Trivial attribute in diagrammatic language:



η_X

Attributes can be used to condition tasks to specific input states.

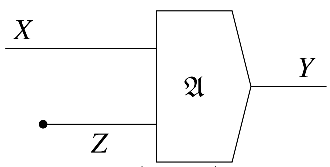
Definition 2.3. Let $\mathfrak{A} : X \times Z \to Y$ be a task and let $S \subseteq Z$ be an attribute on states in Z. The **pre-conditioned task** is defined to be the task obtained by forgetting all information about the Z input of $\mathfrak{A}$ other than the fact that the input state has attribute S:



$$= \left\{ x \mapsto y \,\middle|\, \exists z \in S.\, (x,z) \overset{\mathfrak{A}}{\mapsto} y \right\}$$

$\mathfrak{A} \circ (\mathrm{id}_X \times S)$

As a special case, we can discard the Z input entirely, by pre-conditioning against the trivial attribute η_Z:



$$= \left\{ x \mapsto y \,\middle|\, \exists z \in Z.\, (x,z) \overset{\mathfrak{A}}{\mapsto} y \right\}$$

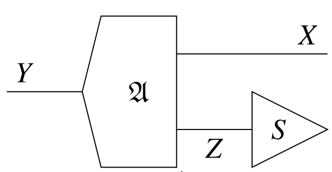
$\mathfrak{A} \circ (\mathrm{id}_X \times \eta_Z)$

The object 1 is *terminal* in **Set**: there is a unique function $\varepsilon_X : X \to 1$ for any set X. However, it is not terminal in **Rel**: the relations $X \to 1$ are exactly the transposes $S^\dagger : X \to 1$ of the attributes $S : 1 \to X$. Explicitly, they are the constant partial functions with the attribute S as their domain:

$$S^\dagger := \{x \mapsto * \mid x \in S\}$$

The transposes of attributes are **tests**, which can be used to condition tasks to specific output states.

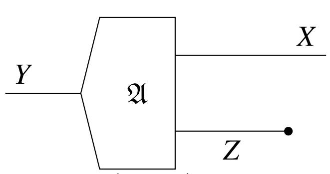
Definition 2.4. Let $\mathfrak{A} : X \to Y \times Z$ be a task and let $S \subseteq Z$ be an attribute on states in Z. The **post-conditioned task** is defined to be the task obtained by forgetting all information about the Z output of $\mathfrak{A}$ other than the fact that the output state has attribute S:



$$= \left\{ x \mapsto y \,\middle|\, \exists z \in S. x \overset{\mathfrak{A}}{\mapsto} (y,z) \right\}$$

$(\mathrm{id}_X \times S^\dagger) \circ \mathfrak{A}$

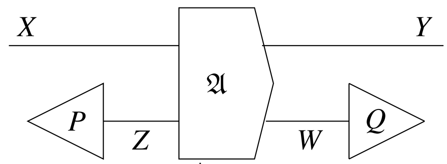
As a special case, we can discard the Z output of the task entirely, by post-conditioning against the trivial attribute on Z:



$$= \left\{ x \mapsto y \,\middle|\, \exists z \in Z. x \overset{\mathfrak{A}}{\mapsto} (y,z) \right\}$$

$(\mathrm{id}_X \times \varepsilon_Z) \circ \mathfrak{A}$

Remark 2.5. We can simultaneously pre-condition a task $\mathfrak{A} : X \times Z \to Y \times W$ against an attribute $P \subseteq Z$ and post-condition it against an attribute $Q \subseteq W$:



$(\mathrm{id}_X \times Q^\dagger) \circ \mathfrak{A} \circ (\mathrm{id}_X \times P)$

$$= \left\{ x \mapsto y \;\middle|\; \exists p \in P, q \in Q.\, (x,p) \overset{\mathfrak{A}}{\mapsto} (y,q) \right\}$$

3 Possible Tasks

Conceivable tasks are a theory-independent concept: they provide a formal universe within which to formulate principles and derive constraints. On the other hand, possible tasks are theory-dependent, induced by the constructors physically available to implement them. In order to determine which tasks are possible, we need to make a choice of substrates within a theory of processes.

Definition 3.1. A **choice of substrates** $(\mathbf{C}, \Sigma, \Gamma)$ comprises:

1. A reference **theory of processes**, in the form of a strict SMC $\mathbf{C} = (\mathrm{obj}(\mathbf{C}), \otimes, I)$. For example, this could be the theory of finite-dimensional quantum systems and unitary transformations.
2. A choice of **substrates**, in the form of a subset $\Sigma \subseteq \mathrm{obj}(\mathbf{C})$ of systems in the theory of processes.
3. A choice of **sets of substrate states**, in the form of a family $\Gamma = (\Gamma_{\mathtt{H}})_{\mathtt{H} \in \Sigma}$ where $\Gamma_{\mathtt{H}} \subseteq \mathrm{states}_{\mathbf{C}}(\mathtt{H})$ is a set of states in $\mathbf{C}$ for each substrate $\mathtt{H} \in \Sigma$.

We require that the choice of substrates be closed under parallel composition: $I \in \Sigma$ and $\mathtt{H} \otimes \mathtt{K} \in \Sigma$ for all $\mathtt{H}, \mathtt{K} \in \Sigma$. We further require that the set of substrate states respects parallel composition of substrates: $\Gamma_I = 1$ and $\Gamma_{\mathtt{H} \otimes \mathtt{K}} = \Gamma_{\mathtt{H}} \times \Gamma_{\mathtt{K}}$ for all $\mathtt{H}, \mathtt{K} \in \Sigma$.

Given two substrates $\mathtt{H}, \mathtt{K} \in \Sigma$, we consider tasks $\Gamma_{\mathtt{H}} \to \Gamma_{\mathtt{K}}$ and ask which ones are *possible* within the given theory of processes: in short, a task is possible when there is a *constructor* which acting as a catalyst enables the task to be performed. Expanding on this, we come to the following definitions.

Definition 3.2. Let $(\mathbf{C}, \Sigma, \Gamma)$ be a choice of substrates and consider two substrates $\mathtt{H}, \mathtt{K} \in \Sigma$. A process $f : \mathtt{H} \to \mathtt{K}$ is **task-inducing** if it maps states in $\Gamma_{\mathtt{H}}$ to states in $\Gamma_{\mathtt{K}}$:

$$\forall \rho \in \Gamma_{\mathtt{H}}.\, f(\rho) \in \Gamma_{\mathtt{K}}$$

We write $\lfloor f \rfloor$ for the task **induced by** f:

$$\lfloor f \rfloor := \{\rho \mapsto f(\rho) \mid \rho \in \Gamma_{\mathtt{H}}\}$$

Definition 3.3. Let $(\mathbf{C}, \Sigma, \Gamma)$ be a choice of substrates and consider a task $\mathfrak{A} : \Gamma_{\mathtt{H}} \to \Gamma_{\mathtt{K}}$. We say that $\mathfrak{A}$ is **possible** if there are:

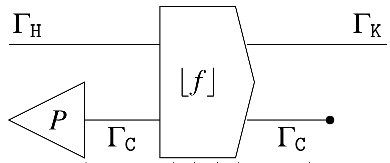
(i) a substrate $\mathtt{C}$ (acting as a **constructor** for the task)

(ii) an attribute $P \subseteq \Gamma_{\mathtt{C}}$ (singling out the relevant constructor states)

(iii) a task-inducing process $f : \mathtt{H} \otimes \mathtt{C} \to \mathtt{K} \otimes \mathtt{C}$ (actually performing the task)

such that the following two conditions are satisfied:

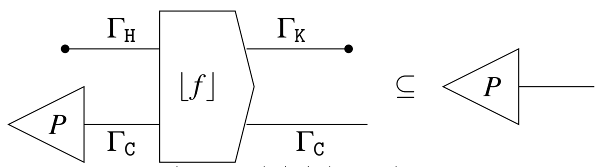
1. Task $\mathfrak{A}$ is obtained from the induced task $\lfloor f \rfloor$ by requiring that the input constructor state has attribute P and discarding the constructor output:

$\mathfrak{A} =$ 

$$(\mathrm{id}_{\Gamma_{\mathtt{K}}} \times \varepsilon_{\Gamma_{\mathtt{C}}}) \circ \lfloor f \rfloor \circ (\mathrm{id}_{\Gamma_{\mathtt{H}}} \times P)$$

$$= \left\{ \rho \mapsto \rho' \;\middle|\; \exists \gamma \in P, \gamma' \in \Gamma_{\mathtt{C}}.\, f(\rho \otimes \gamma) = \rho' \otimes \gamma' \right\}$$

2. The attribute P is preserved by the induced task $\lfloor f \rfloor$. While a particular constructor state $\gamma \in P$ may be modified to become γ' by the underlying process of the induced task $\lfloor f \rfloor$, γ' remains a constructor state for the same induced task $\lfloor f \rfloor$, i.e. $\gamma' \in P$. In **Rel**, this constraint is equivalently expressed as the induced task $\lfloor f \rfloor$ sending the set of constructors P to a subset of itself, regardless of the input and output on the substrates $\mathtt{H}, \mathtt{K}$:



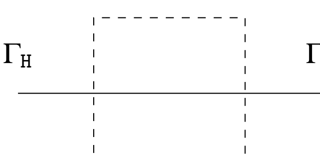
$$(\varepsilon_{\Gamma_{\mathtt{K}}} \times \mathrm{id}_{\Gamma_{\mathtt{C}}}) \circ \lfloor f \rfloor \circ (\eta_{\Gamma_{\mathtt{H}}} \times P) \subseteq P$$

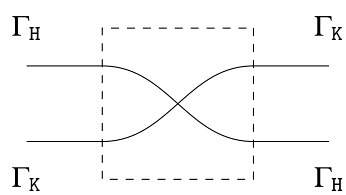
We write $(\mathbf{C}, \Sigma, \Gamma)^{\checkmark}$ for the set of possible tasks under the given choice of substrates.

The main result of this section is that possible tasks for a choice of substrate form a sub-SMC of **Rel**, i.e. that they are closed under composition in arbitrary (acyclic) networks.

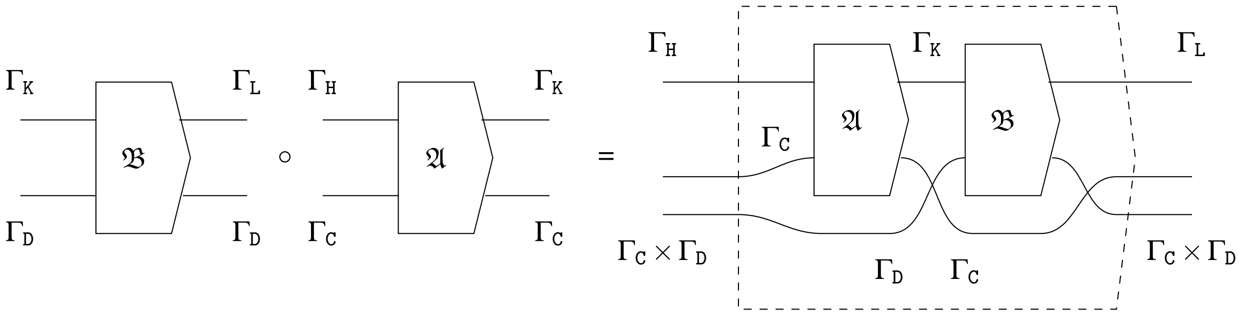
Proposition 3.4. The possible tasks $(\mathbf{C}, \Sigma, \Gamma)^{\checkmark}$ for a given choice of substrates form a sub-SMC of **Rel**.

Proof. Write $\mathbf{C} = (\mathrm{obj}(\mathbf{C}), \otimes, I)$. The identity tasks and swap tasks for all systems are made possible by the identity and symmetry isomorphisms of $\mathbf{C}$, with trivial constructor $\mathtt{C} := I$:

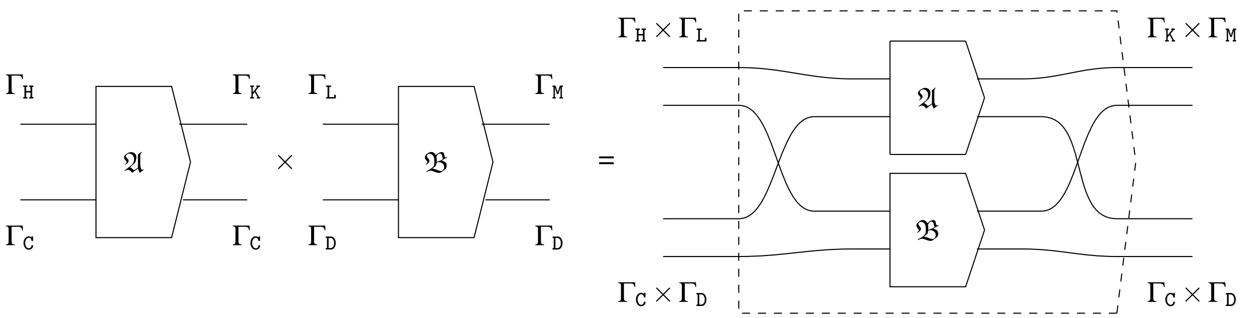




The sequential composition $\mathfrak{B} \circ \mathfrak{A}$ of possible tasks $\mathfrak{A}$ and $\mathfrak{B}$, with constructors C and D respectively, is possible with constructor $C \otimes D$: [2]



The parallel composition $\mathfrak{B} \times \mathfrak{A}$ of possible tasks $\mathfrak{A}$ and $\mathfrak{B}$, with constructors C and D respectively, is possible with constructor $C \otimes D$: [3]



This completes our proof. □

4 Attributes as states

More modern perspectives in constructor theory argue that tasks should be defined on the attributes of a substrate, rather than on the underlying states. This captures the idea that the abstract specification of (possible) tasks—the basis upon which constructor theorists judge other theories of physics—should be based on the observable "macrostates" of a physical system (attributes/subsets of a set), rather than on the unobserved "microstates" which constitute them (states/elements of a set). In this section, we show how the attribute-based perspective can be derived from the state-based perspective, in a compositionally sound way, by performing a suitable coarse-graining.

To start with, we define a notion of "coarse-graining" for tasks, moving from tasks defined on states (the "microstates", to stick to the thermodynamical metaphor) to tasks defined on attributes (the "macrostates", using the same metaphor). We allow for the attributes involved to have non-trivial overlap—that is, we don't ask for them to form a partition—but we disallow nesting $S \subset T$ of different attributes; formally, we require for the set of attributes involved to form an "antichain" in the inclusion order $\subseteq$.

[2]This step of the proof becomes more complicated if constructors are forced to have individual identities (i.e. in a partially monoidal category) and the same constructor must be reused by task $\mathfrak{B}$ after being used by task $\mathfrak{A}$. We leave the handling of this more sophisticated process-theoretic interpretation of constructor theory to future work.

[3]This step of the proof becomes more complicated if constructors are forced to have individual identities and the same constructor must be simultaneously used by task $\mathfrak{B}$ and task $\mathfrak{A}$. We leave the handling of this more sophisticated process-theoretic interpretation of constructor theory to future work.

Definition 4.1. Let X be a set. A set $\bar{X} \subseteq \mathscr{P}(X)$ of attributes on X is an **antichain** if no two attributes are nested into each other:

$$\forall S, T \in \bar{X}. S \subseteq T \Rightarrow S = T$$

Having fixed a choice of attributes $\bar{X}$ on X and $\bar{Y}$ on Y, any task $\mathfrak{A} : X \to Y$ induces a "coarse-grained task" on the sets of attributes, as follows: for attributes $S \in \bar{X}$ and $T \in \bar{Y}$, we say that $S \mapsto T$ in the coarse-grained task if whenever an input state $x \in X$ has attribute S, i.e. whenever $x \in S$, then at least one of the possible outputs states $\left\{ y \in Y \,\middle|\, x \overset{\mathfrak{A}}{\mapsto} y \right\}$ has attribute T, i.e. $\exists y \in T. x \overset{\mathfrak{A}}{\mapsto} y$. Put it another way, $S \mapsto T$ in the coarse-grained task means that the output of task $\mathfrak{A}$ *can have* attribute T whenever the input *has* attribute S.

Definition 4.2. Let $\mathfrak{A} : X \to Y$ be a task. Let $\bar{X} \subseteq \mathscr{P}(X)$ and $\bar{Y} \subseteq \mathscr{P}(Y)$ be sets of attributes of X and Y respectively. Then the **coarse-grained task** $\mathfrak{A}|_{\bar{X}}^{\bar{Y}} : \bar{X} \to \bar{Y}$ is defined as follows:

$$\mathfrak{A}|_{\bar{X}}^{\bar{Y}} := \left\{ S \mapsto T \,\middle|\, S \in \bar{X}, T \in \bar{Y}, S \subseteq \mathfrak{A}^{\dagger} \circ T \right\}$$

We conclude this section with three results, piecing the coarse-graining story together. Firstly, we prove that given any process theory of tasks—including, amongst many others, the theory of all conceivable tasks and all theories of possible tasks—the coarse-grainings of the tasks can themselves be arranged into a process theory. This shows that tasks defined on attributes are just as compositionally sound as those defined on states. Secondly, we remark how the original ordinary tasks, defined on states, can be compositionally embedded into the universe of coarse-grained tasks, proving that the latter are a sound generalisation of the former. Finally, we remark that coarse-grained tasks can be embedded back into the universe of ordinary tasks, proving that ordinary tasks are as expressive as coarse-grained ones.

Proposition 4.3. Let $\mathscr{C}$ be a sub-SMC of **Rel**, i.e. a collection of systems and tasks closed under parallel and sequential composition. The following defines a SMC $\overline{\mathscr{C}}$, which we refer to as the theory of **coarse-grained tasks** associated to $\mathscr{C}$:

- objects are all possible antichains of attributes for all possible sets of states:

$$\mathrm{obj}\left(\overline{\mathscr{C}}\right) := \bigcup_{X \in \mathrm{obj}(\mathscr{C})} \{\bar{X} \subseteq \mathscr{P}(X) \mid \bar{X} \text{ antichain}\}$$

- morphisms $\bar{X} \to \bar{Y}$ in $\bar{\mathscr{C}}$ are coarse-grained tasks corresponding to tasks $X \to Y$:

$$\overline{\mathscr{C}}(\bar{X}, \bar{Y}) := \left\{ \mathfrak{A}|_{\bar{X}}^{\bar{Y}} \,\middle|\, \mathfrak{A} : X \xrightarrow{\mathscr{C}} Y \right\}$$

- sequential composition $\circ$ is inherited from $\mathscr{C}$
- parallel composition $\boxtimes$ on objects is defined as:

$$\bar{X} \boxtimes \bar{Y} := \{S \times T \mid S \in \bar{X}, T \in \bar{Y}\}$$

- parallel composition $\boxtimes$ on morphisms arises by coarse-graining from that of $\mathscr{C}$:

$$\mathfrak{A}|_{\bar{X}}^{\bar{Y}} \boxtimes \mathfrak{B}|_{\bar{Z}}^{\bar{W}} := (\mathfrak{A} \times \mathfrak{B})\,|_{\bar{X} \boxtimes \bar{Z}}^{\bar{Y} \boxtimes \bar{W}}$$

- identity and symmetry isomorphisms arise by coarse-graining from those of $\mathscr{C}$:

$$\mathrm{id}_X|_{\bar{X}}^{\bar{X}} = \mathrm{id}_{\bar{X}} \qquad\qquad \sigma_{X,Y}|_{\bar{X} \boxtimes \bar{Y}}^{\bar{Y} \boxtimes \bar{X}} = \sigma_{\bar{X}, \bar{Y}}$$

In particular, morphisms are well-defined, i.e. whenever $\bar{X} = \bar{X}'$ and $\bar{Y} = \bar{Y}'$ we have:

$$\left\{\bar{\mathfrak{A}} \,\middle|\, \mathfrak{A} : X \to Y\right\} = \left\{\bar{\mathfrak{A}} \,\middle|\, \mathfrak{A} : X' \to Y'\right\}$$

Proof. Objects are clearly well-defined, but well-definition of morphisms requires proof. Let X and Y be sets, let $\bar{X} \subseteq \mathscr{P}(X)$ and $\bar{Y} \subseteq \mathscr{P}(Y)$ be antichains. It suffices to show the following for $X' := \bigcup \bar{X} \subseteq X$ and $Y' := \bigcup \bar{Y} \subseteq Y$:

$$\left\{\mathfrak{A}|_{\bar{X}}^{\bar{Y}} \,\middle|\, \mathfrak{A} : X \to Y\right\} = \left\{\mathfrak{A}|_{\bar{X}}^{\bar{Y}} \,\middle|\, \mathfrak{A} : X' \to Y'\right\}$$

Write $\pi_{X'} := \{x \mapsto x | x \in X'\} : X \to X$ and $\pi_{Y'} := \{y \mapsto y | y \in Y'\} : Y \to Y$. If $S \in \bar{X}$ and $T \in \bar{Y}$, then $S \subseteq X'$ and $T \subseteq Y'$, and hence:

$$S \subseteq \mathfrak{A}^\dagger \circ T \;\Leftrightarrow\; S \subseteq \pi_{X'} \circ \mathfrak{A}^\dagger \circ T \;\Leftrightarrow\; S \subseteq \pi_{X'} \circ \mathfrak{A}^\dagger \circ \pi_{Y'} \circ T$$

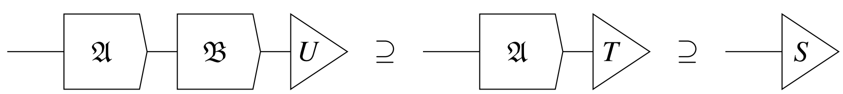
Observing that $\pi_{Y'} \circ \mathfrak{A} \circ \pi_{X'}$ is a task $X' \to Y'$ completes the proof that morphisms are well-defined. For identities, we want to show that $\mathrm{id}_X|_{\bar{X}}^{\bar{X}} = \mathrm{id}_{\bar{X}}$, and this is exactly the definition of $\bar{X}$ being an antichain:

$$\mathrm{id}_X|_{\bar{X}}^{\bar{X}} = \mathrm{id}_{\bar{X}} \Leftrightarrow [S \subseteq T \Rightarrow S = T]$$

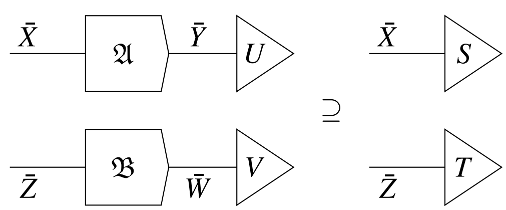
For symmetry isomorphisms, we want to show that $\sigma_{X,Y}|_{\bar{X}\times\bar{Y}}^{\bar{X}\times\bar{Y}} = \sigma_{\bar{X},\bar{Y}}$, and this again follows from the antichain requirement:

$$\begin{aligned}\sigma_{X,Y}|_{\bar{X}\boxtimes\bar{Y}}^{\bar{X}\boxtimes\bar{Y}} = \sigma_{\bar{X},\bar{Y}} &\Leftrightarrow \Big[\Big(S \times T \subseteq \sigma_{X,Y}^\dagger \circ (T' \times S')\Big) \Rightarrow \big(S = S' \text{ and } T = T'\big)\Big] \\ &\Leftrightarrow \big[\big(S \subseteq S' \text{ and } T \subseteq T'\big) \Rightarrow \big(S = S' \text{ and } T = T'\big)\big]\end{aligned}$$

For sequential composition to be well-defined, we need to show that $S \subseteq \mathfrak{A}^\dagger \circ T$ and $T \subseteq \mathfrak{B}^\dagger \circ U$ imply $S \subseteq (\mathfrak{B} \circ \mathfrak{A})^\dagger \circ U$:



For parallel composition to be well-defined, we need to show that $S \subseteq \mathfrak{A}^\dagger \circ U$ and $T \subseteq \mathfrak{B}^\dagger \circ V$ imply $S \times T \subseteq (\mathfrak{A} \boxtimes \mathfrak{B})^\dagger \circ (U \times V)$:



The remaining checks are all straightforward, on similar lines. □

Remark 4.4. Any sub-SMC $\mathscr{C}$ of **Rel** embeds into the associated theory of coarse-grained tasks $\overline{\mathscr{C}}$. The embedding is the functor—faithful and injective on objects—defined by sending each set to the set of its singleton subsets:

$$\begin{aligned}F(X) &:= \{\{x\} \mid x \in X\} \\ F(\mathfrak{A} : X \to Y) &:= \mathfrak{A}|_{\bar{X}}^{\bar{Y}} = \left\{\{x\} \mapsto \{y\} \,\middle|\, x \overset{\mathfrak{A}}{\mapsto} y\right\}\end{aligned}$$

It is straightforward to check that the mapping defined above is a strict monoidal functor, i.e. that it preserves both sequential and parallel composition exactly (as well as identities, and symmetry isomorphisms, in this case).

Remark 4.5. Let $\mathscr{C}$ be a sub-SMC of **Rel**. The associated theory of coarse-grained tasks $\overline{\mathscr{C}}$ embeds back into **Rel**, via the identity functor:

$$F(\bar{X}) := \bar{X} \qquad\qquad F\left(\mathfrak{A}|_{\bar{X}}^{\bar{Y}}\right) := \mathfrak{A}|_{\bar{X}}^{\bar{Y}}$$

The functor is strict monoidal when restricted to (the embedding of) $\mathscr{C}$ (into $\overline{\mathscr{C}}$). It is not strict (or strong) monoidal in general, because the tensor product on sets of attributes is not the same as the tensor product on sets of states:

$$\bar{X} \boxtimes \bar{Y} := \{S \times T \mid S \in \bar{X},\, T \in \bar{Y}\} \neq \{(S,T) \mid S \in \bar{X},\, T \in \bar{Y}\} = \bar{X} \times \bar{Y}$$

It is, however, lax monoidal, with the following structure morphisms:

$$[(S,T) \mapsto S \times T] : \bar{X} \times \bar{Y} \to \bar{X} \boxtimes \bar{Y} \qquad\qquad [\{*\} \mapsto *] : \bar{1} \to 1$$

To see this, it suffices to observe that not only do $S \subseteq \mathfrak{A}^\dagger \circ U$ and $T \subseteq \mathfrak{B}^\dagger \circ V$ imply $S \times T \subseteq (\mathfrak{A} \boxtimes \mathfrak{B})^\dagger \circ (U \times V)$, but also $S \times T \subseteq (\mathfrak{A} \boxtimes \mathfrak{B})^\dagger \circ (U \times V)$ implies both $S \subseteq \mathfrak{A}^\dagger \circ U$ and $T \subseteq \mathfrak{B}^\dagger \circ V$.

5 Conclusion and historical remarks

We have given categorical semantics for constructor theory in its most general form, interpreting to the best of our ability the desired mathematical foundations both set out in Deutsch's original paper [17] and expressed to us by current practitioners. We remark, without further comment, that the diagrammatic syntax we have used to formally incarnate constructor theory is also interpretable in other symmetric monoidal categories. A long form presentation of the same content with worked examples from the constructor theory literature is in preparation.

We close with two historical case studies intended to inform constructor theorists of the topically-relevant history of process theories as applied to quantum theory, and to encourage the pursuit of the possible-impossible dichotomy by illustrating some of the fruitful outcomes that may result.

Process theories arose from counterfactual reasoning. *Possibility*, read as what *could* happen, is at the heart of constructor theory: here, constructor theory and process theories share a lineage of counterfactual reasoning, tracing back to Aristotle's distinction between "actual" and "potential". One ancestor of process theories along this lineage is the Geneva school of quantum logic [32, 29], which defined the properties of physical systems in terms of experiments that could be performed [31], resulting in the linearity of physical processes [19] due to an adjunction between cause and consequence (cf. weakest precondition semantics in computer science [27]). This led to the development of a process-theoretic framework for quantum theory, which encoded the structural consequences of an adjunction between causes and consequences in terms of a quantaloid [11]. The underlying structure of spaces (= quantum logics) was induced at the level of processes, and efforts were made to cast the composition of systems in those terms through process-state duality [9]. However, the current success of process theories relies on dumping quantum logics and replacing them with specially chosen processes (cf. cups and caps [1]). A process theory, when formulated as a concrete symmetric monoidal category, is about possible and impossible processes that obey the axioms of the corresponding category. Reconstructions of quantum theory in terms of process theories turn these categorical axioms into physical postulates that are considered more reasonable by some [23, 33].

Quantum from no-cloning. In constructor theory, the cut between possible and impossible tasks is used to define theories, and it has been suggested that the impossibility to clone should yield quantum theory, at least in a broad sense. In categorical quantum mechanics [1, 15, 14], dating back to at least 2006, classicality was indeed defined by the ability to clone [13]: this has resulted in the development of spiders [12] and the ZX-calculus [10], now a prominent formalism in quantum foundations, quantum computation, and general education on quantum theory.

Acknowledgements

With many thanks to **Maria Violaris** and **Anicet Tibau Vidal** for their clarity and patience when presenting constructor theory at the Wolfson quantum foundations discussion, and to **Nicola Pinzani** for his insights in conversation with one of the authors.

References

[1] S. Abramsky & B. Coecke (2004): *A categorical semantics of quantum protocols*. In: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pp. 415–425, doi:10.5555/1018438.1021878. arXiv:quant-ph/0402130.

[2] Mario Alvarez-Picallo, Dan Ghica, David Sprunger & Fabio Zanasi (2022): *Rewriting for Monoidal Closed Categories: 7th International Conference on Formal Structures for Computation and Deduction, FSCD 2022*. *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, doi:10.4230/LIPIcs.FSCD.2022.29. Available at `http://www.scopus.com/inward/record.url?scp=85133671634&partnerID=8YFLogxK`. Publisher: Schloss Dagstuhl.

[3] John C. Baez & Jade Master (2020): *Open Petri Nets*. *Mathematical Structures in Computer Science* 30(3), pp. 314–341, doi:10.1017/S0960129520000043. Available at `http://arxiv.org/abs/1808.05415`. ArXiv:1808.05415 [cs, math].

[4] Guillaume Boisseau & Paweł Sobociński (2022): *String Diagrammatic Electrical Circuit Theory*. *Electronic Proceedings in Theoretical Computer Science* 372, pp. 178–191, doi:10.4204/EPTCS.372.13. Available at `http://arxiv.org/abs/2106.07763`. ArXiv:2106.07763 [cs].

[5] 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)*, pp. 1–12, doi:10.1109/LICS.2019.8785877.

[6] Filippo Bonchi, Pawel Sobocinski & Fabio Zanasi (2017): *Interacting Hopf Algebras*. *Journal of Pure and Applied Algebra* 221(1), pp. 144–184, doi:10.1016/j.jpaa.2016.06.002. Available at `http://arxiv.org/abs/1403.7048`. ArXiv:1403.7048 [cs, math].

[7] Filippo Bonchi, Pawel Sobociński & Fabio Zanasi (2014): *A Categorical Semantics of Signal Flow Graphs*. CONCUR 2014 - Concurrency Theory - 25th International Conference, doi:10.1007/978-3-662-44584-6_30. Available at `https://hal.science/hal-02134182`.

[8] Kenta Cho & Bart Jacobs (2019): *Disintegration and Bayesian Inversion via String Diagrams*. *Mathematical Structures in Computer Science* 29(7), pp. 938–971, doi:10.1017/S0960129518000488. Available at `http://arxiv.org/abs/1709.00322`. ArXiv:1709.00322 [cs].

[9] B. Coecke (2000): *Structural characterization of compoundness*. International Journal of Theoretical Physics 39, pp. 585–594, doi:10.1023/A:1003677418744.

[10] B. Coecke & R. Duncan (2011): *Interacting quantum observables: categorical algebra and diagrammatics*. New Journal of Physics 13, p. 043016, doi:10.1088/1367-2630/13/4/043016. arXiv:quant-ph/09064725.

[11] B. Coecke, D.J. Moore & I. Stubbe (2001): *Quantaloids describing causation and propagation of physical properties*. Foundations of Physics Letters 14, pp. 133–146, doi:10.1023/A:1012377520222. ArXiv:quant-ph/0009100.

[12] B. Coecke, É. O. Paquette & D. Pavlović (2010): *Classical and quantum structuralism*. In S. Gay & I. Mackie, editors: *Semantic Techniques in Quantum Computation*, Cambridge University Press, pp. 29–69, doi:10.1017/CBO9781139193313.003. arXiv:0904.1997.

[13] B. Coecke & D. Pavlovic (2007): *Quantum measurements without sums*. In G. Chen, L. Kauffman & S. Lamonaco, editors: *Mathematics of Quantum Computing and Technology*, Taylor and Francis, pp. 567–604, doi:10.1201/9781584889007. arXiv:quant-ph/0608035.

[14] Bob Coecke & Stefano Gogioso (2023): *Quantum in Pictures: A New Way to Understand the Quantum World*. Cambridge Quantum.

[15] Bob Coecke & Aleks Kissinger (2017): *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge, doi:10.1017/9781316219317. Available at `https://www.cambridge.org/core/books/picturing-quantum-processes/1119568B3101F3A685BE832FEEC53E52`.

[16] Geoffrey SH Cruttwell, Bruno Gavranović, Neil Ghani, Paul Wilson & Fabio Zanasi (2022): *Categorical foundations of gradient-based learning*. In: *Programming Languages and Systems: 31st European Symposium on Programming, ESOP 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2–7, 2022, Proceedings*, Springer International Publishing Cham, pp. 1–28, doi:10.1007/978-3-030-99336-8_1.

[17] David Deutsch (2013): *Constructor theory*. *Synthese* 190(18), pp. 4331–4359, doi:10.1007/s11229-013-0279-z.

[18] David Deutsch & Chiara Marletto (2015): *Constructor theory of information*. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471, p. 20140540, doi:10.1098/rspa.2014.0540.

[19] C.-A. Faure, D. J. Moore & C. Piron (1995): *Deterministic evolutions and Schrödinger flows*. *Helvetica Physica Acta* 68(2), pp. 150–157.

[20] Tobias Fritz, Tomáš Gonda & Paolo Perrone (2021): *De Finetti's Theorem in Categorical Probability*. *Journal of Stochastic Analysis* 2(4), doi:10.31390/josa.2.4.06. Available at `http://arxiv.org/abs/2105.02639`. ArXiv:2105.02639 [cs, math, stat].

[21] Dan R. Ghica & Achim Jung (2016): *Categorical semantics of digital circuits*. In: *Proceedings of the 16th Conference on Formal Methods in Computer-Aided Design*, FMCAD '16, FMCAD Inc, Austin, Texas, pp. 41–48, doi:10.1109/FMCAD.2016.7886659.

[22] Stefano Gogioso (2019): *A process-theoretic church of the larger hilbert space*. *arXiv preprint arXiv:1905.13117*, doi:10.48550/arXiv.1905.13117.

[23] L. Hardy (2011): *Foliable operational structures for general probabilistic theories*. In H. Halvorson, editor: *Deep Beauty: Understanding the Quantum World through Mathematical Innovation*, Cambridge University Press, pp. 409–442, doi:10.1017/CBO9780511976971.013. arXiv:0912.4740.

[24] Nathan Haydon & Paweł Sobociński (2020): *Compositional Diagrammatic First-Order Logic*. In Ahti-Veikko Pietarinen, Peter Chapman, Leonie Bosveld-de Smet, Valeria Giardino, James Corter & Sven Linker, editors: *Diagrammatic Representation and Inference*, Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 402–418, doi:10.1007/978-3-030-54249-8_32.

[25] Jules Hedges (2015): *String diagrams for game theory*, doi:10.48550/arXiv.1503.06072. Available at `http://arxiv.org/abs/1503.06072`. ArXiv:1503.06072 [cs, math].

[26] James Hefford, Vincent Wang & Matthew Wilson (2020): *Categories of Semantic Concepts*, doi:10.48550/arXiv.2004.10741. Available at `http://arxiv.org/abs/2004.10741`. ArXiv:2004.10741 [quant-ph].

[27] C. A. R. Hoare & J. He (1987): *The weakest prespecification*. Information Processing Letters 24, pp. 127–132, doi:10.1016/0020-0190(87)90106-2.

[28] Bart Jacobs, Aleks Kissinger & Fabio Zanasi (2019): *Causal Inference by String Diagram Surgery*, doi:10.48550/arXiv.1811.08338. Available at `http://arxiv.org/abs/1811.08338`. ArXiv:1811.08338 [cs, math].

[29] J. M. Jauch (1968): *Mathematical Foundations of Quantum Mechanics*. Addison-Wesley.

[30] Chiara Marletto (2021): *The Science of Can and Can't: A Physicist's Journey Through the Land of Counterfactuals*. Allen Lane, doi:10.1016/S0262-4079(21)00658-8.

[31] D. J. Moore (1999): *On state spaces and property lattices*. Studies in History and Philosophy of Modern Physics 30(1), pp. 61–83, doi:10.1016/S1355-2198(98)00033-1.

[32] C. Piron (1976): *Foundations of quantum physics*. W. A. Benjamin.

[33] J. H. Selby, C. M. Scandolo & B. Coecke (2018): *Reconstructing quantum theory from diagrammatic postulates*. arXiv preprint arXiv:1802.00367, doi:10.22331/q-2021-04-28-445.

[34] Vincent Wang-Mascianica & Bob Coecke (2021): *Talking Space: inference from spatial linguistic meanings*, doi:10.48550/arXiv.2109.06554. Available at `http://arxiv.org/abs/2109.06554`. ArXiv:2109.06554 [cs].

[35] Matthew Wilson, James Hefford, Guillaume Boisseau & Vincent Wang (2021): *The Safari of Update Structures: Visiting the Lens and Quantum Enclosures*. Electronic Proceedings in Theoretical Computer Science 333, pp. 1–18, doi:10.4204/EPTCS.333.1. Available at `http://arxiv.org/abs/2005.05293`. ArXiv:2005.05293 [quant-ph].

Optics for Premonoidal Categories

James Hefford
University of Oxford, UK
james.hefford@cs.ox.ac.uk

Mario Román
Tallinn University of Technology, Estonia
mroman@ttu.ee

We further the theory of optics or "circuits-with-holes" to encompass premonoidal categories: monoidal categories without the interchange law. Every premonoidal category gives rise to an effectful category (i.e. a generalised Freyd-category) given by the embedding of the monoidal subcategory of central morphisms. We introduce "pro-effectful" categories and show that optics for premonoidal categories exhibit this structure.

Pro-effectful categories are the non-representable versions of effectful categories, akin to the generalisation of monoidal to promonoidal categories. We extend a classical result of Day to this setting, showing an equivalence between pro-effectful structures on a category and effectful structures on its free tight cocompletion. We also demonstrate that pro-effectful categories are equivalent to prostrong promonads.

1 Introduction

Monoidal categories play a central role in many categorical models, from programming semantics [3], to quantum theory [2, 15], to electrical circuits [8], for they provide the necessary mathematical structure to describe the interaction of systems over time (by composition) and space (by tensor product). Monoidal categories have a graphical calculus known as string diagrams [30] which provides a formalisation of circuit diagrams. There has been much interest in studying the categorical structure of *circuits-with-holes* [12, 32, 48] over a given monoidal category $\mathscr{C}$; that is, incomplete diagrams in $\mathscr{C}$.

The categorical methods required to describe these circuits-with-holes have their roots in the study of strong profunctors, or *Tambara modules* [46, 6]. These modules are the algebras for a certain promonad [35] with the resulting category of free algebras known as the category of *optics* by the functional programming community, where it is used to model various bidirectional data accessors [14, 40, 41]. The category $\mathsf{Optic}(\mathscr{C})$ has objects given by pairs (a,a') of objects of $\mathscr{C}$ and homs $\mathsf{Optic}(\mathscr{C})((a,a'),(b,b'))$ given by the quotiented sets of the form in Figure 1 [35, 40, 14]. The idea is to produce a category of holes in circuits from $\mathscr{C}$, where two circuits are equivalent if they can be rewritten into each other by sliding boxes. This equivalence relation is handled by the coend $\int^{xy}\mathscr{C}(a, x\otimes b\otimes y)\times\mathscr{C}(x\otimes b'\otimes y, a')$. Outside of functional programming, this category has been suggested as a way to model holes in general monoidal categories [42]. In particular, incomplete diagrams have applications in quantum theory where they are known as *combs* [12] and capture a certain subset of the more general quantum supermaps [13]: optics have been suggested to formalise these structures [25].

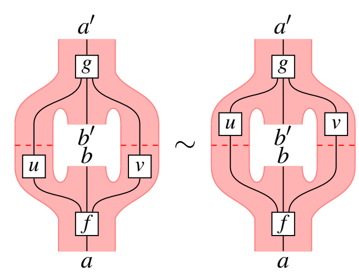


Figure 1: Optics equivalence relation.

What is still missing is a full description of optics for *pre*monoidal categories. Informally, premonoidal categories are like monoidal categories but dropping the interchange law so that in general

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 152–171, doi:10.4204/EPTCS.397.10

$(1 \otimes f)(g \otimes 1) \neq (g \otimes 1)(1 \otimes f)$. Such categories are useful in the modelling of computational side-effects and it was with precisely this motivation that Power and Robinson introduced them [38]. This manuscript is indebted to the research into premonoidal categories [29, 34, 36, 39, 45].

Since not all morphisms in a premonoidal category interchange, there is now an additional subtlety to formalising optics. The coends that are usually used to quotient to allow for the sliding of morphisms can now only be taken over the *centre* $Z\mathscr{C}$ of the premonoidal category $\mathscr{C}$ - the wide monoidal subcategory comprised of the morphisms that interchange with all others.

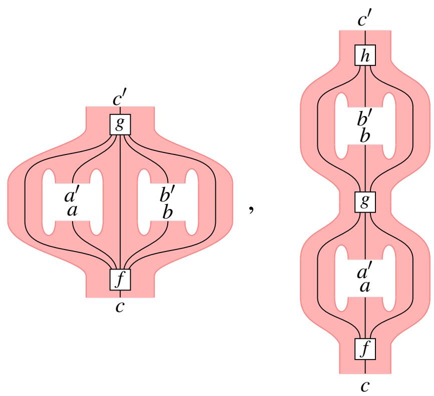


Figure 2: *Left*: $\otimes_H$, *Right*: $\otimes_V$.

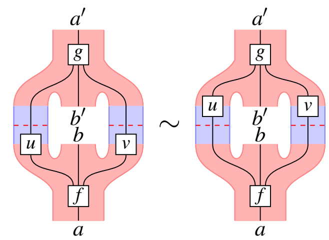


Figure 3: Premonoidal optics equivalence relation.

In this article, we will develop the machinery required to deal with optics for premonoidal categories. There is a category $\mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})$ with objects given by pairs (a,a') of those of $\mathscr{C}$ and homs given by the sets of the form in Figure 3, handled by the coend $\int^{xy \in Z\mathscr{C}} \mathscr{C}(a, x \otimes b \otimes y) \times \mathscr{C}(x \otimes b' \otimes y, a')$. While premonoidal optics can be seen to be a special case of generalised optics for an actegory [14, 40], the full monoidal-like structure of the category has not been discussed before. Optics over a monoidal category $\mathscr{C}$ are equipped with two promonoidal tensor products, $\otimes_H, \otimes_V : \mathsf{Optic}(\mathscr{C}) \times \mathsf{Optic}(\mathscr{C}) \nrightarrow \mathsf{Optic}(\mathscr{C})$, which capture the horizontal and vertical composition of holes (see Figure 2). These two tensors interact to make $\mathsf{Optic}(\mathscr{C})$ into a produoidal category [21].

Over a premonoidal category $\mathscr{C}$ we might hope to equip $\mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})$ with two tensors analogous to those in Figure 2. While the vertical tensor $\otimes_V$ poses no immediate difficulties, the horizontal tensor $\otimes_H$ does: we cannot expect this to be promonoidal because $\mathscr{C}$ does not satisfy interchange. This requires us to introduce the notion of a *pro-effectful* category which combines the structures of premonoidal and promonoidal categories together – this is our main technical contribution.

We prove that pro-effectful categories are equivalently: *(i)* prostrong promonads; *(ii)* biproactegories (two-sided actions in the category of profunctors) which suitably extend a canonical action on the centre of the category; and *(iii)* pseudomonoids in the bicategory of tight $\mathscr{V}^2$-profunctors.

Each of these gives a different perspective on pro-effectful categories, connecting them, respectively, with monads; the action definition of Freyd-categories given by Levy [34]; the pseudomonoid definition of effectful categories given by Román [43] and the work on closed effectful categories due to Power [36, 37]. In particular, this final perspective demonstrates that pro-effectful categories are equivalent to closed effectful categories on the free tight cocompletion, where the effectful structure is given by a version of Day convolution.

Finally, there is an additional challenge with premonoidal optics: there is the category $\mathsf{Optic}(Z\mathscr{C})$ of optics over the monoidal centre and an embedding, $\mathsf{Optic}(Z\mathscr{C}) \to \mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})$, of these central optics into the optics over the entire premonoidal category. $\mathsf{Optic}(Z\mathscr{C})$ is equipped with the two promonoidal structures, $\otimes_H$ and $\otimes_V$, and we would like to understand how these behave in relation to any tensors we can define on $\mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})$. This requires us to keep track of the centre and understand fully how it behaves in relation to the rest of the premonoidal category.

2 Premonoidal and Effectful Categories

Let us start by formally defining premonoidal categories enriched over a fixed cosmos $\mathscr{V}$, taken to be bicomplete and closed symmetric monoidal. We take some space to spell this out as there are some technicalities involved which do not appear to have been explicitly discussed elsewhere. Unless otherwise indicated "category," "functor," "natural transformation" etc. should be taken to mean $\mathscr{V}$-category etc. We write $\otimes$ for the tensor of $\mathscr{V}$ and for the enriched tensor of categories [31].

Definition 1 (Binoidal Category)**.** A category $\mathscr{C}$ is binoidal when, for each object a, it is equipped with a pair of functors $a \ltimes - : \mathscr{C} \to \mathscr{C}$ and $- \rtimes a : \mathscr{C} \to \mathscr{C}$ such that for all a and b, $a \ltimes b = a \rtimes b$.

Remark. In the case of $\mathscr{V} = \mathsf{Set}$, the previous definition is equivalent to the one in terms of the funny tensor product [23, 47], though this must be avoided in the enriched case because it relies on the discrete category $\mathscr{C}_0$ of objects of a category $\mathscr{C}$ which is ill-defined over arbitrary $\mathscr{V}$.

We now generalize the notion of *central* natural transformation to the enriched case.

Definition 2 (Central Natural Transformation)**.** Let $\mathscr{C}$ be a binoidal category and $F, G : \mathscr{D} \to \mathscr{C}$ be two functors. Let $\eta : F \to G$ be a natural transformation, so that we have a family of morphisms $\eta_a : I_{\mathscr{V}} \to \mathscr{C}(Fa, Ga)$ of $\mathscr{V}$ satisfying the usual naturality diagrams. The component η_a is central when the following diagram commutes for all objects c and d, and η is central when all components are central.

$$\begin{tikzcd}
\mathscr{C}(c,d) \otimes I_{\mathscr{V}} \arrow[r, "1 \otimes \eta_a"] & \mathscr{C}(c,d) \otimes \mathscr{C}(Fa,Ga) \arrow[r, "(Ga \ltimes -) \otimes (- \rtimes c)"] & \mathscr{C}(Ga \ltimes c, Ga \ltimes d) \otimes \mathscr{C}(Fa \rtimes c, Ga \rtimes c) \arrow[d, "\circ"] \\
\mathscr{C}(c,d) \arrow[u, "\rho"] \arrow[d, "\lambda"'] & & \mathscr{C}(Fa \rtimes c, Ga \rtimes d) \\
I_{\mathscr{V}} \otimes \mathscr{C}(c,d) \arrow[r, "\eta_a \otimes 1"'] & \mathscr{C}(Fa,Ga) \otimes \mathscr{C}(c,d) \arrow[r, "(- \rtimes d) \otimes (Fa \ltimes -)"'] & \mathscr{C}(Fa \rtimes d, Ga \rtimes d) \otimes \mathscr{C}(Fa \ltimes c, Fa \ltimes d) \arrow[u, "\circ"']
\end{tikzcd}$$

Binoidal categories give us the necessary machinery to define premonoidal categories.

Definition 3 (Premonoidal Category)**.** A premonoidal category $\mathscr{C}$ is a binoidal category endowed with an object i and central natural isomorphisms, $(a \otimes b) \otimes c \cong a \otimes (b \otimes c)$ and $a \otimes i \cong a \cong i \otimes a$, such that the triangle and pentagon equations hold.

As we have just seen, enriched premonoidal categories are require some care to define formally - since the coherence isomorphisms need to be central we are required to define binoidal categories first so that we can make sense of this centrality. Even in the case $\mathscr{V} = \mathsf{Set}$, the 2-category Cat fails to be monoidal under the funny tensor product because the funny tensor of natural transformations is not well-defined unless the components are all central [38]. This prevents the swift and elegant definition "*a premonoidal category is a pseudomonoid in* $\mathsf{Cat}_{\square}$."

Power realised that premonoidal categories are more algebraically well-defined when one shifts to working with premonoidal categories with a specified subcategory of central morphisms [36]. We call these *effectful categories* [43]. In order to define them we first need to give the definition of a functor between premonoidal categories.

Definition 4 (Centre Piece)**.** Let $\mathscr{C}$ be a binoidal category. A centre piece at objects (a,b) in $\mathscr{C}$ is an object $U(a,b)$ in $\mathscr{V}$, endowed with an arrow $\iota : U(a,b) \to \mathscr{C}(a,b)$, such that for any objects (c,d) the following diagrams commute.

$$\begin{tikzcd}
U(a,b) \otimes \mathscr{C}(c,d) \arrow[r, "\iota\otimes 1"] \arrow[d, "\iota\otimes 1"'] & \mathscr{C}(a,b) \otimes \mathscr{C}(c,d) \arrow[r, "(-\rtimes c)\otimes(b\ltimes -)"] & \mathscr{C}(a \rtimes c, b \rtimes c) \otimes \mathscr{C}(b \ltimes c, b \ltimes d) \arrow[d, "\circ\sigma"] \\
\mathscr{C}(a,b) \otimes \mathscr{C}(c,d) \arrow[r, "(-\rtimes d)\otimes(a\ltimes -)"'] & \mathscr{C}(a \rtimes d, b \rtimes d) \otimes \mathscr{C}(a \ltimes c, a \ltimes d) \arrow[r, "\circ"'] & \mathscr{C}(a \rtimes c, b \rtimes d)
\end{tikzcd}$$

$$\begin{tikzcd}
\mathscr{C}(c,d) \otimes U(a,b) \arrow[r, "1\otimes\iota"] \arrow[d, "1\otimes\iota"'] & \mathscr{C}(c,d) \otimes \mathscr{C}(a,b) \arrow[r, "(-\rtimes a)\otimes(d\ltimes -)"] & \mathscr{C}(c \rtimes a, d \rtimes a) \otimes \mathscr{C}(d \ltimes a, d \ltimes b) \arrow[d, "\circ\sigma"] \\
\mathscr{C}(c,d) \otimes \mathscr{C}(a,b) \arrow[r, "(-\rtimes b)\otimes(c\ltimes -)"'] & \mathscr{C}(c \rtimes b, d \rtimes b) \otimes \mathscr{C}(c \ltimes a, c \ltimes b) \arrow[r, "\circ"'] & \mathscr{C}(c \rtimes a, d \rtimes b)
\end{tikzcd}$$

Definition 5 (Premonoidal Functor)**.** A premonoidal functor $F : \mathscr{C} \to \mathscr{D}$ between premonoidal categories is a functor which preserves the centre pieces of $\mathscr{C}$ such that for all centre pieces $\iota : U(a,b) \to \mathscr{C}(a,b)$, composition with F gives a centre piece $U(a,b) \xrightarrow{\iota} \mathscr{C}(a,b) \xrightarrow{F_{a,b}} \mathscr{D}(Fa,Fb)$ at (Fa,Fb) in $\mathscr{D}$. Furthermore F must preserve the premonoidal structure up to natural transformations $Fa \otimes Fb \to F(a \otimes b)$ and $i \to Fi$ subject to coherence conditions like those for a monoidal functor. A premonoidal functor is strict when these transformations are identities.

Definition 6 (Effectful Category)**.** An effectful category consists of a monoidal category $\mathscr{C}_0$, a premonoidal category $\mathscr{C}_1$ with the same objects as $\mathscr{C}_0$ and a strict, identity on objects, premonoidal functor $J : \mathscr{C}_0 \to \mathscr{C}_1$.

Remark. If $\mathscr{C}_0$ is cartesian, then an effectful category is known as a *Freyd category* [45, 34]. At least one good reason to relax from requiring cartesian structure to arbitrary monoidal structure concerns applications of effectful categories outside of computer science. For instance, there has been interest in effectful structure for models of spacetime and quantum theory where products are not the canonical way of taking joint systems [16, 26], and in the study of Petri nets [4].

Effectful categories can be seen as particular instances of actegories [34] - that is, a category with an action by a monoidal category [5, 11]. An effectful category $J : \mathscr{C}_0 \to \mathscr{C}_1$ specifies a left and right $\mathscr{C}_0$-action on $\mathscr{C}_1$, making $\mathscr{C}_1$ into a $\mathscr{C}_0$-$\mathscr{C}_0$-biactegory. These actions $\ltimes : \mathscr{C}_0 \otimes \mathscr{C}_1 \to \mathscr{C}_1$ and $\rtimes : \mathscr{C}_1 \otimes \mathscr{C}_0 \to \mathscr{C}_1$, are special because they preserve the canonical actions of $\mathscr{C}_0$ on itself, i.e. J extends the action $J\boxtimes = \ltimes(1 \otimes J)$ and $J\boxtimes = \rtimes(J \otimes 1)$, and preserves the coherence isomorphisms.

Effectful categories are also precisely the same thing as strong promonads [27, 24, 43]. Given a strong promonad $T : \mathscr{C} \nrightarrow \mathscr{C}$ there is a canonical premonoidal structure on the Kleisli category Kl_T which on objects acts like the tensor of $\mathscr{C}$. The free functor $F : \mathscr{C} \to \mathsf{Kl}_T$ then constitutes an effectful category. Conversely, given an effectful category $J : \mathscr{C}_0 \to \mathscr{C}_1$, there is a promonad $\mathscr{C}_1(J-,J-) : \mathscr{C}_0 \nrightarrow \mathscr{C}_0$ which can be shown to be strong with strengths induced by the action of $\mathscr{C}_0$ on $\mathscr{C}_1$.

Effectful Categories as Pseudomonoids

In this section, we show that effectful categories are pseudomonoids in the category of $\mathscr{V}^2$-enriched categories equipped with a modified version of the funny tensor product, where $\mathscr{V}^2 = [\to, \mathscr{V}]$ is the category of arrows and commutative squares in $\mathscr{V}$. In doing so we place effectful categories on the same footing as monoidal and promonoidal categories, showing that they are representations of the same underlying algebraic data. This builds upon the work of Power who first studied the algebraicity of effectful categories in $\mathscr{V}^2$-Cat [36, 37].

Proposition 1. *Let $\mathscr{V}$ be a complete, cocomplete, closed symmetric monoidal category. Then $\mathscr{V}^2$ is also a complete, cocomplete, closed symmetric monoidal category and therefore constitutes a cosmos.*

Since $\mathscr{V}^2$ is a cosmos, we can consider categories enriched in $\mathscr{V}^2$ [37]. A $\mathscr{V}^2$-category consists of a pair of categories $\mathscr{C}_0$ and $\mathscr{C}_1$ with the same objects, and an identity on objects functor $J : \mathscr{C}_0 \to \mathscr{C}_1$. A $\mathscr{V}^2$-functor $F : J_{\mathscr{C}} \to J_{\mathscr{D}}$ consists of a pair of functors $F_0 : \mathscr{C}_0 \to \mathscr{D}_0$ and $F_1 : \mathscr{C}_1 \to \mathscr{D}_1$ such that $F_1 J_{\mathscr{C}} = J_{\mathscr{D}} F_0$. A $\mathscr{V}^2$-natural transformation $\eta : F \Rightarrow G$ between $\mathscr{V}^2$-functors $F, G : J_{\mathscr{C}} \to J_{\mathscr{D}}$ consists of natural transformations $\eta^0 : F_0 \Rightarrow G_0$ and $\eta^1 : F_1 \Rightarrow G_1$ with components that satisfy $J_{\mathscr{D}}(\eta^0_c) = \eta^1_c$. When $J_{\mathscr{D}}$ is an embedding, we can think of this transformation simply as having central components, in $\mathscr{D}_0$.

There is a bicategory $\mathscr{V}^2$-Cat of $\mathscr{V}^2$-categories, $\mathscr{V}^2$-functors and $\mathscr{V}^2$-natural transformations. This bicategory has an interesting tensor that arises as a slight modification of the funny tensor product.

Definition 7 (Funny Tensor of $\mathscr{V}^2$-Categories)**.** Given two $\mathscr{V}^2$-categories $J_{\mathscr{C}}$ and $J_{\mathscr{D}}$, their funny tensor $J_{\mathscr{C}\square\mathscr{D}} : \mathscr{C}_0 \otimes \mathscr{D}_0 \to \mathscr{C}_1 \square \mathscr{D}_1$ is the identity on objects functor given by the diagonal of the following pushout in $\mathscr{V}$-Cat.

$$\begin{tikzcd}
\mathscr{C}_0 \otimes \mathscr{D}_0 \arrow[r, "J_{\mathscr{C}}\otimes 1"] \arrow[d, "1\otimes J_{\mathscr{D}}"'] & \mathscr{C}_1 \otimes \mathscr{D}_0 \arrow[d, "i_0"] \\
\mathscr{C}_0 \otimes \mathscr{D}_1 \arrow[r, "i_1"'] \arrow[ur, phantom, "\ulcorner" very near end] & \mathscr{C}_1 \square \mathscr{D}_1
\end{tikzcd} \tag{1}$$

The pushout exists because $\mathscr{V}$ is cocomplete and thus $\mathscr{V}$-Cat is also cocomplete [49]. Given $\mathscr{V}^2$-functors $F : J_{\mathscr{A}} \to J_{\mathscr{B}}$ and $G : J_{\mathscr{C}} \to J_{\mathscr{D}}$ their funny tensor $F \square G$ has components $(F \square G)_0 = F_0 \otimes G_0$ and $(F \square G)_1 = F_1 \square G_1$ given by the unique arrow induced by the pushout. The funny tensor is also well-behaved on $\mathscr{V}^2$-natural transformations because their components $J_{\mathscr{D}}(\eta^0_c) = \eta^1_c$ are central and thus interchange with all other morphisms in $\mathscr{C} \square \mathscr{D}$.

Theorem 1. *$\mathscr{V}^2$-Cat is a monoidal 2-category under the funny tensor $\square$.*

Proof sketch. In Appendix A.1. $\square$

This leads to the main theorem of this section. Theorem 2 is equivalent to the result of Román [43]: effectful categories are pseudomonoids in the bicategory of promonads, promonad homomorphisms and promonad modifications.

Theorem 2. *An effectful category is a pseudomonoid in $\mathscr{V}^2$-$\mathsf{Cat}_{\square}$.*

Proof sketch. A pseudomonoid in $\mathscr{V}^2$-$\mathsf{Cat}_{\square}$ consists of a $\mathscr{V}^2$-category $J : \mathscr{C}_0 \to \mathscr{C}_1$ equipped with $\mathscr{V}^2$-functors $\boxtimes : J \square J \to J$ and $I : 1 \to J$, such that there are $\mathscr{V}^2$-natural isomorphisms $\boxtimes(\boxtimes \otimes 1) \overset{\alpha}{\cong} \boxtimes(1 \otimes \boxtimes)$ and $\boxtimes(I \otimes 1) \overset{\lambda}{\cong} 1 \overset{\rho}{\cong} \boxtimes(1 \otimes I)$.

Note that $\boxtimes$ consists of two functors $\boxtimes_0 : \mathscr{C}_0 \otimes \mathscr{C}_0 \to \mathscr{C}_0$ and $\boxtimes_1 : \mathscr{C}_1 \square \mathscr{C}_1 \to \mathscr{C}_1$ such that $J\boxtimes_0 = \otimes_1 J_{\mathscr{C}\square\mathscr{C}}$. $\boxtimes_0$ together with I_0 and the natural isomorphisms α_0, ρ_0 and λ_0, give a monoidal structure on $\mathscr{C}_0$.

The $\mathscr{C}_0$-biaction on $\mathscr{C}_1$ is given by the compositions $\ltimes := \boxtimes i_1$ and $\rtimes := \boxtimes i_0$. That J preserves the canonical actions given by $\otimes_0$ on $\mathscr{C}_0$ follows by the diagram (1) and the equality $J\otimes_0 = \otimes_1 J_{\mathscr{C}\square\mathscr{C}}$, together with the fact that α_1, ρ_1 and λ_1 have components in the image of J. The coherence equations of the biaction are a consequence of those of α_1, ρ_1 and λ_1: for instance α_1 is a natural isomorphism between functors with type $\mathscr{C}_1 \square \mathscr{C}_1 \square \mathscr{C}_1 \to \mathscr{C}_1$. This amounts to "separate" naturality in each $\mathscr{C}_1$ of the domain which in turns induces the left, bimodule and right coherences for the biaction. □

Theorem 3. *There is an equivalence of bicategories $\mathscr{V}^2\text{-}\mathsf{Cat}_{\square} \cong \mathscr{V}\text{-}\mathsf{Promonad}$ between the bicategories of $\mathscr{V}^2$-categories under the funny tensor product and the bicategory of promonads.*

Proof sketch. The result follows upon unwinding the definitions [43] and comparing the rest of the definitions there with those of the present section. Promonads use the "pure tensor" in [43]. □

3 Closed Effectful Categories

Now that we have a thorough understanding of effectful categories, we can start to work towards their "pro-" analogue. To start, recall that a promonoidal category is equivalently a *closed* monoidal presheaf category. This suggests we should turn our attention to the closure of effectful categories, which will be the focus of this section.

Power gave the following definition of closure for effectful categories, where there is still an adjunction between tensoring and the internal-hom, but only for the centre [36].

Definition 8 (Closed Effectful Category [36])**.** An effectful category $J : \mathscr{C}_0 \to \mathscr{C}_1$ is right-closed when for each object X, $J(-) \otimes X : \mathscr{C}_0 \to \mathscr{C}_1$ has a right adjoint $[X, -] : \mathscr{C}_1 \to \mathscr{C}_0$. An effectful category is left-closed when for each X, $X \otimes J(-) : \mathscr{C}_0 \to \mathscr{C}_1$ has a right adjoint. We say an effectful category is closed if it is both left and right-closed.

Power proved the following result which generalises Day's result that every monoidal category embeds into a closed monoidal category [18].

Theorem 4 ([36])**.** *Every (small) effectful category embeds into a closed effectful category.*

We say that an effectful category $J : \mathscr{C}_0 \to \mathscr{C}_1$ is *small* when both $\mathscr{C}_0$ and $\mathscr{C}_1$ are small. Given small J, we can take the strong promonad $T(-,-) := \mathscr{C}_1(J-, J-) : \mathscr{C}_0 \twoheadrightarrow \mathscr{C}_0$ and lift it to a strong cocontinuous monad on the presheaf category $\widehat{T} : \widehat{\mathscr{C}_0} \to \widehat{\mathscr{C}_0}$. The Kleisli category $\mathsf{Kl}_{\widehat{T}}$ has as objects presheaves $F : \mathscr{C}_0^{\mathrm{op}} \to \mathscr{V}$ and homs $\mathsf{Kl}_{\widehat{T}}(F,G) = \widehat{\mathscr{C}_0}(F, \widehat{T}G)$. Moreover, $\widehat{\mathscr{C}_0}$ is monoidal under Day convolution while $\mathsf{Kl}_{\widehat{T}}$ is premonoidal. As a result there is an effectful category given by the identity on objects functor $\widehat{\mathscr{C}_0} \to \mathsf{Kl}_{\widehat{T}}$.

Power gave another characterisation of the effectful category $\widehat{\mathscr{C}_0} \to \mathsf{Kl}_{\widehat{T}}$ as the free *tight* cocompletion of the $\mathscr{V}^2$-category $J : \mathscr{C}_0 \to \mathscr{C}_1$ - that is, the cocompletion in only $\mathscr{V}$-colimits, not all $\mathscr{V}^2$-colimits. In the case of $\mathscr{V} = \mathsf{Set}$ these are precisely the "conical" colimits. The name "tight" was first suggested in [33] where the theory of categories enriched in Cat^2 is studied in some detail.

Theorem 5 ([37])**.** *The free tight cocompletion of a small $\mathscr{V}^2$-category $J : \mathscr{C}_0 \to \mathscr{C}_1$ is the bijective on objects functor $Lan^L_{J^{op}} : \widehat{\mathscr{C}_0} \to \overline{\mathscr{C}_1}$ induced by the functor $Lan_{J^{op}} : \widehat{\mathscr{C}_0} \to \widehat{\mathscr{C}_1}$, via its canonical factorisation into a bijective on objects functor followed by a fully faithful functor (its bo-ff factorisation).*

The category $\overline{\mathscr{C}_1} := \mathrm{Im}(\mathrm{Lan}_{J^{\mathrm{op}}})$ given by the bo-ff factorisation of $\mathrm{Lan}_{J^{\mathrm{op}}}$ has as objects presheaves $F : \mathscr{C}_0^{\mathrm{op}} \to \mathscr{V}$ and as homs $\overline{\mathscr{C}_1}(F,G) = \widehat{\mathscr{C}_1}(\mathrm{Lan}_{J^{\mathrm{op}}}F, \mathrm{Lan}_{J^{\mathrm{op}}}G)$. By the adjunction between extension and restriction of presheaves along J there is a natural isomorphism

$$\mathsf{Kl}_{\widehat{T}}(F,G) = \widehat{\mathscr{C}_0}(F, \widehat{T}G) \cong \widehat{\mathscr{C}_1}(\mathrm{Lan}_{J^{\mathrm{op}}}F, \mathrm{Lan}_{J^{\mathrm{op}}}G) = \overline{\mathscr{C}_1}(F,G)$$

Thus to give a natural transformation $F \Rightarrow \widehat{T}G$ is equivalent to giving one $\mathrm{Lan}_{J^{\mathrm{op}}}F \Rightarrow \mathrm{Lan}_{J^{\mathrm{op}}}G$. This demonstrates an isomorphism $\overline{\mathscr{C}_1} \cong \mathsf{Kl}_{\widehat{T}}$

As a consequence, the following diagram commutes, giving a factorisation (y^L, y^R) of the Yoneda embedding $y : J \to [J^{\mathrm{op}}, \mathscr{V}^2]$, via the free $\mathscr{V}$-cocompletion.

$$\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "J"] \arrow[d, "y_0^L"] \arrow[dd, bend right=60, "y_0"'] & \mathscr{C}_1 \arrow[d, "y_1^L"] \arrow[dd, bend left=60, "y_1"] \\
\widehat{\mathscr{C}_0} \arrow[r, "\mathrm{Lan}^L_{J^{\mathrm{op}}}"] \arrow[d, "y_0^R"] & \overline{\mathscr{C}_1} \arrow[d, "y_1^R"] \\
{[J^{\mathrm{op}}, \mathscr{V}^2]_0} \arrow[r] & {[J^{\mathrm{op}}, \mathscr{V}^2]_1}
\end{tikzcd}$$

So we now have an effectful category $\mathrm{Lan}^L_{J^{\mathrm{op}}}$ into which J embeds. The last thing to do is to check that it is closed, which follows by noting that $\mathrm{Lan}_{J^{\mathrm{op}}}$ is left adjoint to the functor which restricts presheaves along J, and taking bo-ff factorisations ensures that $\mathrm{Lan}^L_{J^{\mathrm{op}}}$ is also a left adjoint [38].

4 $\mathscr{V}^2$-Profunctors

In the previous Section we studied the notion of closure for effectful categories. At this point we could stop and define *pro*-effectful categories as "closed effectful presheaf categories" in analogy to promonoidal categories. In fact, this definition is more subtle than it might first appear and it is certainly worth taking a little care. In particular, given that the closed effectful embedding of any effectful category J is given by the free *tight* cocompletion $\mathrm{Lan}^L_{J^{\mathrm{op}}}$ and not the free cocompletion $[J^{\mathrm{op}}, \mathscr{V}^2]$, we must take care of what we mean by "presheaf" category here. Furthermore, we would like to place pro-effectful categories on the same footing as promonoidal categories - as pseudomonoids in some form of bicategory of profunctors.

This will be the aim of this Section; to study the structure of $\mathscr{V}^2$-profunctors $P : J_{\mathscr{D}}^{\mathrm{op}} \boxtimes J_{\mathscr{C}} \to \mathscr{V}^2$. By the following result we are able to unpack P into a pair of $\mathscr{V}$-profunctors together with a natural transformation between them. The $\mathscr{V}^2$-natural transformations $\phi : P \Rightarrow Q$ can also be similarly unpacked.

Proposition 2. *Let $P : J_{\mathscr{D}}^{op} \otimes J_{\mathscr{C}} \to \mathscr{V}^2$ be a $\mathscr{V}^2$-profunctor. Then P is a triple of:*

1. *a $\mathscr{V}$-profunctor $P_0 : \mathscr{D}_0^{op} \otimes \mathscr{C}_0 \to \mathscr{V}$,*
2. *a $\mathscr{V}$-profunctor $P_1 : \mathscr{D}_1^{op} \otimes \mathscr{C}_1 \to \mathscr{V}$,*
3. *a $\mathscr{V}$-natural transformation $\eta : P_0 \Rightarrow P_1(J^{op} \otimes J)$.*

A $\mathscr{V}^2$-natural transformation $\phi : P \Rightarrow Q$ consists of $\mathscr{V}$-natural transformations $\phi_0 : P_0 \Rightarrow Q_0$ and $\phi_1 : P_1 \Rightarrow Q_1$ such that $(\phi_1(J^{op} \otimes J))\eta^P = \eta^Q \phi_0$.

Proof. This follows by applying a $\mathscr{V}$-enriched version of a result by Power [37, Prop. 24] to the functor category $[J_{\mathscr{D}}^{\mathrm{op}} \otimes J_{\mathscr{C}}, \mathscr{V}^2] \cong \mathsf{Prof}(J_{\mathscr{C}}, J_{\mathscr{D}})$. □

The next proposition demonstrates that the coend of a $\mathscr{V}^2$-profunctor P is given by the coends of P_0 and P_1 together with a canonical arrow between them.

Proposition 3. *Let $P : J^{op} \otimes J \to \mathscr{V}^2$ be a $\mathscr{V}^2$-endoprofunctor. Then the coend $\int^c P(c,c)$ is given by the arrow $\int^c P_0(c,c) \to \int^c P_1(c,c)$ induced by η and the adjunction $y_J \dashv y^J$ in $\mathscr{V}$-Prof.*

Proof. In Appendix A.2. □

$\mathscr{V}^2$-endoprofunctors and the $\mathscr{V}^2$-natural transformations assemble into a $\mathscr{V}^2$-category $[J^{\mathrm{op}} \otimes J, \mathscr{V}^2] \cong \mathsf{Prof}(J,J)$. The category $\mathsf{Prof}(J,J)_0$ consists of the $\mathscr{V}^2$-profunctors and $\mathscr{V}^2$-natural transformations as outlined in Proposition 2, while $\mathsf{Prof}(J,J)_1$ has homs consisting of only the components ϕ_1 of the natural transformations. The identity on objects functor $\mathsf{Prof}(J,J)_0 \to \mathsf{Prof}(J,J)_1$ forgets the ϕ_0 components.

As with any other category of endoprofunctors $\mathsf{Prof}(J,J)$ has a closed monoidal structure given by composition of the profunctors. Given $P = (P_0, P_1, \eta_P)$ and $Q = (Q_0, Q_1, \eta_Q)$, their composition is given by $QP = (Q_0P_0, Q_1P_1, \eta_{QP})$ - we compose the underlying profunctors and take η_{QP} to be given by

$$\int^c Q(-,c) \otimes P(c,-) \xRightarrow{\int \eta_Q \otimes \eta_P} \int^{c \in \mathscr{C}_0} Q(J-,Jc) \otimes P(Jc,J-) \xRightarrow{y_J \dashv y^J} \int^{c \in \mathscr{C}_1} Q(J-,c) \otimes P(c,J-).$$

4.1 Tight Profunctors

In Section 3 we saw that effectful structure on $J : \mathscr{C}_0 \to \mathscr{C}_1$ induced a closed effectful structure on the free tight cocompletion of J. It turns out that this effectful structure on J is only a sufficient and not necessary condition for closed effectful structure on the free tight cocompletion of J. Analogously to the case of monoidal categories where, in order for the presheaf category $\widehat{\mathscr{C}}$ to be closed monoidal it is only necessary that the category $\mathscr{C}$ is promonoidal [18, 17], we only require J to be a "pro-effectful" category. To define these categories we need firstly to study the class of profunctors which factor through the tight cocompletion. This will be the aim of this section.

To define pro-effectful categories we would like to replace the functors of a effectful category with profunctors, but we have a problem: we cannot consider arbitrary $\mathscr{V}^2$-profunctors $P : J_{\mathscr{D}}^{\mathrm{op}} \otimes J_{\mathscr{C}} \to \mathscr{V}^2$ because these assign arbitrary presheaves $J_{\mathscr{D}}^{\mathrm{op}} \to \mathscr{V}^2$ to objects of $J_{\mathscr{C}}$. These presheaves will not in general be contained in the free tight cocompletion. Thus, we need a restricted class of profunctors, those that we call the *tight* profunctors.

Definition 9 (Tight $\mathscr{V}^2$-Profunctor)**.** A tight $\mathscr{V}^2$-profunctor $P : J_{\mathscr{C}} \nrightarrow J_{\mathscr{D}}$ is a $\mathscr{V}^2$-functor $P : J_{\mathscr{C}} \to \overline{J_{\mathscr{D}}}$, where $\overline{J_{\mathscr{D}}} \cong \mathrm{Lan}^L_{J_{\mathscr{D}}^{\mathrm{op}}}$ is the free tight cocompletion of $J_{\mathscr{D}}$.

Remark. Tight $\mathscr{V}^2$-profunctors can be unpacked component-wise analogously to Proposition 2, to see that they are precisely the $\mathscr{V}^2$-profunctors where η is a natural *isomorphism*.

Similarly to how a profunctor $P : \mathscr{C} \nrightarrow \mathscr{D}$ is equivalently a cocontinuous functor between free cocompletions $\widehat{P} : \widehat{\mathscr{C}} \to \widehat{\mathscr{D}}$, tight $\mathscr{V}^2$-profunctors are *tightly* cocontinuous functors between free tight cocompletions.

Definition 10 (Tightly Cocontinuous Functor)**.** A $\mathscr{V}^2$-functor $F : J_{\mathscr{C}} \to J_{\mathscr{D}}$ between tightly cocomplete categories is tightly cocontinuous if it preserves all tight colimits.

Theorem 6 ([31])**.** *Let $\overline{J_{\mathscr{C}}}$ be the closure of $J_{\mathscr{C}}$ in $[J_{\mathscr{C}}^{op}, \mathscr{V}^2]$ under tight colimits and write $y^L : J_{\mathscr{C}} \to \overline{J_{\mathscr{C}}}$ for the inclusion. Then for tightly cocomplete $J_{\mathscr{D}}$, there is an equivalence*

$$Lan_{y^L} : [J_{\mathscr{C}}, J_{\mathscr{D}}] \cong \mathsf{Cocont}_{tight}(\overline{J_{\mathscr{C}}}, J_{\mathscr{D}})$$

where the right-hand is the category of tightly cocontinuous functors. This exhibits $\overline{J_{\mathscr{C}}}$ as the free tight cocompletion of $J_{\mathscr{C}}$.

Indeed, y^L is fully faithful so that there is a natural isomorphism $F \cong (\mathrm{Lan}_{y^L} F) y^L$. Consequently, we can think of a tight $\mathscr{V}^2$-profunctor $P : J_{\mathscr{C}} \to \overline{J_{\mathscr{D}}}$ as a tightly cocontinuous functor $\tilde{P} : \overline{J_{\mathscr{C}}} \to \overline{J_{\mathscr{D}}}$. We can now define the following bicategory of tight $\mathscr{V}^2$-profunctors.

Definition 11. Denote by $\mathscr{V}^2$-$\mathsf{Prof}^{\mathsf{Tight}}$ the bicategory that has

- 0-cells the $\mathscr{V}^2$-categories $J : \mathscr{C}_0 \to \mathscr{C}_1$,
- 1-cells, $P : J_{\mathscr{C}} \nrightarrow J_{\mathscr{D}}$, the tight $\mathscr{V}^2$-profunctors $P : J_{\mathscr{C}} \to \overline{J_{\mathscr{D}}}$,
- 2-cells the $\mathscr{V}^2$-natural transformations.

Composition of 1-cells is given by taking the left Kan extension along y^L and composing the functors we obtain $Q \circ P = (\mathrm{Lan}_{y^L} Q)P$.

Remark. We could also have defined tight $\mathscr{V}^2$-profunctors $J_{\mathscr{C}} \to \overline{J_{\mathscr{D}}}$ as usual $\mathscr{V}^2$-profunctors $J_{\mathscr{C}} \to [J_{\mathscr{D}}^{\mathrm{op}}, \mathscr{V}^2]$ that factorise via the embedding $y^R : \overline{J_{\mathscr{D}}} \to [J_{\mathscr{D}}^{\mathrm{op}}, \mathscr{V}^2]$. Their usual composition as profunctors coincides (up to natural isomorphism) with the composition defined previously because y^R is fully faithful and thus the unit of the Kan extension along y^R is an isomorphism, $F \cong (\mathrm{Lan}_{y^R} F)y^R$. It follows that

$$Q \circ P = (\mathrm{Lan}_y Q)P = (\mathrm{Lan}_{y^R y^L} Q)P \cong (\mathrm{Lan}_{y^R} \mathrm{Lan}_{y^L} Q)P = (\mathrm{Lan}_{y^R} \mathrm{Lan}_{y^L} Q)y^R P' \cong (\mathrm{Lan}_{y^L} Q)P'.$$

Remark. There is a more abstract but cleaner way to define the bicategory $\mathscr{V}^2$-$\mathsf{Prof}^{\mathsf{Tight}}$, by noting that it is the Kleisli bicategory of a certain relative pseudomonad on $\mathscr{V}^2$-Cat. Relative pseudomonads were introduced in [22] where it was also demonstrated that Prof is the Kleisli bicategory of the relative pseudomonad $\widehat{(\cdot)}$ of presheaves, which freely adds colimits by acting on 0-cells as $\mathscr{C} \mapsto \widehat{\mathscr{C}}$. Due to size issues, $\widehat{(\cdot)}$ is a relative pseudomonad and not just a plain pseudomonad: $\widehat{(\cdot)}$ sends small categories to locally small categories and so it is only a relative pseudomonad over the inclusion $\mathsf{Cat} \to \mathsf{CAT}$ of the 2-category of small categories into the 2-category of locally small categories.

In the same fashion there is a relative pseudomonad $\overline{(\cdot)}$ over the inclusion $\mathscr{V}^2$-$\mathsf{Cat} \to \mathscr{V}^2$-$\mathsf{CAT}$ which sends a small $\mathscr{V}^2$-category to its free tight cocompletion. It is then fairly straightforward to check that $\mathscr{V}^2$-$\mathsf{Prof}^{\mathsf{Tight}}$ is the Kleisli bicategory of this relative pseudomonad and therefore also check that it is indeed a bicategory.

$\mathscr{V}^2$-$\mathsf{Prof}^{\mathsf{Tight}}$ has an interesting tensor product given by generalising the funny tensor product.

Proposition 4 (External Tensor Product)**.** *Let $J_{\mathscr{C}}$ and $J_{\mathscr{D}}$ be $\mathscr{V}^2$-categories and write $\overline{J_{\mathscr{C}}}$ and $\overline{J_{\mathscr{D}}}$ be their free tight cocompletions. Then there is a $\mathscr{V}^2$-functor*

$$\hat{\otimes} : \overline{J_{\mathscr{C}}} \,\square\, \overline{J_{\mathscr{D}}} \to \overline{J_{\mathscr{C} \square \mathscr{D}}} \tag{2}$$

with components that act on objects as $(F \hat{\otimes} G)(c,d) := Fc \otimes Gd$.

Proof. In Appendix A.3. □

Definition 12 (Funny Tensor Product of Tight $\mathscr{V}^2$-Profunctors)**.** On categories the funny tensor acts like in $\mathscr{V}^2$-Cat. On tight $\mathscr{V}^2$-profunctors $P : J_{\mathscr{A}} \to \overline{J_{\mathscr{B}}}$ and $Q : J_{\mathscr{C}} \to \overline{J_{\mathscr{D}}}$ we define their funny tensor to be given by their funny tensor in $\mathscr{V}^2$-Cat composed with the external tensor of free tight cocompletions (2):

$$J_{\mathscr{A}} \,\square\, J_{\mathscr{C}} \xrightarrow{P \square Q} \overline{J_{\mathscr{B}}} \,\square\, \overline{J_{\mathscr{D}}} \xrightarrow{\hat{\otimes}} \overline{J_{\mathscr{B} \square \mathscr{D}}}$$

Theorem 7. $\mathscr{V}^2$-$\mathsf{Prof}^{\mathsf{Tight}}$ *is a monoidal bicategory under the funny tensor product.*

Proof sketch. $\mathscr{V}^2$-$\mathsf{Prof}^{\mathsf{Tight}}$ is the Kleisli bicategory of the relative pseudomonad $\overline{(\cdot)}$ that adds tight colimits. Under the funny tensor product on $\mathscr{V}^2$-Cat, this pseudomonad is monoidal, and therefore its Kleisli bicategory is also monoidal. □

5 Pro-effectful Categories

Finally in this Section we are in a position to define pro-effectful categories: as pseudomonoids in $\mathscr{V}^2\text{-}\mathsf{Prof}_{\square}^{\mathsf{Tight}}$, placing them on equal footing algebraically with monoidal, promonoidal and effectful categories.

Definition 13. A pro-effectful category is a pseudomonoid in $\mathscr{V}^2\text{-}\mathsf{Prof}_{\square}^{\mathsf{Tight}}$. Explicitly, a pro-effectful category $J_{\mathscr{C}}$ is a $\mathscr{V}^2$-category equipped with a tensor product tight $\mathscr{V}^2$-profunctor $P : J_{\mathscr{C}\square\mathscr{C}} \nrightarrow J_{\mathscr{C}}$ and a unit tight $\mathscr{V}^2$-profunctor $I : 1 \nrightarrow J_{\mathscr{C}}$, together with $\mathscr{V}^2$-natural isomorphisms $P(P \square 1) \overset{\alpha}{\cong} P(1 \square P)$ and $P(I \square 1) \overset{\lambda}{\cong} 1 \overset{\rho}{\cong} P(1 \square I)$ such that the triangle and pentagon equations hold.

Like their effectful counterparts, pro-effectful categories also have an "actegorical definition" - they are a particular instance of a category equipped with an action by a promonoidal category. This requires us firstly to weaken actegories to proactegories.

Definition 14 (Proactegory)**.** A left proactegory is a promonoidal category $(\mathscr{C}_0, P, I)$ and a category $\mathscr{C}_1$ equipped with a left proaction by $\mathscr{C}_0$, that is, a profunctor $L : \mathscr{C}_0 \otimes \mathscr{C}_1 \nrightarrow \mathscr{C}_1$ and natural isomorphisms

$$\int^{X\in\mathscr{C}_1} L(A,B,X)\otimes L(X,C,D) \overset{a}{\cong} \int^{X\in\mathscr{C}_0} L(A,X,D)\otimes P(X,B,C), \qquad \int^{X\in\mathscr{C}_0} L(A,X,B)\otimes I(X) \overset{l}{\cong} \mathscr{C}_1(A,B),$$

satisfying similar coherence diagrams as for an actegory. A biproactegory is simultaneously a left and right proactegory with an additional natural isomorphism

$$\int^{X} R(D,X,C)\otimes L(X,A,B) \overset{b}{\cong} \int^{X} L(D,A,X)\otimes R(X,B,C)$$

satisfying similar coherences as for a biactegory.

The following result generalises the equivalence between effectful categories and certain actegories [34] to the pro-effectful case.

Proposition 5. *A pro-effectful category is equivalently the following data:*

- *a promonoidal category $(\mathscr{C}_0, P_0, I_0)$,*
- *a category $\mathscr{C}_1$ with the same objects as $\mathscr{C}_0$ and an identity on objects functor $J : \mathscr{C}_0 \to \mathscr{C}_1$,*
- *left and right $\mathscr{C}_0$-proactions on $\mathscr{C}_1$, $P_1^L : \mathscr{C}_0 \otimes \mathscr{C}_1 \nrightarrow \mathscr{C}_1$ and $P_1^R : \mathscr{C}_1 \otimes \mathscr{C}_0 \nrightarrow \mathscr{C}_1$, which extend the canonical proactions of $\mathscr{C}_0$ on itself:*

$$\begin{tikzcd} \mathscr{C}_0\otimes\mathscr{C}_0 \arrow[r, "P_0", "\shortmid"{marking}] \arrow[d, "1\otimes yJ"', "\shortmid"{marking}] & \mathscr{C}_0 \arrow[d, "yJ", "\shortmid"{marking}] \\ \mathscr{C}_0\otimes\mathscr{C}_1 \arrow[r, "P_1^L"', "\shortmid"{marking}] & \mathscr{C}_1 \end{tikzcd} \qquad \begin{tikzcd} \mathscr{C}_0\otimes\mathscr{C}_0 \arrow[r, "P_0", "\shortmid"{marking}] \arrow[d, "yJ\otimes 1"', "\shortmid"{marking}] & \mathscr{C}_0 \arrow[d, "yJ", "\shortmid"{marking}] \\ \mathscr{C}_1\otimes\mathscr{C}_0 \arrow[r, "P_1^R"', "\shortmid"{marking}] & \mathscr{C}_1 \end{tikzcd} \tag{3}$$

- *a natural isomorphism $P_1^R(P_1^L \otimes 1) \cong P_1^L(1 \otimes P_1^R)$ making $\mathscr{C}_1$ into a $\mathscr{C}_0$-$\mathscr{C}_0$-biproactegory.*

Proof. In Appendix A.4. □

The next proposition generalises the equivalence between effectful categories and strong promonads [27, 24, 43] to the pro-effectful case. The proof methods are related to those for promonoidal monads in [19].

Proposition 6. *A pro-effectful category is equivalently a prostrong promonad.*

Proof. Take a prostrong promonad $T : \mathscr{C} \nrightarrow \mathscr{C}$. We will show we have the data of Proposition 5.

T has a Kleisli category in $\mathscr{V}$-Prof and there is an identity on objects free functor $F : \mathscr{C} \to \mathsf{Kl}_T$. By assumption $\mathscr{C}$ has a promonoidal structure (P_0, I_0) and we can use the left and right prostrengths to define left and right proactions of $\mathscr{C}$ on Kl_T. On objects the left proaction acts as $P_1^L(-,c,Fc') := \int^x \mathsf{Kl}_T(-,Fx) \otimes P_0(x,c,c')$ extending the canonical proaction on the centre, so that (3) commutes. Its action on homs is induced by the strength $\int^c P_0(-,-,c) \otimes T(c,-) \Rightarrow \int^c T(-,c) \otimes P_0(c,-,-)$.

Conversely, suppose we are given a pro-effectful category $J : \mathscr{C}_0 \to \mathscr{C}_1$. Then $T(-,=) := \mathscr{C}_1(J-,J=)$ a promonad on $\mathscr{C}_0$ where the promonad multiplication and units are given by composition in $\mathscr{C}_1$. Moreover, $\mathscr{C}_1$ is precisely the Kleisli category of T. Now, since J is pro-effectful, $\mathscr{C}_0$ is promonoidal and we are left to show that T is prostrong over this structure. By Proposition 5, we have left and right proactions of $\mathscr{C}_1$ on $\mathscr{C}_0$ which preserve the canonical proaction on the centre and from these one can construct the prostrength of T. □

Pro-effectful categories are also exactly what is required to place a closed effectful structure on the free tight cocompletion of a $\mathscr{V}^2$-category. This generalises Day's theorem [18, 17] from monoidal to effectful categories, thus also generalising the result of Power on closed effectful embeddings of effectful categories [36, 37]. The result follows by generalising the methods of Day's original proof, and from the folklore results regarding Day convolution for actegories, see [28, 10].

Theorem 8. *There is an equivalence between pro-effectful structures on J and closed effectful structures on the free tight cocompletion* $\overline{J} = Lan^L_{J^{op}}$.

Proof. In Appendix A.5. □

Finally, we note some connections between pro-effectful categories and the premulticategories of Staton and Levy [45], which generalise multicategories by dropping the interchange law. Just as how promonoidal categories are examples of (co)multicategories [20], pro-effectful categories are examples of (co)premulticategories. Given a pro-effectful category $J : \mathscr{C}_0 \to \mathscr{C}_1$, there is a co-premulticategory $\mathfrak{C}$ with objects given by those of $\mathscr{C}_1$. For $a, b \in \mathfrak{C}$ the class of arrows is given by $\mathfrak{C}(a;b) := \mathscr{C}_1(a,b)$ and for $a,b,c \in \mathfrak{C}$ the class of arrows is given by $\mathfrak{C}(a;b,c) := P_1(a,b,c)$. The rest of the classes of arrows are defined inductively.

It is worth noting that there exist examples of pro-effectful categories which provide non-degenerate examples of premulticategories where the interchange law does not hold (in contrast to promonoidal and monoidal categories which are multicategories) and where the "tensor" is not representable (in contrast to monoidal and premonoidal categories). For instance, the premonoidal optics introduced in the next section are an example of such a category.

6 Premonoidal Optics

In a seminal work on optics, Riley [40] introduced the notion of "effectful optics": optics over the Kleisli category of a strong monad. These optics allow the emergence of side-effects, and extend the optics of pure functional programming to other programming languages with effects; with a similar purpose, Abou-Saleh et al [1] have introduced "monadic lenses". More recently, much applied category theory has been written about optics that create effects in different categories [7, 9, 14, 44].

We introduce a novel definition of optic over an effectful category that justifies this previous terminology: optics over the Kleisli category of a strong monad are particular cases of our effectful optics. We also introduce a proeffectful algebra over them that had been previously neglected. In this section we will present the category of optics over a premonoidal category and outline its two tensor-like structures, analogous to those in Figure 2.

Suppose we fix a premonoidal category $\mathscr{C}$ and write $J : Z\mathscr{C} \to \mathscr{C}$ for the inclusion of the centre. There is a $\mathscr{V}^2$-category $\mathsf{Optic}(J)$ with objects given by pairs $\mathbf{a} := (a, a')$ of those of J, i.e. pairs of those of the underlying premonoidal category $\mathscr{C}$. The homs are given by $\int^{xy} Z\mathscr{C}(a, x \otimes b \otimes y) \otimes Z\mathscr{C}(x \otimes b' \otimes y, a') \to \int^{xy \in Z\mathscr{C}} \mathscr{C}(a, x \otimes b \otimes y) \otimes \mathscr{C}(x \otimes b' \otimes y, a')$, as in Figure 4. Thus $\mathsf{Optic}(J)_0 = \mathsf{Optic}(Z\mathscr{C})$ is the usual category of optics over the centre and $\mathsf{Optic}(J)_1 = \mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})$ is the category of optics given by the action of the centre $Z\mathscr{C}$ on the whole premonoidal category $\mathscr{C}$. The identity on objects functor $\mathsf{Optic}(Z\mathscr{C}) \to \mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})$ is the one induced by J.

Theorem 9. $\mathsf{Optic}(J)$ *is a promonoidal* $\mathscr{V}^2$*-category. The* $\mathscr{V}^2$*-profunctors forming the tensor product* $P : \mathsf{Optic}(J) \otimes \mathsf{Optic}(J) \nrightarrow \mathsf{Optic}(J)$ *and unit* $I : 1 \nrightarrow \mathsf{Optic}(J)$ *have components given in Figures 5 and 6. These are explicitly,*

$$P_0(\mathbf{c}, \mathbf{a}, \mathbf{b}) = \int^{xx'yy'} Z\mathscr{C}(c, x \otimes a \otimes x') \otimes Z\mathscr{C}(x \otimes a' \otimes x', y \otimes b \otimes y') \otimes Z\mathscr{C}(y \otimes b' \otimes y', c'),$$

$$P_1(\mathbf{c}, \mathbf{a}, \mathbf{b}) = \int^{xx'yy' \in Z\mathscr{C}} \mathscr{C}(c, x \otimes a \otimes x') \otimes \mathscr{C}(x \otimes a' \otimes x', y \otimes b \otimes y') \otimes \mathscr{C}(y \otimes b' \otimes y', c'),$$

$$I_0(\mathbf{a}) = Z\mathscr{C}(a, a'), \qquad I_1(\mathbf{a}) = \mathscr{C}(a, a').$$

Proof. In Appendix A.6. □

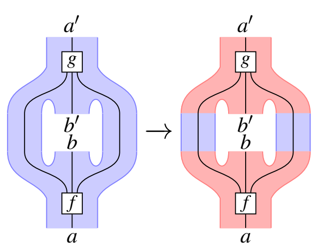


Figure 4: Homs.

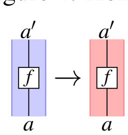


Figure 5: Promonoidal unit I.

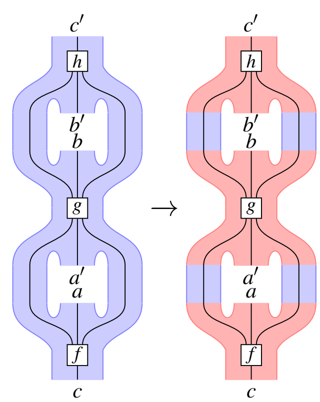


Figure 6: Promonoidal tensor P.

Now let us turn our attention to another tensor-like structure on $\mathsf{Optic}_{1_{Z\mathscr{C}}}(J)$, this one induced by the premonoidal structure on $\mathscr{C}$.

Theorem 10. $\mathsf{Optic}(J)$ *is a pro-effectful category. The tight* $\mathscr{V}^2$*-profunctors forming the tensor product* $P : \mathsf{Optic}(J) \otimes \mathsf{Optic}(J) \to \overline{\mathsf{Optic}(J)}$ *and unit* $I : 1 \to \overline{\mathsf{Optic}(J)}$ *have components which act on objects as,*

$$\begin{aligned} P_0(\mathbf{c}, \mathbf{a}, \mathbf{b}) = P_1(\mathbf{c}, \mathbf{a}, \mathbf{b}) &= \int^{xyz} Z\mathscr{C}(c, x \otimes a \otimes y \otimes b \otimes z) \otimes Z\mathscr{C}(x \otimes a' \otimes y \otimes b' \otimes z, c'), \\ I_0(\mathbf{a}) = I_1(\mathbf{a}) &= Z\mathscr{C}(a, a'). \end{aligned} \tag{4}$$

Proof. In Appendix A.7. □

On the homs of $Z\mathscr{C}$, P_0 and I_0 act in the expected way, essentially by nesting of optics. On the homs of $\mathscr{C}$, P_1 and I_1 act somewhat unusually. Formally non-central optics are sent to natural transformations between left Kan extensions of the expressions in (4), that is between presheaves of the form:

$$(\mathrm{Lan}_{J^{\mathrm{op}}\otimes J} P_0)(\mathbf{c},\mathbf{a},\mathbf{b}) \cong \int^{wvxyz} \mathscr{C}(c,Jw)\otimes Z\mathscr{C}(w,x\otimes a\otimes y\otimes b\otimes z)\otimes Z\mathscr{C}(x\otimes a'\otimes y\otimes b'\otimes z,v)\otimes \mathscr{C}(Jv,c')$$
$$\cong \int^{xyz\in Z\mathscr{C}} \mathscr{C}(c,x\otimes a\otimes y\otimes b\otimes z)\otimes \mathscr{C}(x\otimes a'\otimes y\otimes b'\otimes z,c')$$
$$(\mathrm{Lan}_{J^{\mathrm{op}}\otimes J} I_0)(\mathbf{a}) \cong \int^{xy} \mathscr{C}(a,Jx)\otimes Z\mathscr{C}(x,y)\otimes \mathscr{C}(Jy,a') \cong \int^{x\in Z\mathscr{C}} \mathscr{C}(a,x)\otimes \mathscr{C}(x,a')$$

This justifies thinking of the pro-effectful structure as having the components described in Figures 7 and 8.

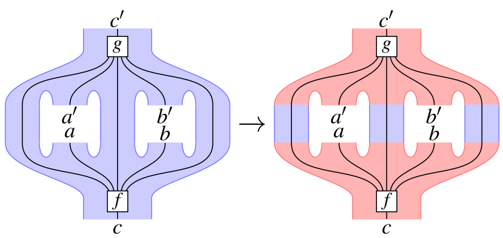


Figure 7: Pro-effectful tensor.

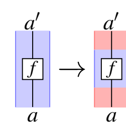


Figure 8: Pro-effectful unit.

Acknowledgements

The authors want to thank Matt Earnshaw and Matt Wilson for discussion; the authors also want to thank the anonymous reviewers at ACT23 for multiple suggestions that improved this article. James Hefford is supported by University College London and the EPSRC [grant number EP/L015242/1]. Mario Román is supported by the European Union through the ESF Estonian IT Academy research measure (2014-2020.4.05.19-0001).

References

[1] Faris Abou-Saleh, James Cheney, Jeremy Gibbons, James McKinna & Perdita Stevens (2016): *Reflections on Monadic Lenses*. In: *A List of Successes That Can Change the World - Essays Dedicated to Philip Wadler on the Occasion of His 60th Birthday*, *Lecture Notes in Computer Science* 9600, Springer, pp. 1–31, doi:10.1007/978-3-319-30936-1_1.

[2] 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.

[3] Andrea Asperti & Guiseppe Longo (1991): *Categories, Types, and Structures*. MIT Press.

[4] John C. Baez, John Foley & Joe Moeller (2019): *Network Models from Petri Nets with Catalysts*. *Compositionality* 1, doi:10.32408/compositionality-1-4.

[5] 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.

[6] Guillaume Boisseau (2020): *String Diagrams for Optics*. In Zena M. Ariola, editor: *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 167, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 17:1–17:18, doi:10.4230/LIPIcs.FSCD.2020.17.

[7] Joe Bolt, Jules Hedges & Philipp Zahn (2019): *Bayesian open games*, doi:10.48550/arXiv.1910.03656. arXiv:1910.03656.

[8] 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.

[9] Dylan Braithwaite & Jules Hedges (2022): *Dependent Bayesian Lenses: Categories of Bidirectional Markov Kernels with Canonical Bayesian Inversion*, doi:10.48550/arXiv.2209.14728. arXiv:2209.14728.

[10] Alexander Campbell (2018): *Skew-Enriched Categories*. *Applied Categorical Structures* 26, p. 597–615, doi:10.1007/s10485-017-9504-0.

[11] Matteo Capucci & Bruno Gavranović (2022): *Actegories for the Working Amthematician*, doi:10.48550/arXiv.2203.16351. arXiv:2203.16351.

[12] Giulio Chiribella, Giacomo Mauro D'Ariano & Paolo Perinotti (2008): *Quantum Circuit Architecture*. *Physical Review Letters* 101(6), doi:10.1103/physrevlett.101.060401.

[13] 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.

[14] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore & Mario Román (2020): *Profunctor optics, a categorical update*, doi:10.48550/arXiv.2001.07488. arXiv:2001.07488.

[15] Bob Coecke & Aleks Kissinger (2017): *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, doi:10.1017/9781316219317.

[16] Marc Comeau (2012): *Premonoidal *-Categories and Algebraic Quantum Field Theory*. Ph.D. thesis, University of Ottawa, doi:10.20381/ruor-5524.

[17] Brian Day (1970): *Construction of Biclosed Categories*. Ph.D. thesis, University of New South Wales, doi:10.26190/unsworks/8048.

[18] 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.

[19] Brian Day (1977): *Note on monoidal monads*. *Journal of the Australian Mathematical Society* 23(3), p. 292–311, doi:10.1017/S1446788700018929.

[20] Brian Day, Elango Panchadcharam & Ross Street (2005): *On centres and lax centres for promonoidal categories*. In: *Colloque International Charles Ehresmann: 100 ans*.

[21] Matt Earnshaw, James Hefford & Mario Román (2023): *The Produoidal Algebra of Process Decomposition*, doi:10.48550/ARXIV.2301.11867. arXiv:2301.11867.

[22] Marcelo Fiore, Nicola Gambino, Martin Hyland & Glynn Winskel (2018): *Relative pseudomonads, Kleisli bicategories, and substitution monoidal structures*. Selecta Mathematica 24, p. 2791–2830, doi:10.1007/s00029-017-0361-3.

[23] 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.

[24] Richard Garner & Ignacio López Franco (2016): *Commutativity*. Journal of Pure and Applied Algebra 220(5), pp. 1707–1751, doi:10.1016/j.jpaa.2015.09.003.

[25] James Hefford & Cole Comfort (2023): *Coend Optics for Quantum Combs*. EPTCS 380, pp. 63–76, doi:10.4204/EPTCS.380.4.

[26] James Hefford & Aleks Kissinger (2023): *On the Pre- and Promonoidal Structure of Spacetime*. EPTCS 380, pp. 284–306, doi:10.4204/EPTCS.380.17.

[27] Bart Jacobs, Chris Heunen & Ichiro Hasuo (2009): *Categorical semantics for arrows*. Journal of Functional Programming 19(3-4), p. 403–438, doi:10.1017/S0956796809007308.

[28] G. Janelidze & G.M. Kelly (2001): *A Note On Actions of a Monoidal Category*. Theory and Applications of Categories 9(4), p. 61–91.

[29] Alan Jeffrey (1997): *Premonoidal categories and flow graphs*. In: Second Workshop on Higher-Order Operational Techniques in Semantics, HOOTS 1997, Stanford, CA, USA, December 8-12, 1997, Electronic Notes in Theoretical Computer Science 10, Elsevier, p. 51, doi:10.1016/S1571-0661(05)80688-7.

[30] 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.

[31] Gregory Kelly (1982): *Basic Concepts of Enriched Category Theory*. Cambridge University Press.

[32] Aleks Kissinger & Sander Uijlen (2019): *A categorical semantics for causal structure*. Logical Methods in Computer Science, doi:10.23638/LMCS-15(3:15)2019.

[33] Stephen Lack & Michael Shulman (2012): *Enhanced 2-categories and limits for lax morphisms*. Advances in Mathematics 229(1), pp. 294–356, doi:10.1016/j.aim.2011.08.014.

[34] Paul Levy (2003): *Call-By-Push-Value*. Springer Dordrecht, doi:10.1007/978-94-007-0954-6.

[35] Craig Pastro & Ross Street (2008): *Doubles for Monoidal Categories*. Theory and Applications of Categories 21(4), pp. 61–75.

[36] John Power (2002): *Premonoidal categories as categories with algebraic structure*. Theoretical Computer Science 278(1), pp. 303–321, doi:10.1016/S0304-3975(00)00340-6. Mathematical Foundations of Programming Semantics 1996.

[37] John Power (2006): *Generic models for computational effects*. Theoretical Computer Science 364(2), pp. 254–269, doi:10.1016/j.tcs.2006.08.006.

[38] 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.

[39] John Power & Hayo Thielecke (2002): *Closed Freyd-and κ-categories*. In: Automata, Languages and Programming: 26th International Colloquium, ICALP'99 Prague, Czech Republic, July 11–15, 1999 Proceedings, Springer, pp. 625–634.

[40] Mitchell Riley (2018): *Categories of Optics*, doi:10.48550/arXiv.1809.00738. arXiv:1809.00738.

[41] Mario Román (2020): *Profunctor optics and traversals*, doi:10.48550/arXiv.2001.08045. arXiv:2001.08045.

[42] Mario Román (2021): *Open Diagrams via Coend Calculus*. *EPTCS* 333, pp. 65–78, doi:10.4204/eptcs.333.5.

[43] Mario Román (2023): *Promonads and String Diagrams for Effectful Categories*. *EPTCS* 380, pp. 344–361, doi:10.4204/EPTCS.380.20.

[44] David I. Spivak (2019): *Generalized Lens Categories via functors* $\mathscr{C}^{\mathrm{op}} \to \mathsf{Cat}$, doi:10.48550/arXiv.1908.02202. arXiv:1908.02202.

[45] Sam Staton & Paul Blain Levy (2013): *Universal Properties of Impure Programming Languages*. *SIGPLAN Not.* 48(1), p. 179–192, doi:10.1145/2480359.2429091.

[46] Daisuke Tambara (2006): *Distributors on a tensor category*. *Hokkaido Mathematical Journal* 35(2), pp. 379 – 425, doi:10.14492/hokmj/1285766362.

[47] Mark Weber (2013): *Free Products of Higher Operad Algebras*. *Theory and Applications of Categories* 28(2).

[48] Matt Wilson, Giulio Chiribella & Aleks Kissinger (2022): *Quantum Supermaps are Characterized by Locality*, doi:10.48550/ARXIV.2205.09844. arXiv:2205.09844.

[49] Harvey Wolff (1974): *V-cat and V-graph*. *Journal of Pure and Applied Algebra* 4(2), pp. 123–135, doi:10.1016/0022-4049(74)90018-8.

A Proofs

A.1 Proof of Theorem 1

Proof sketch. The behaviour of the funny tensor on functors is encapsulated by the following cube.

$$\begin{tikzcd}
& \mathscr{B}_0 \otimes \mathscr{D}_0 \arrow[rr] \arrow[dd] & & \mathscr{B}_1 \otimes \mathscr{D}_0 \arrow[dd] \\
\mathscr{A}_0 \otimes \mathscr{C}_0 \arrow[ur, "F_0 \otimes G_0"] \arrow[rr] \arrow[dd] & & \mathscr{A}_1 \otimes \mathscr{C}_0 \arrow[ur, "F_1 \otimes G_0"'] \arrow[dd] & \\
& \mathscr{B}_0 \otimes \mathscr{D}_1 \arrow[rr] & & \mathscr{B}_1 \,\square\, \mathscr{D}_1 \arrow[uull, phantom, "\ulcorner", very near start] \\
\mathscr{A}_0 \otimes \mathscr{C}_1 \arrow[ur, "F_0 \otimes G_1"] \arrow[rr] & & \mathscr{A}_1 \,\square\, \mathscr{C}_1 \arrow[ur, dashed, "F_1 \square G_1"'] \arrow[uull, phantom, "\ulcorner", very near start] &
\end{tikzcd}$$

Functoriality of $\square$ on 1-cells follows by pasting of cubes and the uniqueness of the arrows induced by the pushout.

Explicitly, we have $(\alpha : F \Rightarrow F') \,\square\, (\beta : G \Rightarrow G')$ has components $(\alpha \,\square\, \beta)^0_{cd} = (\alpha^0_c, \beta^0_d)$ and $(\alpha \,\square\, \beta)^1_{cd} = (\alpha^1_c, \beta^1_d) = (J\alpha^0_c, J\beta^0_d)$. Naturality of this transformation follows from naturality of α and β and from the centrality of the components. $\square$

A.2 Proof of Proposition 3

Proof. Suppose we have a $\mathscr{V}^2$-extranatural family $w_c : P(c,c) \to d$. Then we have the following commutative diagram:

$$\begin{tikzcd}
\mathscr{C}_1(c,c') \otimes P_1(c',c) \arrow[rrr] \arrow[ddd] & & & P_1(c',c') \arrow[ddd, "w^1_{c'}"] \\
& \mathscr{C}_0(c,c') \otimes P_0(c',c) \arrow[r] \arrow[d] \arrow[ul, "J\otimes\eta_{c'c}"'] & P_0(c',c') \arrow[d, "w^0_{c'}"] \arrow[ur, "\eta_{c'c'}"] & \\
& P_0(c,c) \arrow[r, "w^0_c"'] \arrow[dl, "\eta_{cc}"'] & d_0 \arrow[dr, "d"] & \\
P_1(c,c) \arrow[rrr, "w^1_c"'] & & & d_1
\end{tikzcd}$$

In particular, the families $w^0_c : P_0(c,c) \to d_0$ and $w^1_c : P_1(c,c) \to d_1$ are $\mathscr{V}$-extranatural and thus factorise via their respective coends giving arrows $\int^c P_0(c,c) \to d_0$ and $\int^c P_1(c,c) \to d_1$ making the obvious diagrams commute. Now note that the arrows $P_0(c,c) \xrightarrow{\eta_{cc}} P_1(c,c) \xrightarrow{\mathrm{copr}_c} \int^c P_1(c,c)$ are $\mathscr{V}$-extranatural, this induces a arrow $\int^c P_0(c,c) \to \int^c P_1(c,c)$. $\square$

A.3 Proof of Proposition 4

Proof. To give (2) is to give a pair of functors such that the following square commutes:

$$\begin{tikzcd}
\widehat{\mathscr{C}_0} \otimes \widehat{\mathscr{D}_0} \arrow[r] \arrow[d, "\hat{\otimes}_0"'] & \overline{\mathscr{C}_1} \,\square\, \overline{\mathscr{D}_1} \arrow[d, "\hat{\otimes}_1"] \\
\widehat{\mathscr{C}_0 \otimes \mathscr{D}_0} \arrow[r, "\mathrm{Lan}^L_{J^{\mathrm{op}}_{\mathscr{C}\square\mathscr{D}}}"'] & \overline{\mathscr{C}_1 \,\square\, \mathscr{D}_1}
\end{tikzcd}$$

The tensor $\hat{\otimes}_0$ acts on objects following the formula $(F \hat{\otimes} G)(c,d) := Fc \otimes Gd$ and on morphisms in the obvious way. The tensor $\hat{\otimes}_1$ also acts following the same formula, note that a morphism of $\overline{\mathscr{C}_1} \,\square\, \overline{\mathscr{D}_1}$ is a free composition of natural transformations $\alpha : \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}}} F \Rightarrow \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}}} F'$ and $\beta : \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{D}}} G \Rightarrow \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{D}}} G'$ with $(1;\beta)(\alpha;1) \neq (\alpha;1)(1;\beta)$ in general. Each such arrow induces a natural transformation $\mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}\square\mathscr{D}}}(F \otimes G) \Rightarrow \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}\square\mathscr{D}}}(F' \otimes G')$, for instance:

$$\begin{aligned}
\mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}\square\mathscr{D}}}(F \otimes G) &\cong \mathrm{Lan}_{\iota_1^{\mathrm{op}}}(\mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}}} F \otimes G) \xRightarrow{\mathrm{Lan}_{\iota_1^{\mathrm{op}}}(\alpha\otimes 1)} \mathrm{Lan}_{\iota_1^{\mathrm{op}}}(\mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}}} F' \otimes G) \cong \mathrm{Lan}_{\iota_0^{\mathrm{op}}}(F' \otimes \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{D}}} G) \\
&\xRightarrow{\mathrm{Lan}_{\iota_0^{\mathrm{op}}}(1\otimes\beta)} \mathrm{Lan}_{\iota_0^{\mathrm{op}}}(F' \otimes \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{D}}} G') \cong \mathrm{Lan}_{J^{\mathrm{op}}_{\mathscr{C}\square\mathscr{D}}}(F' \otimes G')
\end{aligned}$$

$\square$

A.4 Proof of Proposition 5

Proof. Fix a pro-effectful category (J,P,I). J is a $\mathscr{V}^2$-category so we have two categories $\mathscr{C}_0$ and $\mathscr{C}_1$ with the same objects and an identity on objects functor $J : \mathscr{C}_0 \to \mathscr{C}_1$.

The tight $\mathscr{V}^2$-profunctor $P : J_{\mathscr{C}\square\mathscr{C}} \nrightarrow J_{\mathscr{C}}$ consists of a profunctor $P_0 : \mathscr{C}_0 \otimes \mathscr{C}_0 \nrightarrow \mathscr{C}_0$ and a functor $P_1 : \mathscr{C}_1 \,\square\, \mathscr{C}_1 \to \overline{\mathscr{C}_1}$. Similarly, the tight $\mathscr{V}^2$-profunctor $I : 1 \nrightarrow J_{\mathscr{C}}$ consists of presheaves $I_0 : \mathscr{C}_0^{\mathrm{op}} \to \mathscr{V}$ and $I_1 = \mathrm{Lan}_{J^{\mathrm{op}}} I_0 : \mathscr{C}_1^{\mathrm{op}} \to \mathscr{V}$. (P_0, I_0) induce a promonoidal structure on $\mathscr{C}_0$.

P_1 induces the left and right proactions of $\mathscr{C}_0$ on $\mathscr{C}_1$. Starting with the left proaction, P_1 induces a functor $y_1^R P_1 i_1 =: P_1^L : \mathscr{C}_0 \otimes \mathscr{C}_1 \to \widehat{\mathscr{C}_1}$. It follows that:

$$P_1^L(1 \otimes J) = y_1^R P_1 i_1 (1 \otimes J) = y_1^R P_1 J_{\mathscr{C} \square \mathscr{C}} = \mathrm{Lan}_{J^{\mathrm{op}} \otimes 1 \otimes 1} P_0$$

showing that (3) commutes and that the left proaction extends the canonical one on $\mathscr{C}_0$. A similar argument holds for the right proaction.

Suppose now that we start with the data specified in the proposition. The equalities (3) together with the universal property of the pushout induce a functor $P_1 : \mathscr{C}_1 \square \mathscr{C}_1 \to \overline{\mathscr{C}_1}$

$$\begin{tikzcd}
\mathscr{C}_0 \otimes \mathscr{C}_0 \arrow[r, "J \otimes 1"] \arrow[d, "1 \otimes J"'] & \mathscr{C}_1 \otimes \mathscr{C}_0 \arrow[d, "i_0"] \arrow[ddr, bend left, "P_1^R"] & \\
\mathscr{C}_0 \otimes \mathscr{C}_1 \arrow[r, "i_1"'] \arrow[drr, bend right, "P_1^L"'] & \mathscr{C}_1 \square \mathscr{C}_1 \arrow[dr, dashed, "P_1"] & \\
 & & \overline{\mathscr{C}_1}
\end{tikzcd}$$

and it follows that $P_1 J_{\mathscr{C} \square \mathscr{C}} = \mathrm{Lan}^L_{J^{\mathrm{op}}} P_0$ making (P_0, P_1) the components of a tight $\mathscr{V}^2$-profunctor $P : J_{\mathscr{C} \square \mathscr{C}} \nrightarrow J_{\mathscr{C}}$. The presheaf $I_0 : \mathscr{C}_0^{\mathrm{op}} \to \mathscr{V}$ together with its Kan extension $I_1 := \mathrm{Lan}_{J^{\mathrm{op}}} I_0$ give the components of a $\mathscr{V}^2$-profunctor $I : 1 \nrightarrow J$. Checking all the coherences is a long but ultimately routine calculation. $\square$

A.5 Proof of Theorem 8

Proof. Suppose $J : \mathscr{C}_0 \to \mathscr{C}_1$ is a pro-effectful category. We will show that $\mathrm{Lan}^L_{J^{\mathrm{op}}} : \widehat{\mathscr{C}_0} \to \overline{\mathscr{C}_1}$ is a closed premonoidal category. Since $\mathscr{C}_0$ is promonoidal, $\widehat{\mathscr{C}_0}$ is closed monoidal under Day convolution.

As for the premonoidal structure on $\overline{\mathscr{C}_1}$: on objects it is the same as on $\widehat{\mathscr{C}_0}$. On morphisms, suppose we are given a $\eta : F \Rightarrow G$ in $\overline{\mathscr{C}_1}$. Then we have a $\eta : \mathrm{Lan}_{J^{\mathrm{op}}} F \Rightarrow \mathrm{Lan}_{J^{\mathrm{op}}} G$ and we can describe the left hand part of the premonoidal structure by

$$\begin{aligned}
\mathrm{Lan}_{J^{\mathrm{op}}}(F \star F')(-) &\cong \int^{abc} \mathscr{C}_1(-, Jc) \otimes P_0(c,a,b) \otimes Fa \otimes F'b \cong \int^{ab} P_1^R(-, Ja, b) \otimes Fa \otimes F'b \\
&\cong \int^{bc} P_1^R(-,c,b) \otimes (\mathrm{Lan}_{J^{\mathrm{op}}} F)(c) \otimes F'b \\
&\xRightarrow{\int \eta} \int^{bc} P_1^R(-,c,b) \otimes (\mathrm{Lan}_{J^{\mathrm{op}}} G)(c) \otimes F'b \cong \mathrm{Lan}_{J^{\mathrm{op}}}(G \star F')(-)
\end{aligned}$$

and similarly for the right hand part. It is easily seen that $\mathrm{Lan}^L_{J^{\mathrm{op}}}$ factorises through the centre of this premonoidal structure.

The internal-hom of the left-closed premonoidal structure, $[G,-] : \overline{\mathscr{C}_1} \to \widehat{\mathscr{C}_0}$ is given by

$$[G,H](a) \cong \int_{cd} \mathscr{V}\left(P_1^L(c,a,d), \mathscr{V}\big((\mathrm{Lan}_{J^{\mathrm{op}}} G)(d), (\mathrm{Lan}_{J^{\mathrm{op}}} H)(c) \big) \right)$$

while the right-closed structure is similar, replacing P_1^L with P_1^R. In both cases, checking we have the required adjunction is a matter of standard coend calculus.

Suppose now that $\mathrm{Lan}^L_{J^{\mathrm{op}}}$ is a closed effectful category. Then it follows that $\widehat{\mathscr{C}_0}$ is a closed monoidal category because:

$$\begin{aligned}\widehat{\mathscr{C}_0}\left(-,[G,\mathrm{Lan}^L_{J^{\mathrm{op}}}(=)]\right) &\cong \overline{\mathscr{C}_1}\left(\mathrm{Lan}^L_{J^{\mathrm{op}}}(-)\boxtimes G,\mathrm{Lan}^L_{J^{\mathrm{op}}}(=)\right) = \overline{\mathscr{C}_1}\left(\mathrm{Lan}^L_{J^{\mathrm{op}}}(-\otimes G),\mathrm{Lan}^L_{J^{\mathrm{op}}}(=)\right)\\ &\cong \widehat{\mathscr{C}_0}\left(-\otimes G, J^{\mathrm{op}*}(\mathrm{Lan}^L_{J^{\mathrm{op}}}(=))\right) \cong \widehat{\mathscr{C}_0}\left(-\otimes G,=\right)\end{aligned}$$

where $J^{\mathrm{op}*}$ is the right adjoint to $\mathrm{Lan}^L_{J^{\mathrm{op}}}$, both of which are ioo. Therefore $\mathscr{C}_0$ is a promonoidal category.

The left $\mathscr{C}_0$-proaction on $\mathscr{C}_1$ is given by $P^L_1(-,a,b) := y^L_0(a)\boxtimes y^L_1(b) = \boxtimes i_1(y^L_0(a),y^L_1(b))$ and similarly for the right. These extend the canonical proaction because:

$$\begin{aligned}P^L_1(-,a,Jb) &= \boxtimes i_1(y^L_0(a),y^L_1(Jb)) = \boxtimes i_1(y^L_0(a),\mathrm{Lan}^L_{J^{\mathrm{op}}}y^L_0(b)) = \boxtimes i_1(1\otimes_{\mathscr{V}}\mathrm{Lan}^L_{J^{\mathrm{op}}})(y^L_0(a),y^L_0(b))\\ &= \mathrm{Lan}^L_{J^{\mathrm{op}}}\otimes(y^L_0(a),y^L_0(b)) = \mathrm{Lan}^L_{J^{\mathrm{op}}}P_0(-,a,b)\end{aligned}$$

where we have written the monoidal operation $\otimes$ on $\widehat{\mathscr{C}_0}$ and the premonoidal operation $\boxtimes$ on $\overline{\mathscr{C}_1}$ with prefix notation. $\square$

A.6 Proof of Theorem 9

Proof. J has commutative left and right actions by the monoidal $\mathscr{V}^2$-category $1_{Z\mathscr{C}} : Z\mathscr{C}\to Z\mathscr{C}$. Consider the $\mathscr{V}^2$-category $\mathsf{Tamb}(J)$ of Tambara modules on J [35, 14], whose objects are the $\mathscr{V}^2$-endoprofunctors $P : J \nrightarrow J$ equipped with left and right strengths over the action by $1_{Z\mathscr{C}}$. The morphisms are the bistrong $\mathscr{V}^2$-natural transformations. We can use Proposition 2 to unpack $\mathsf{Tamb}(J)$ into two $\mathscr{V}$-categories and an identity on objects functor, $\mathsf{Tamb}(J)_0 \to \mathsf{Tamb}(J)_1$. The objects of $\mathsf{Tamb}(J)_0$ and $\mathsf{Tamb}(J)_1$ are the bistrong endoprofunctors $P : J \nrightarrow J$ which are equivalently triples $(P_0 : Z\mathscr{C} \nrightarrow Z\mathscr{C}, P_1 : \mathscr{C}\nrightarrow\mathscr{C}, \eta : P_0 \Rightarrow P_1(J^{\mathrm{op}}\otimes J))$. $\mathsf{Tamb}(J)_0$ has arrows $\phi : P\Rightarrow Q$ given by pairs $(\phi_0 : P_0 \Rightarrow Q_0, \phi_1 : P_1 \Rightarrow Q_1)$ while $\mathsf{Tamb}(J)_1$ has only the ϕ_1 as arrows.

It is known that the category of Tambara modules is equivalent to the presheaf category of the category of optics [35, 14], which in this particular case implies $[\mathsf{Optic}(J)^{\mathrm{op}},\mathscr{V}^2]\cong\mathsf{Tamb}(J)$. The $\mathscr{V}^2$-category $\mathsf{Optic}(J)$ has objects given by pairs $\mathbf{a} = (a,a')$ of $\mathsf{Optic}(J)$ and homs given by

$$\mathsf{Optic}(J)(\mathbf{a},\mathbf{b}) = \int^{xy\in 1_{Z\mathscr{C}}} J(a,x\otimes b\otimes y)\otimes J(x\otimes b'\otimes y,a')$$

where $J(-,-) := Z\mathscr{C}(-,-)\to\mathscr{C}(-,-)$ is the hom of J as a $\mathscr{V}^2$-category and the coend is taken in this fully enriched setting. By Proposition 3 this coend is given by the following arrow.

$$\int^{xy} Z\mathscr{C}(a,x\otimes b\otimes y)\otimes Z\mathscr{C}(x\otimes b'\otimes y,a') \to \int^{xy\in Z\mathscr{C}}\mathscr{C}(a,x\otimes b\otimes y)\otimes\mathscr{C}(x\otimes b'\otimes y,a')$$

As a result, the identity on objects functor equivalent to $\mathsf{Optic}(J)$ is given by $\mathsf{Optic}(Z\mathscr{C})\to\mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})$ as expected.

Now, since $\mathsf{Tamb}(J)$ has a closed monoidal structure given by composition of the profunctors, there is an induced promonoidal structure on $\mathsf{Optic}(J)$. To arrive at the explicit expressions claimed in the Theorem, take objects $\mathbf{a}$ and $\mathbf{b}$ of $\mathsf{Optic}(J)$ and consider the tensor (i.e. composition as profunctors) of the associated representable presheaves.

$$(y_{\mathbf{a}}\otimes y_{\mathbf{b}})(-)\cong\int^{wxyz\in 1_{Z\mathscr{C}}} J(-,w\otimes a\otimes x)\otimes J(w\otimes a'\otimes x,y\otimes b\otimes z)\otimes J(y\otimes b'\otimes z,-)$$

This can be unpacked by Proposition 3 to give the result.

Finally note that the unit of the monoidal structure on $\mathsf{Tamb}(J)$ is $1_J : J\nrightarrow J$, which is $(1_{Z\mathscr{C}}, 1_{\mathscr{C}}, \eta : 1_{Z\mathscr{C}}\Rightarrow y^J y_J)$. $\square$

A.7 Proof of Theorem 10

Proof. The free tight cocompletion of $\mathsf{Optic}(J)$ is given by $[\mathsf{Optic}(Z\mathscr{C})^{\mathrm{op}}, \mathscr{V}] \to \overline{\mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})}$. We will show that this is a closed effectful category and then by Theorem 8 we will be done.

Start by considering the effectful category $J^{\mathrm{op}} \otimes J : Z\mathscr{C}^{\mathrm{op}} \otimes Z\mathscr{C} \to \mathscr{C}^{\mathrm{op}} \otimes \mathscr{C}$. The free tight cocompletion of this category is $\mathrm{Lan}^{L}_{J^{\mathrm{op}}\otimes J} : \mathsf{Prof}(Z\mathscr{C}) \to \overline{\mathsf{Prof}(\mathscr{C})}$ which is closed effectful. The domain is the duoidal category $\mathsf{Prof}(Z\mathscr{C})$ of endoprofunctors on $Z\mathscr{C}$ and it has a closed monoidal structure given by Day convolution over the monoidal structure of $Z\mathscr{C}$:

$$P * Q := \int^{aa'bb'} Z\mathscr{C}(-, a \otimes a') \otimes P(a,b) \otimes Q(a',b') \otimes Z\mathscr{C}(b \otimes b', -) \tag{5}$$

The premonoidal structure on $\overline{\mathsf{Prof}(\mathscr{C})}$ is given on objects by (5), and on homs, given a $\eta : P \Rightarrow P'$ in $\overline{\mathsf{Prof}(\mathscr{C})}$ (that is, a $\eta : \mathrm{Lan}_{J^{\mathrm{op}}\otimes J} P \Rightarrow \mathrm{Lan}_{J^{\mathrm{op}}\otimes J} P'$) the left side of the premonoidal structure is given by:

$$\begin{aligned}
\mathrm{Lan}_{J^{\mathrm{op}}\otimes J}(P * Q) &\cong \int^{aa'bb'} \mathscr{C}(-, J(a \otimes a')) \otimes P(a,b) \otimes Q(a',b') \otimes \mathscr{C}(J(b \otimes b'), -) \\
&\cong \int^{a'b' \in Z\mathscr{C}, cd \in \mathscr{C}} \mathscr{C}(-, c \rtimes a')) \otimes (\mathrm{Lan}_{J^{\mathrm{op}}\otimes J} P)(c,d) \otimes Q(a',b') \otimes \mathscr{C}(d \rtimes b', -) \\
&\xRightarrow{\int \eta} \int^{a'b' \in Z\mathscr{C}, cd \in \mathscr{C}} \mathscr{C}(-, c \rtimes a')) \otimes (\mathrm{Lan}_{J^{\mathrm{op}}\otimes J} P')(c,d) \otimes Q(a',b') \otimes \mathscr{C}(d \rtimes b', -) \\
&\cong \mathrm{Lan}_{J^{\mathrm{op}}\otimes J}(P' * Q)
\end{aligned}$$

Since $\mathrm{Lan}^{L}_{J^{\mathrm{op}}\otimes J}$ is a left adjoint, it follows that it is a closed effectful category.

There is a $\mathscr{V}^2$-category $\mathsf{Tamb}(Z\mathscr{C}) \to \overline{\mathsf{Tamb}(\mathscr{C})}$ with objects given by the Tambara modules on $Z\mathscr{C}$. The homs of $\mathsf{Tamb}(Z\mathscr{C})$ are the bistrong natural transformations while the homs of $\overline{\mathsf{Tamb}(\mathscr{C})}$ are the bistrong natural transformations between the left Kan extensions along $J^{\mathrm{op}} \otimes J$ of the Tambara modules. This $\mathscr{V}^2$-category inherits a closed effectful structure from $\mathrm{Lan}^{L}_{J^{\mathrm{op}}\otimes J}$ given by a certain quotient of (5) which acts to normalise the duoidal structure on $\mathsf{Prof}(Z\mathscr{C})$ [24, 21].

Finally note that the presheaf category of optics is equivalent to the category of Tambara modules, $\widehat{\mathsf{Optic}(Z\mathscr{C})^{\mathrm{op}}} \cong \mathsf{Tamb}(Z\mathscr{C})$ [14], and we can finally check that we also have $\overline{\mathsf{Optic}_{Z\mathscr{C}}(\mathscr{C})} \cong \overline{\mathsf{Tamb}(\mathscr{C})}$. □

Overdrawing Urns using Categories of Signed Probabilities

Bart Jacobs and Dario Stein
iHub, Radboud University Nijmegen
December 14, 2023
bart.jacobs@ru.nl dario.stein@ru.nl

A basic experiment in probability theory is drawing without replacement from an urn filled with multiple balls of different colours. Clearly, it is physically impossible to overdraw, that is, to draw more balls from the urn than it contains. This paper demonstrates that overdrawing does make sense mathematically, once we allow signed distributions with negative probabilities. A new (conservative) extension of the familiar hypergeometric ('draw-and-delete') distribution is introduced that allows draws of arbitrary sizes, including overdraws. The underlying theory makes use of the dual basis functions of the Bernstein polynomials, which play a prominent role in computer graphics. Negative probabilities are treated systematically in the framework of categorical probability and the central role of datastructures such as multisets and monads is emphasised.

1 Introduction

For drawing (multiple) coloured balls from a statistical urn, we distinguish three well-known modes:

1. *hypergeometric* or *draw-and-delete*, which is drawing a ball from the urn without replacement, so that the urn shrinks;
2. *multinomial* or *draw-and-replace*: drawing with replacement, so that the urn remains the same;
3. *Pólya* or *draw-and-duplicate*, which is drawing a ball from the urn and replacing it together with an additional ball of the same colour, so that the urn grows.

Multinomial and Pólya draws may be of arbitrary size, but hypergeometric draws are limited in size by the number of balls in the urn. In this paper we lift this limitation and allow hypergeometric draws of arbitrary size, including 'overdraws', containing more balls than in the urn. Physically this is strange, but, as will show, mathematically it makes sense once we allow negative probabilities.

Negative probabilities have emerged in quantum physics (*e.g.* in double slit experiments) and have been discussed in the work of famous physicists like Wigner, Dirac, and Feynman (see *e.g.* [7] and the references mentioned there). There are also 'classical' (non-quantum) examples, such as the one of Piponi (discussed in [1]) or of Székely [22] with two half coins, involving infinitely many both positive and negative probabilities, whose (convolution) sum is an ordinary (fair) coin. Also, negative probabilities have come up in finance, see *e.g.* [20]. Despite the lack of the clear operational meaning that their nonnegative counterparts have, negative probabilities appear as convenient tools in a variety of contexts in mathematics and physics, see [2, 23, 1, 7].

We briefly explain the nature of our extension, already using some notation that will be explained below. The Pólya distribution can be expressed as a mixture of multinomial draws; that is, we can break up such a draw into two stages: first sample a random distribution ω from the Dirichlet distribution, and then make independent (multinomial) draws from ω. The self-reinforcing behaviour of Pólya's urn is entirely captured by the latent Dirichlet distribution. In the Kleisli category of the Giry-monad,

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 172–189, doi:10.4204/EPTCS.397.11

with (Kleisli) morphisms called channels, this corresponds to a factorisation of the Pólya channel $pol[K]$ through the multinomial channel $mn[K]$, with draws of size K.

$$pol[K] \quad = \quad \mathcal{M}[N](X) \xrightarrow{Dir} \mathcal{D}(X) \xrightarrow{mn[K]} \mathcal{M}[K](X) \tag{1}$$

We write $\mathcal{M}[N](X)$ for the space of multisets (urns) on a set X of size N, and $\mathcal{D}(X)$ is the set of finite distributions. Since such factorisations arise from De Finetti's famous theorem, we call (1) a De Finetti factorisation for the Pólya's distribution (*e.g.* [15]).

The hypergeometric distribution does not admit such a De Finetti factorisation since an urn containing N balls is exhausted after N draws. However, there is a way out, if we extend our notion of probability to allow negative (signed) probabilities. Such models satisfy the usual axioms of categorical probability, and we can find a De Finetti factorization of the hypergeometric channel $hg[K]$, for draws of size K, of the form:

$$hg[K] \quad = \quad \mathcal{M}[N](X) \xrightarrow{DDir} \mathcal{D}(X) \xrightarrow{mn[K]} \mathcal{M}[K](X) \tag{2}$$

It uses a signed 'Dual Dirichlet' distribution $DDir$ which we develop in analogy to the Dirichlet distribution occurring in Pólya's urn. Existence of the factorisation can be deduced from earlier work [16, 5, 18] connecting finite versions of De Finetti's theorem to signed probability. In this case, the factorisation (2) is not unique. We claim that a canonical choice is given by the Dual Bernstein polynomials, which have been studied widely in computer graphics [19, 6, 24, 17], but their appearance in a probabilistic context is novel. Evaluating (2) for overdraws $K \geq N$ defines a signed extension of the hypergeometric distributions which includes overdraws, while agreeing with the usual distribution for ordinary draws $K \leq N$.

Our contributions are:

1. a principled approach to signed probability (discrete and continuous) using multisets and monads;
2. conceptualizing dual Bernstein polynomials as signed probability densities;
3. defining signed hypergeometric distributions that conservatively extend hypergeometric draws while preserving good properties;
4. explicating the dual Dirichlet distribution and its conjugate prior relationships via string diagrams.

2 Multisets

A multiset, also known as bag, is like a subset except that elements may occur multiple times. We shall use ket notation $n_1|x_1\rangle + \cdots + n_k|x_k\rangle$ to describe a multiset with k elements, where element x_i, say from a set X, occurs $n_i \in \mathbb{N}$ many times. Equivalently, such a multiset may be described as a function $\varphi\colon X \to \mathbb{N}$ with finite support $supp(\varphi) \coloneqq \{x \in X \mid \varphi(x) \neq 0\}$. The number $\varphi(x) \in \mathbb{N}$ is the multiplicity of $x \in X$; it says how many times x occurs in the multiset φ.

We shall write $\mathcal{M}(X)$ for the set of multisets with elements from a set X, and $\mathcal{M}_{\mathrm{fs}}(X) \subseteq \mathcal{M}(X)$ for the subset of multisets φ with full support, that is, with $supp(\varphi) = X$. The latter only makes sense when X is a finite set. As canonical finite sets we write $\boldsymbol{n} \coloneqq \{0, 1, \ldots, n-1\}$, for $n \in \mathbb{N}$.

The size $\|\varphi\|$ of a multiset φ is the total number of elements, including multiplicities. Thus, $\|\varphi\| \coloneqq \sum_x \varphi(x)$, or, in ket notation, $\|\sum_i n_i|x_i\rangle\| = \sum_i n_i$. We write $\mathcal{M}[K](X) \coloneqq \big\{\varphi \in \mathcal{M}(X) \bigm| \|\varphi\| = K\big\}$ for the set of multisets of size $K \in \mathbb{N}$. When the set X has $n \geq 1$ elements, the number of multisets of size K in $\mathcal{M}[K](X)$ is $\left(\!\binom{n}{K}\!\right) = \binom{n+K-1}{K} = \frac{(n+K-1)!}{K!\cdot(n-1)!}$. For instance, for a set $X = \{a,b,c\}$ with three elements there are $\left(\!\binom{3}{3}\!\right) = \frac{5!}{3!\cdot 2!} = 10$ multisets of size $K = 3$, namely: $3|a\rangle$, $3|b\rangle$, $3|c\rangle$, $2|a\rangle + 1|b\rangle$, $2|a\rangle + 1|c\rangle$,

$1|a\rangle + 2|b\rangle,\ 2|b\rangle + 1|c\rangle,\ 1|a\rangle + 2|c\rangle,\ 1|b\rangle + 2|c\rangle,\ 1|a\rangle + 1|b\rangle + 1|c\rangle$. Only the last one has full support. The factorial $n!$ and binomial coefficients $\binom{n}{m}$ and $\left(\!\binom{n}{m}\!\right)$ are extended from numbers to multisets (as in [13]).

Definition 1 *Let $\varphi, \psi \in \mathcal{M}(X)$ be two multisets. We define*

1. $\varphi! := \prod_x \varphi(x)!$;
2. $(\varphi) := \frac{\|\varphi\|!}{\varphi!}$;
3. $\varphi \leq \psi$ *iff* $\varphi(x) \leq \psi(x)$ *for each* $x \in X$, *and* $\varphi \leq_K \psi$ *iff* $\varphi \leq \psi$ *and* $\|\varphi\| = K$;
4. $(\psi - \varphi)(x) = \psi(x) - \varphi(x)$, *when* $\varphi \leq \psi$;
5. $\binom{\psi}{\varphi} := \frac{\psi!}{\varphi! \cdot (\psi-\varphi)!} = \prod_x \binom{\psi(x)}{\varphi(x)}$, *when* $\varphi \leq \psi$;
6. $\left(\!\binom{\psi}{\varphi}\!\right) := \frac{(\psi+\varphi-\mathbf{1})!}{\varphi! \cdot (\psi-\mathbf{1})!} = \prod_x \left(\!\binom{\psi(x)}{\varphi(x)}\!\right)$ *when* ψ *has full support, where* $\mathbf{1} = \sum_x 1|x\rangle$ *is the multiset of singletons.*

3 Discrete distributions

A discrete probability distribution $\sum_i r_i|x_i\rangle$ looks like a multiset, except that the multiplicities r_i are now in the unit interval $[0,1] \subseteq \mathbb{R}$ and add up to one: $\sum_i r_i = 1$. We write $\mathcal{D}(X)$ for the set of such distributions with $x_i \in X$. Alternatively, like for multisets, elements $\omega \in \mathcal{D}(X)$ may be described as functions $\omega\colon X \to [0,1]$ with finite support and with $\sum_x \omega(x) = 1$. When the set X is finite, we write $\mathcal{D}_{\mathrm{fs}}(X) \subseteq \mathcal{D}(X)$ for the discrete distributions with full support: $\mathit{supp}(\omega) = X$. An example is the uniform distribution $\sum_{x\in X} \frac{1}{n}|x\rangle$, where $n \geq 1$ is the number of elements of a non-empty set X. Concretely, a fair coin is described by the distribution $\frac{1}{2}|H\rangle + \frac{1}{2}|T\rangle$ for $X = \{H, T\}$.

Each non-empty multiset $\varphi \in \mathcal{M}(X)$ can be turned into a distribution via normalisation. We call this frequentist learning, since it involves learning a distribution by counting, and write it as:

$$\mathit{flrn}(\varphi) := \frac{\varphi}{\|\varphi\|} = \sum_{x\in X} \frac{\varphi(x)}{\|\varphi\|}\big|x\big\rangle \in \mathcal{D}(X). \tag{3}$$

This frequentist learning is natural in X, but it is not a map of monads, from (non-empty) multisets to distributions.

The set $\mathcal{D}(\boldsymbol{n})$ of distributions on $\boldsymbol{n} = \{0, \ldots, n-1\}$ can be identified with the simplex $\Delta^n \subseteq \mathbb{R}^n$, where:

$$\Delta^n := \big\{(r_0, \ldots, r_{n-1}) \in \mathbb{R}_{\geq 0} \;\big|\; \textstyle\sum_i r_i = 1\big\}. \tag{4}$$

This is commonly called the $n-1$ simplex.

There are three famous 'draw' distributions, called multinomial, hypergeometric and Pólya. We briefly describe them in the style of [14] and refer there for more information. These distributions are all parameterised by a draw size K and form distributions on the set $\mathcal{M}[K](X)$ of multisets (as draws). One may think of X as a set of colors.

Definition 2 *We fix a set X and a number $K \in \mathbb{N}$.*

1. *For a distribution $\omega \in \mathcal{D}(X)$, used as abstract urn, the multinomial distribution $mn[K](\omega) \in \mathcal{D}\big(\mathcal{M}[K](X)\big)$ is defined as:*

$$mn[K](\omega) := \sum_{\varphi\in\mathcal{M}[K](X)} (\varphi) \cdot \prod_{x\in X} \omega(x)^{\varphi(x)} \big|\varphi\big\rangle.$$

2. *For an 'urn' multiset $\upsilon \in \mathcal{M}(X)$, with size $L := \|\upsilon\| \geq K$ there is a hypergeometric distribution $hg[K](\upsilon) \in \mathcal{D}\big(\mathcal{M}[K](X)\big)$ with:*

$$hg[K](\upsilon) := \sum_{\varphi \leq_K \upsilon} \frac{\binom{\upsilon}{\varphi}}{\binom{L}{K}} \, \big| \, \varphi \, \big\rangle.$$

The size restriction $K \leq L$ excludes overdraws.

3. *Similarly, for a multiset $\upsilon \in \mathcal{M}[L](X)$ with full support, there is the Pólya distribution $pol[K](\upsilon) \in \mathcal{D}\big(\mathcal{M}[K](X)\big)$ with:*

$$pol[K](\upsilon) := \sum_{\varphi \in \mathcal{M}[K](X)} \frac{\left(\!\binom{\upsilon}{\varphi}\!\right)}{\left(\!\binom{L}{K}\!\right)} \, \big| \, \varphi \, \big\rangle.$$

Intuitively, in the multinomial case drawn balls are returned to the urn, so that the urn does not change and can be described abstractly as a discrete distribution. In the hypergeometric case a drawn ball is removed from the urn, and in the Pólya case the drawn ball is returned together with a new ball of the same colour. Thus in the hypergeometric case the urns shrinks, whereas in the Pólya case the urn grows.

Given two distributions $\omega \in \mathcal{D}(X)$ and $\rho \in \mathcal{D}(Y)$ we can form a parallel (tensor) product $\omega \otimes \rho \in \mathcal{D}(X \times Y)$, via $(\omega \otimes \rho)(x,y) = \omega(x) \cdot \rho(y)$.

The mapping $X \mapsto \mathcal{D}(X)$ is a monad on the category of sets. We will not spell out what this means, but we will use the resulting Kleisli category $\mathcal{K}\ell(\mathcal{D})$, whose maps $c \colon X \to \mathcal{D}(Y)$ will be called channels and written as $c \colon X \rightsquigarrow Y$. For instance, the distributions from Definition 2 can be described as channels $mn[K] \colon \mathcal{D}(X) \rightsquigarrow \mathcal{M}[K](X)$, $hg[K] \colon \mathcal{M}[L](X) \rightsquigarrow \mathcal{M}[K](X)$ and $pol[K] \colon \mathcal{M}_{\mathrm{fs}}(X) \rightsquigarrow \mathcal{M}[K](X)$.

For a channel $c \colon X \rightsquigarrow Y$ and a distribution $\omega \in \mathcal{D}(X)$ on the domain X we can form a distribution $c \gg\!= \omega$ on the codomain Y via pushforward (also called state transformation):

$$\big(c \gg\!= \omega\big)(y) := \sum_{x \in X} \omega(x) \cdot c(x)(y).$$

Given another channel $d \colon Y \rightsquigarrow Z$ one can form a composite channel $d \odot c \colon X \rightsquigarrow Z$ via $(d \odot c)(x) := d \gg\!= c(x)$. Notice that we use a special circle $\odot$, with a dot, for composition of channels.

A basic channel is $DD \colon \mathcal{M}[K{+}1](X) \rightsquigarrow \mathcal{M}[K](X)$, where DD stands for draw-delete. It probabilistically draws and removes one ball from an urn $\upsilon \in \mathcal{M}[K{+}1](X)$ with $K{+}1$ balls, via:

$$DD(\upsilon) := \sum_{x \in supp(\upsilon)} \frac{\upsilon(x)}{\|\upsilon\|} \, \big| \, \upsilon - 1|x\rangle \, \big\rangle. \tag{5}$$

We recall, without proof, the following basic properties of draw distributions, mostly from [13], expressed in terms of channels.

Proposition 3 1. *$flrn \odot hg[K] = flrn$;*

2. *$flrn \odot mn[K] = sam$, where $sam \colon \mathcal{D}(X) \rightsquigarrow X$ is the identity map, considered as channel;*
3. *$hg[K] \odot mn[K{+}L] = mn[K]$;*
4. *$hg[K] \odot hg[K{+}L] = hg[K]$;*
5. *$hg[K] \odot DD = hg[K]$;*

6. $DD \circledcirc hg[K+1] = hg[K]$;
7. $DD \circledcirc mn[K+1] = mn[K]$;
8. $DD \circledcirc pol[K+1] = pol[K]$. ◀

The last two items express that multinomial and Pólya form cones for the infinite chain of draw-delete channels that appears in a categorical perspective on De Finetti's theorem, see [15] and [16]. This fails in the hypergeometric case since the draw size K must remain smaller than the size of the urn.

4 Continuous distributions

In the previous section, we have seen finite discrete probability distributions over an arbitrary set. There are also continuous distributions, defined on measurable spaces. Here we need such distributions only on one particular kind of spaces, namely on simplices Δ^n, see (4). The only distributions that we need are given by (polynomial) functions $f\colon \Delta^n \to \mathbb{R}$ with $\int_{\Delta^n} f = 1$. Such an f is called a (probability) density function. It gives rise to probability measure Φ that sends a measurable subset $M \subseteq \Delta^n$ to the probability $\int_M f \in [0,1]$. Such measures are elements of the set $\mathscr{G}(\Delta^n)$, where $\mathscr{G}$ is the Giry monad, see *e.g.* [21, 11] for further information. At first we require that such density functions are nonnegative, so $f \geq 0$, but later we drop this requirement, for so-called signed distributions (see the next section).

Definition 4 *Let $\upsilon \in \mathscr{M}_{\mathrm{fs}}(\boldsymbol{n})$ be an urn with full support (for $n \geq 1$).*

1. *It gives rise to the Dirichlet density $dir(\upsilon)\colon \Delta^n \to \mathbb{R}_{\geq 0}$ given on $\boldsymbol{r} \in \Delta^n$ by:*

$$dir(\upsilon)(\boldsymbol{r}) \;:=\; \frac{(\|\upsilon\|-1)!}{(\upsilon-\mathbf{1}_n)\,!\!\!\!\circ} \cdot \prod_{i\in\boldsymbol{n}} r_i^{\upsilon(i)-1} \qquad \textit{where} \qquad \mathbf{1}_n \;:=\; \sum_{0\leq i<n} 1|i\rangle \;\in\; \mathscr{M}_{\mathrm{fs}}(\boldsymbol{n}).$$

   We shall drop the index n from $\mathbf{1}_n$ when it is clear from the context.

2. *The associated probability measure $Dir(\upsilon)$ is defined on measurable subsets $M \subseteq \Delta^n$ as:*

$$Dir(\upsilon)(M) \;:=\; \int_{\boldsymbol{r}\in M} dir(\upsilon)(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}.$$

The function $dir(\upsilon)$ in (1) is a proper probability density because of the following standard equation that explains the form of the Dirichlet normalisation constant, for $\upsilon \in \mathscr{M}_{\mathrm{fs}}(\boldsymbol{n})$.

$$\int_{\boldsymbol{r}\in\Delta^n} \prod_{i\in\boldsymbol{n}} r_i^{\upsilon(i)-1}\,\mathrm{d}\boldsymbol{r} \;=\; \frac{(\upsilon-\mathbf{1})\,!\!\!\!\circ}{(\|\upsilon\|-1)!}. \tag{6}$$

We use Dirichlet for urns υ with positive natural numbers as multiplicities. This can be generalised to urns with positive real numbers as multiplicities — using the Gamma function instead of factorials — but that is not needed in the current setting.

In the sequel we shall use the bind notation $c \ggg \Phi$ also for continuous measures, but in a very restricted form, namely for measures Φ on Δ^n given by a probability density function f and for channels $c\colon \mathscr{D}(\boldsymbol{n}) \rightsquigarrow \mathscr{M}[K](\boldsymbol{n})$. Categorically, this bind is the Kleisli extension for the Giry monad $\mathscr{G}$, see *e.g.* [21, 8] for details. We will not elaborate this background and will simply use the relevant equation, which is of the following form, for $\varphi \in \mathscr{M}[K](\boldsymbol{n})$,

$$c \ggg \Phi := \sum_{\varphi\in\mathscr{M}[K](\boldsymbol{n})} \left(\int_{\boldsymbol{r}\in\Delta^n} c(\boldsymbol{r})(\varphi)\cdot f(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\right)\big|\,\varphi\,\big\rangle, \qquad \text{where we identify } \mathscr{D}(\boldsymbol{n}) \text{ and } \Delta^n. \tag{7}$$

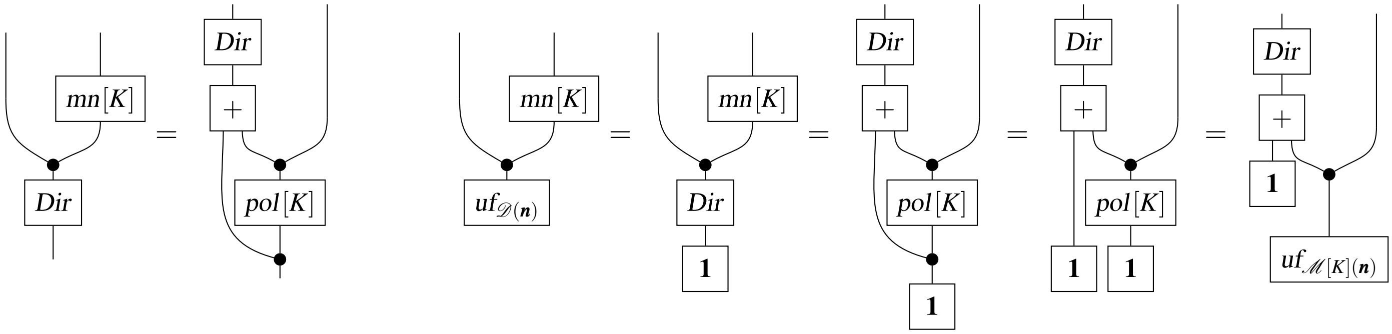


Figure 1: On the left the equation expressing that multinomial is a sufficient statistic for Dirichlet; on the right the string diagrammatic proof that the dagger of Dirichlet is multinomial, with the uniform distribution as prior, see Theorem 5 for details. The boxed $\mathbf{1}$ is the point/singleton distribution $1|\mathbf{1}\rangle$. Such point distributions commute with copiers.

The next result summarises the close relationship between multinomial, Pólya, and Dirichlet distributions. The first two points are well-known, but the third one probably a bit less — although it follows easily from the conjugate prior situation (see also [12]). We use string diagrammatic notation, with flows from bottom to top, since it best displays what is going on, see [8] for details. Proofs are in Appendix B.

Theorem 5 *Let $\upsilon \in \mathscr{M}_{\mathrm{fs}}(\boldsymbol{n})$ be a multiset / urn with full support, of size $L := \|\upsilon\|$, where $n \geq 1$, and let K be an arbitrary number.*

1. *Multinomial over Dirichlet is Pólya: $\mathrm{mn}[K] \gg\!= Dir(\upsilon) = pol[K](\upsilon)$.*
2. *Multinomial is conjugate prior of Dirichlet: updating $\mathrm{Dir}(\upsilon)$ with the predicate / likelihood $\mathrm{mn}[K](-)(\varphi)$ is $\mathrm{Dir}\big(\upsilon+\varphi\big)$. This is expressed diagrammatically on the left in Figure 1.*
3. *Multinomial is the dagger of Dirichlet w.r.t. the uniform distribution $\mathrm{uf}_{\mathscr{D}(\boldsymbol{n})}$ on $\mathscr{D}(\boldsymbol{n})$. In the language / notation of [4, 3] this is expressed as: $\mathrm{mn}[K] = Dir\big(\mathbf{1}+-\big)^{\dagger}_{\mathrm{uf}_{\mathscr{D}(\boldsymbol{n})}}$.*
4. *When we slightly massage the sample channel from Proposition 3 (2) to $\mathrm{sam}\colon \Delta^n \rightsquigarrow \boldsymbol{n}$ given by $\mathrm{sam}(\boldsymbol{r}) = \sum_{i\in\boldsymbol{n}} r_i|\,i\,\rangle$, then: $\mathrm{sam} \gg\!= Dir(\upsilon) = \mathit{flrn}(\upsilon)$.* ◀

The first item of Theorem 5 tells that Pólya is multinomial over Dirichlet. This is an important starting point for this paper, since we asked ourselves the question whether there is also a distribution, like Dirichlet, such that multinomial over it is hypergeometric. We shall see below that the so-called 'signed' Dirichlet distributions achieve this. But first we need to set the scene for these signed distributions.

5 Signed distributions

We now introduce signed distribution, both in the discrete case and in the continuous case. As before, we only need continuous distributions on simplices.

Definition 6 1. *A signed discrete probability distribution on a set X is a function $\sigma\colon X \to \mathbb{R}$ with finite support $\mathrm{supp}(\sigma) := \{x \in X \mid \sigma(x) \neq 0\}$ and with $\sum_{x\in X}\sigma(x) = 1$. We may equivalently write such a signed discrete distribution in ket notation as a finite formal sum $\sum_i r_i|x_i\rangle$ where $r_i \in \mathbb{R}$ satisfy $\sum_i r_i = 1$. We shall write $\mathscr{S}(X)$ for the set of signed discrete probability distributions on X.*

2. *A signed continuous probability distribution, on a simplex Δ^n, is given by a signed density function $f\colon \Delta^n \to \mathbb{R}$ with $\int_{\boldsymbol{r}\in\Delta^n} f(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = 1$.*

An example of a signed discrete distribution is $\frac{1}{2}|a\rangle - \frac{1}{4}|b\rangle + \frac{3}{4}|c\rangle$. We do not offer an operational explanation for what such negative probabilities mean but treat signed distributions as mathematical objects of their own. It is not hard to see that signed discrete distributions $\mathscr{S}$ form a monad on the category of sets and functions. It is affine, in the sense that $\mathscr{S}(\mathbf{1}) \cong \mathbf{1}$, but it differs from $\mathscr{D}$ for instance because it is not *strongly* affine, as defined in [10].

6 Dual bases

The probability mass function of the multinomial distribution is of a particularly tractable form, namely a polynomial function $\Delta^n \to \mathbb{R}$, on the simplex Δ^n.

Definition 7 *For $\varphi \in \mathscr{M}[K](\boldsymbol{n})$, we define the* multinomial $\mathfrak{m}_\varphi$ *as*

$$\mathfrak{m}_\varphi(\boldsymbol{x}) := (\varphi) \cdot \boldsymbol{x}^\varphi = (\varphi) \cdot \prod_{i \in \boldsymbol{n}} x_i^{\varphi(i)} \qquad \text{with 'monomial'} \qquad \boldsymbol{x}^\varphi := \prod_{i \in \boldsymbol{n}} x_i^{\varphi(i)}.$$

For every probability vector $\boldsymbol{r} \in \Delta^n$, one has $\mathfrak{m}_\varphi(\boldsymbol{r}) = mn[K](\boldsymbol{r})(\varphi)$, via the identification $\Delta^n \cong \mathscr{D}(\boldsymbol{n})$.

Definition 8 *For numbers n, K we write $P_K(\Delta^n)$ for the real vector space of polynomial functions $\Delta^n \to \mathbb{R}$ of degree K. The multinomials $\mathfrak{m}_\varphi$ for $\varphi \in \mathscr{M}[K](\boldsymbol{n})$ form a basis of this space, and so we have as dimension $\dim\big(P_K(\Delta^n)\big) = \big(\!\binom{n}{K}\!\big)$. This vector space $P_K(\Delta^n)$ is a Hilbert space via an inner product defined on $f, g \colon \Delta^n \to \mathbb{R}$ as:*

$$\langle f, g \rangle := \int_{\boldsymbol{r} \in \Delta^n} f(\boldsymbol{r}) \cdot g(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}. \tag{8}$$

The dual of a basis (b_i) of a space V is generally understood as a basis of the dual space V^*. In a Hilbert space such a dual basis can be described as the elements (d_i) of the space itself which are uniquely determined by the relationship $\langle b_i, d_j \rangle = \delta_{ij}$, so that $\langle b_i, d_i \rangle = 1$ and $\langle b_i, d_j \rangle = 0$ for $i \neq j$.

Definition 9 *The* dual multinomials $(\mathfrak{d}_\varphi)$ *are defined as the dual basis of $P_K(\Delta^n)$ to the multinomials $(\mathfrak{m}_\varphi)$, and are as such uniquely characterised by the property $\varphi, \psi \in \mathscr{M}[K](\boldsymbol{n})$,*

$$\langle \mathfrak{m}_\varphi, \mathfrak{d}_\psi \rangle = \delta_{\varphi,\psi} = \begin{cases} 1 & \text{if } \varphi = \psi \\ 0 & \text{if } \varphi \neq \psi, \end{cases} \tag{9}$$

What do we know about this dual basis? Of course we can express the dual basis vectors $\mathfrak{d}_\psi$ in terms of the original basis, say via scalars $c_{\chi,\psi}$ satisfying, for each $\psi \in \mathscr{M}[K](\boldsymbol{n})$,

$$\mathfrak{d}_\psi = \sum_{\chi \in \mathscr{M}[K](\boldsymbol{n})} c_{\chi,\psi} \cdot \boldsymbol{x}^\chi = \sum_{\chi \in \mathscr{M}[K](\boldsymbol{n})} \frac{c_{\chi,\psi}}{(\chi)} \cdot \mathfrak{m}_\chi. \tag{10}$$

By exploiting the equations (9) and using the linearity of the inner product in each of its arguments (*i.e.* bilinearity), we obtain the equation

$$\delta_{\varphi,\psi} = \langle \mathfrak{m}_\varphi, \mathfrak{d}_\psi \rangle = \sum_{\chi \in \mathscr{M}[K](\boldsymbol{n})} c_{\chi,\psi} \cdot \frac{\langle \mathfrak{m}_\varphi, \mathfrak{m}_\chi \rangle}{(\chi)}. \tag{11}$$

We note that:

$$\begin{aligned} \langle \mathfrak{m}_\varphi, \mathfrak{m}_\chi \rangle &\overset{(8)}{=} (\varphi) \cdot (\chi) \cdot \int_{\boldsymbol{r} \in \Delta^n} \left(\prod_{i \in \boldsymbol{n}} r_i^{\varphi(i)} \right) \cdot \left(\prod_{i \in \boldsymbol{n}} r_i^{\chi(i)} \right) \mathrm{d}\boldsymbol{r} = (\varphi) \cdot (\chi) \cdot \int_{\boldsymbol{r} \in \Delta^n} \prod_{i \in \boldsymbol{n}} r_i^{(\varphi+\chi)(i)} \, \mathrm{d}\boldsymbol{r} \\ &\overset{(6)}{=} (\varphi) \cdot (\chi) \cdot \frac{(\varphi+\chi)!}{(2K+n-1)!} \end{aligned}$$

There are three square matrices at hand, of size $\left(\!\binom{n}{K}\!\right) \times \left(\!\binom{n}{K}\!\right)$, with multisets as indices, namely:

$$C = \Big(c_{\varphi,\psi}\Big)_{\varphi,\psi\in\mathcal{M}[K](\boldsymbol{n})} \qquad FS = \Big((\varphi+\psi)\,!!\Big)_{\varphi,\psi\in\mathcal{M}[K](\boldsymbol{n})} \qquad D = \Big((\varphi)\cdot\delta_{\varphi,\psi}\Big)_{\varphi,\psi\in\mathcal{M}[K](\boldsymbol{n})}.$$

This C is the matrix of scalars that we are looking for, FS contains the factorials-of-sums of multisets, and D is a diagonal matrix with multiset coefficients. Equation (11) can now be written as:

$$(2K+n-1)!\cdot\delta_{\varphi,\psi} \;=\; \sum_{\chi\in\mathcal{M}[K](\boldsymbol{n})} (\varphi)\cdot FS_{\varphi,\chi}\cdot C_{\chi,\psi} \;=\; \Big(D\cdot FS\cdot C\Big)_{\varphi,\psi}.$$

We then get $(2K+n-1)!\cdot FS^{-1}\cdot D^{-1} = C$, so that the coefficients that we seek are obtained as:

$$c_{\varphi,\psi} \;=\; \frac{(2K+n-1)!}{(\psi)}\cdot\Big(FS^{-1}\Big)_{\varphi,\psi}. \tag{12}$$

These matrix inverses FS^{-1} exist since FS is a symmetric positive definite matrix. We give an extended example calculation in the appendix (Example 19).

The following is a crucial property of dual bases, as introduced in Definition 9.

Proposition 10 *Each dual basis function $\mathfrak{d}_\psi \in P_K(\Delta^n)$, associated with a multiset $\psi\in\mathcal{M}[K](\boldsymbol{n})$, is a continuous signed probability density on Δ^n, that is:*

$$\int_{\boldsymbol{r}\in\Delta^n} \mathfrak{d}_\psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;=\; 1.$$

Proof We use that multinomial distributions $mn[K]$ form a probability distribution; this means that the multinomials $\mathfrak{m}_\varphi$ form a partition of unity, *i.e.* for all $\boldsymbol{r}\in\Delta^n$:

$$\sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} \mathfrak{m}_\varphi(\boldsymbol{r}) \;=\; 1.$$

Hence we obtain

$$\begin{aligned}
\int_{\boldsymbol{r}\in\Delta^n} \mathfrak{d}_\psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;=\; \int_{\boldsymbol{r}\in\Delta^n} 1\cdot\mathfrak{d}_\psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;&=\; \int_{\boldsymbol{r}\in\Delta^n}\Bigg(\sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})}\mathfrak{m}_\varphi(\boldsymbol{r})\Bigg)\cdot\mathfrak{d}_\psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \\
&=\; \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})}\int_{\boldsymbol{r}\in\Delta^n}\mathfrak{m}_\varphi(\boldsymbol{r})\cdot\mathfrak{d}_\psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \\
&\overset{(8)}{=}\; \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})}\big\langle\mathfrak{m}_\varphi,\mathfrak{d}_\psi\big\rangle \overset{(9)}{=} \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})}\delta_{\varphi,\psi} \;=\; 1. \qquad\blacktriangleleft
\end{aligned}$$

In the beginning of this proof, we use that the pointwise sum of the multinomial basis functions $\mathfrak{m}_\varphi$, for $\varphi\in\mathcal{M}[K](\boldsymbol{n})$, is equal to the constant-one function $\mathbf{1}\colon\Delta^n\to\mathbb{R}$. The sum of the dual basis functions $\mathfrak{d}_\varphi$ is also constant.

Proposition 11 *Fix numbers n,K and consider the dual basis function $\mathfrak{d}_\varphi\in P_K(\Delta^n)$, for $\varphi\in\mathcal{M}[K](\boldsymbol{n})$. Then their pointwise sum is a constant function:*

$$\sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})}\mathfrak{d}_\varphi \;=\; \frac{(K+n-1)!}{K!}.$$

Proof Since the vectors $\mathfrak{d}_\varphi$ form a basis we can express the constant-one function $\mathbf{1}\colon \Delta^n \to \mathbb{R}$ with respect to this basis, say as: $\mathbf{1} = \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} a_\varphi \cdot \mathfrak{d}_\varphi$, for certain coefficients a_φ. For a fixed multiset $\psi \in \mathcal{M}[K](\boldsymbol{n})$ we compute the constant a_ψ as follows.

$$\begin{aligned} a_\psi &= \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} a_\varphi \cdot \delta_{\psi,\varphi} = \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} a_\varphi \cdot \langle \mathfrak{m}_\psi, \mathfrak{d}_\varphi \rangle = \langle \mathfrak{m}_\psi, \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} a_\varphi \cdot \mathfrak{d}_\varphi \rangle \\ &= \langle \mathfrak{m}_\psi, \mathbf{1} \rangle = \int_{\boldsymbol{r}\in\Delta^n} \mathfrak{m}_\psi(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} = (\!(\psi)\!) \cdot \int_{\boldsymbol{r}\in\Delta^n} \boldsymbol{r}^\psi(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \overset{(6)}{=} \frac{K!}{\psi!} \cdot \frac{\psi\,!}{(K{+}n{-}1)!} = \frac{K!}{(K{+}n{-}1)!}. \end{aligned}$$

Thus, all these constants a_ψ are the same. As a result:

$$\mathbf{1} = \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} \frac{K!}{(K{+}n{-}1)!} \cdot \mathfrak{d}_\varphi = \frac{K!}{(K{+}n{-}1)!} \cdot \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} \mathfrak{d}_\varphi.$$

By moving the fraction to the other side we are done. ◀

7 Dual Dirichlet and signed hypergeometric

In the previous section we have introduced the dual basis vectors $\mathfrak{d}_\varphi$ as duals to the multinomial vectors $\mathfrak{m}_\varphi$ and have seen that each of these $\mathfrak{d}_\varphi$ forms a signed probability density. We can now start harvesting results, first by defining the associated continuous probability measure.

Definition 12 *Let $\upsilon \in \mathcal{M}(\boldsymbol{n})$ be an multiset (thought of as an urn).*

1. *We write $DDir(\upsilon)$ for the signed probability measure on Δ^n given by the density $\mathfrak{d}_\upsilon$. We call it the* dual Dirichlet *distribution.*
2. *For each number K we define the* signed hypergeometric *channel as the composite with multinomial draws*

$$shg[K] := mn[K] \circ DDir : \mathcal{M}(\boldsymbol{n}) \to \mathcal{S}\big(\mathcal{M}[K](\boldsymbol{n})\big).$$

   This means:

$$shg[K](\upsilon) = mn[K] \gg\!= DDir(\upsilon) = \sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} \left(\int_{\boldsymbol{r}\in\Delta^n} \mathfrak{m}_\varphi(\boldsymbol{r}) \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \right) \big|\, \varphi \,\big\rangle.$$

In Example 20 in Appendix A a signed hypergeometric distribution is computed concretely. In the multivariate case we do not have an explicit formula, as exists in the bivariate case, see Equation (13) in Appendix C for details.

The next result shows that when there is no overdrawing, there is no difference between signed and ordinary hypergeometric probabilities. This means that the dual Dirichlet distribution confirms the question that we originally set ourselves: there is a distribution over which multinomials yield hypergeometric distributions, in analogy with Theorem 5 (1).

Theorem 13 *Let urn $\upsilon \in \mathcal{M}(\boldsymbol{n})$ have size $L = \|\upsilon\|$. Then $shg[K](\upsilon) = hg[K](\upsilon)$, for each $K \leq L$.*

This says that when the size of the draw is at most the size of the urn — so when there are no overdraws — signed hypergeometric coincides with ordinary hypergeometric. In particular, in this case no negative probabilities appear in the signed hypergeometric.

Proof We first note that:

$$1 \overset{(9)}{=} \big\langle \mathfrak{m}_\upsilon, \mathfrak{d}_\upsilon \big\rangle \;=\; \int_{\boldsymbol{r}\in\Delta^n} \mathfrak{m}_\upsilon(\boldsymbol{r}) \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \overset{(7)}{=} \Big(mn[L] \gg\!= DDir(\upsilon)\Big)(\upsilon).$$

As a result: $mn[L] \gg\!= DDir(\upsilon) = 1|\,\upsilon\,\rangle$. By combining this fact with Proposition 3 (3) we are done:

$$\begin{aligned} hg[K](\upsilon) \;=\; hg[K] \gg\!= 1|\,\upsilon\,\rangle \;&=\; hg[K] \gg\!= \Big(mn[L] \gg\!= DDir(\upsilon)\Big) \\ &=\; \Big(hg[K] \circ mn[L]\Big) \gg\!= DDir(\upsilon) \;=\; mn[K] \gg\!= DDir(\upsilon). \end{aligned}$$ ◀

Corollary 14 *Let $\upsilon \in \mathscr{M}[L](\boldsymbol{n})$ be a multiset/urn of size $L \geq 1$.*

1. *For each multiset $\varphi \leq_K \upsilon$, that is, for each $\varphi \in \mathscr{M}[K](\boldsymbol{n})$ with $\varphi \leq \upsilon$, and thus $K \leq L$,*

$$\int_{\boldsymbol{r}\in\Delta^n} \boldsymbol{r}^{\varphi} \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;=\; \frac{\upsilon\,! \cdot (L-K)!}{(\upsilon-\varphi)\,! \cdot L!}.$$

2. *In particular, for each $i \in supp(\upsilon) \subseteq \boldsymbol{n}$.*

$$\int_{\boldsymbol{r}\in\Delta^n} r_i \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;=\; flrn(\upsilon)(i).$$

Proof 1. Since:

$$\begin{aligned} \int_{\boldsymbol{r}\in\Delta^n} \boldsymbol{r}^{\varphi} \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;=\; \frac{1}{\boldsymbol{(}\varphi\boldsymbol{)}} \cdot \int_{\boldsymbol{r}\in\Delta^n} \mathfrak{m}_\varphi(\boldsymbol{r}) \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;&=\; \frac{1}{\boldsymbol{(}\varphi\boldsymbol{)}} \cdot \Big(mn[K] \gg\!= DDir(\upsilon)\Big)(\varphi) \\ &=\; \frac{1}{\boldsymbol{(}\varphi\boldsymbol{)}} \cdot hg[K](\upsilon)(\varphi) \;=\; \frac{\varphi\,!}{K!} \cdot \frac{\binom{\upsilon}{\varphi}}{\binom{L}{K}} \;=\; \frac{\upsilon\,! \cdot (L-K)!}{(\upsilon-\varphi)\,! \cdot L!}. \end{aligned}$$

2. We apply the previous point with $\varphi = 1|\,i\,\rangle$ of size 1. Then:

$$\int_{\boldsymbol{r}\in\Delta^n} r_i \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;=\; \int_{\boldsymbol{r}\in\Delta^n} \boldsymbol{r}^{1|i\rangle} \cdot \mathfrak{d}_\upsilon(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \;=\; \frac{\upsilon\,! \cdot (L-1)!}{(\upsilon - 1|\,i\,\rangle)\,! \cdot L!} \;=\; \frac{\upsilon(i)}{L} \;=\; \frac{\upsilon(i)}{\|\upsilon\|} \;=\; flrn(\upsilon)(i).$$ ◀

Proposition 15 1. *We write $\boldsymbol{0}$ for the empty multiset and $\boldsymbol{1}$ for the multiset of singletons, say on $\boldsymbol{n} = \{0,\ldots,n-1\}$. Then $DDir(\boldsymbol{0}) = Dir(\boldsymbol{1})$ is the uniform measure on Δ^n.*

2. *The sample channel $\mathrm{sam}\colon \Delta^n \to \mathscr{D}(\boldsymbol{n})$ from Theorem 5 (4) gives, like for ordinary Dirichlet,*

$$sam \gg\!= DDir(\upsilon) \;=\; flrn(\upsilon).$$

Proof 1. For $K = 0$ the set $\mathscr{M}[K](\boldsymbol{n})$ of multisets of size 0 contains the empty multiset $\boldsymbol{0}$ as sole element. The associated factor sum matrix FS from (12) is thus the singleton matrix (1), with (1) as inverse. Hence the only coefficient $c_{\boldsymbol{0},\boldsymbol{0}}$ of the polynomial $\mathfrak{d}_{\boldsymbol{0}}$ in (12) is $(n-1)!$. This makes $\mathfrak{d}_{\boldsymbol{0}}$ the constant function $\boldsymbol{r} \mapsto (n-1)!$, which is the density of the uniform measure $Dir(\boldsymbol{1})$ on Δ^n.

2. Using Corollary 14 (2), the reasoning is precisely as in the proof of Theorem 5 (4). ◀

Theorem 13 says that the signed hypergeometric distribution is a 'conservative' extension of the ordinary hypergeometric distribution in the sense that these distributions coincide for draws of size below the size of the urn. We now illustrate draws of arbitrary size, also bigger than the size of the urn. The physical interpretation of such overdraws is unclear. But mathematically all works well.

We continue with some basic properties of signed hypergeometric distributions, as analogues of (some of the items of) Proposition 3.

Proposition 16 *1. $shg[K](\mathbf{0})$ is the uniform distribution on $\mathscr{M}[K](\boldsymbol{n})$;*

2. *$flrn \circledcirc shg[K] = flrn$;*

3. *$shg[K] \circledcirc mn[L{+}K] = mn[K]$;*

4. *$shg[K] \circledcirc shg[K{+}L] = shg[K]$;*

5. *$DD \circledcirc shg[K{+}1] = shg[K]$.*

The last equation shows that the signed hypergeometric form a cone for the draw-delete maps and thus fit in a categorial approach to 'De Finetti', following [15]. The equation $hg[K] \circledcirc DD = hg[K]$ from Proposition 3 (5) holds for ordinary hypergeometric, but its analogue for signed hypergeometric fails.

Proof 1. Via Proposition 15 (1): $shg[K](\mathbf{0}) = mn[K] \ggg DDir(\mathbf{0}) = mn[K] \ggg Dir(\mathbf{1}) = pol[K](\mathbf{1})$. The latter Pólya distribution on $\mathscr{M}[K](\boldsymbol{n})$ is uniform, see Theorem 5.

2. By combining Proposition 3 (2) with Proposition 15 (2) we get: $flrn \circledcirc shg[K] = flrn \circledcirc mn[K] \circledcirc DDir = sam \circledcirc DDir = flrn$.

3. In the composite $shg[K] \circledcirc mn[L{+}K]$ the signed hypergeometric draws of size K are applied to the urns that appear as draws of size $L{+}K$ coming out of the multinomial $mn[L{+}K]$. Hence the draw size is less than the urn size, so Theorem 13 applies, and the signed hypergeometric is an ordinary hypergeometric. Thus Proposition 3 (3) gives: $shg[K] \circledcirc mn[L{+}K] = hg[K] \circledcirc mn[L{+}K] = mn[K]$.

4. By the previous point: $shg[K] \circledcirc shg[L{+}K] = shg[K] \circledcirc mn[L{+}K] \circledcirc DDir = mn[K] \circledcirc DDir = shg[K]$.

5. Via Proposition 3 (7): $DD \circledcirc shg[K{+}1] = DD \circledcirc mn[K{+}1] \circledcirc DDir = mn[K] \circledcirc DDir = shg[K]$. ◀

8 Signed hypergeometric channels as Bayesian inversion

This section shows that the signed hypergeometric channel $shg[K]\colon \mathscr{M}(\boldsymbol{n}) \to \mathscr{S}\big(\mathscr{M}[K](\boldsymbol{n})\big)$ can be obtained as dagger, that is as Bayesian inversion, see [4, 3]. For this we need the following new signed distribution, that builds on the result from Proposition 11 that the sum of dual basis functions is constant.

Definition 17 *For a distribution $\omega \in \mathscr{D}(\boldsymbol{n})$ we define signed* dual multinomial *distribution* $dmn[K](\omega) \in \mathscr{S}\big(\mathscr{M}[K](\boldsymbol{n})\big)$ *as:*

$$dmn[K](\omega) := \sum_{\varphi \in \mathscr{M}[K](\boldsymbol{n})} \frac{K!}{(K{+}n{-}1)!} \cdot \mathfrak{d}_\varphi(\omega) \, \big| \, \varphi \, \big\rangle.$$

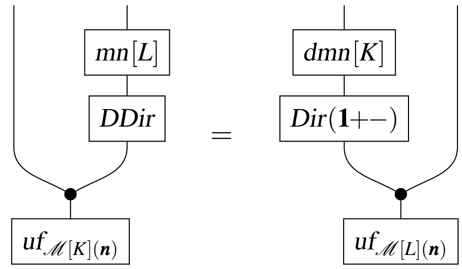
We thus get a 'signed' channel $dmn[K]\colon \mathscr{D}(\boldsymbol{n}) \to \mathscr{S}\big(\mathscr{M}(\boldsymbol{n})\big)$.

For instance:

$$\begin{aligned} & dmn[2]\Big(\tfrac{1}{2}|0\rangle + \tfrac{1}{6}|1\rangle + \tfrac{1}{3}|2\rangle\Big) \\ &= -\tfrac{1}{4}\Big|\, 2|0\rangle \Big\rangle + \tfrac{1}{6}\Big|\, 1|0\rangle + 1|1\rangle \Big\rangle - \tfrac{1}{4}\Big|\, 2|1\rangle \Big\rangle + \tfrac{11}{6}\Big|\, 1|0\rangle + 1|2\rangle \Big\rangle + \tfrac{1}{6}\Big|\, 1|1\rangle + 1|2\rangle \Big\rangle - \tfrac{2}{3}\Big|\, 2|2\rangle \Big\rangle. \end{aligned}$$

We have no (operational) interpretation for these distributions, but they do make sense mathematically, as in the next result.

Theorem 18 *1. There is an equality of string diagrams which involves swapping dual and ordinary Dirichlet distributions*



2. *The signed hypergeometric channel* $shg[L]\colon \mathscr{M}[K](\boldsymbol{n}) \to \mathscr{S}\big(\mathscr{M}[L](\boldsymbol{n})\big)$ *is the dagger of the composite* $dmn[K] \circ Dir(\mathbf{1}{+}{-})\colon \mathscr{M}[L](\boldsymbol{n}) \to \mathscr{S}\big(\mathscr{M}[K](\boldsymbol{n})\big)$ *with the uniform distribution* $\mathrm{uf}_{\mathscr{M}[L](\boldsymbol{n})}$ *as prior. In a formula:*

$$shg[L] \,=\, \Big(dmn[K] \circ Dir(\mathbf{1}{+}{-})\Big)^{\dagger}_{\mathrm{uf}_{\mathscr{M}[L](\boldsymbol{n})}}.$$

Proof 1. Let $\varphi \in \mathscr{M}[K](\boldsymbol{n})$ and $\psi \in \mathscr{M}[L](\boldsymbol{n})$.

$$\begin{aligned}
\Big(\langle dmn[K] \circ Dir(\mathbf{1}{+}{-}), id\rangle \gg\!= \mathrm{uf}_{\mathscr{M}[L](\boldsymbol{n})}\Big)(\varphi,\psi) &= \frac{1}{\left(\!\binom{n}{L}\!\right)} \cdot \int_{\boldsymbol{r}} \frac{K!}{(K{+}n{-}1)!} \cdot \mathfrak{d}_{\varphi}(\omega) \cdot \frac{(L{+}n{-}1)!}{\psi!} \cdot \boldsymbol{r}^{\psi}\, \mathrm{d}\boldsymbol{r} \\
&= \frac{K! \cdot (n{-}1)!}{(K{+}n{-}1)!} \cdot \int_{\boldsymbol{r}} \mathfrak{d}_{\varphi}(\omega) \cdot \frac{L!}{\psi!} \cdot \boldsymbol{r}^{\psi}\, \mathrm{d}\boldsymbol{r} \\
&= \frac{1}{\left(\!\binom{n}{K}\!\right)} \cdot \int_{\boldsymbol{r}} \mathfrak{d}_{\varphi}(\boldsymbol{r}) \cdot \mathfrak{m}_{\psi}(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \\
&= \frac{1}{\left(\!\binom{n}{K}\!\right)} \cdot \Big(mn[L] \gg\!= DDir(\varphi)\Big)(\psi) \\
&= \Big(\langle id\,, mn[L] \circ DDir\rangle \gg\!= \mathrm{uf}_{\mathscr{M}[K](\boldsymbol{n})}\Big)(\varphi,\psi).
\end{aligned}$$

2. This is a reformulation of the previous point, using that $shg[L] = mn[L] \circ DDir$, occurring in the above string diagram on the left of the equation. ◀

In Definition 12 we have introduced the signed hypergeometric $shg[L]$ as $mn[L] \circ DDir$ via the dual Dirichlet distribution $DDir$. The above result tells that we can also obtain $shg[L]$ as dagger from ordinary Dirichlet Dir (plus the dual multinomial distribution dmn). This diagrammatic description of the dagger coincides with the one on right in Figure 1.

9 Conclusions and further work

This paper covers a fascinating topic, namely negative probabilities. It does not offer operational meaning, but it does provide a solid mathematical basis for the emergence of negative probabilities in classical, non-quantum probability theory. The techniques of categorical probability provide the toolbox for describing the relevant properties.

There is plenty of further work. High on our list is an explicit formula for the dual basis vectors in the general multivariate case (like in the bivariate case). Also, we would like to develop a deeper understanding of the conjugate situation described in the string diagrams in Theorem 18.

References

[1] S. Abramsky & A. Brandenburger (2014): *An Operational Interpretation of Negative Probabilities and No-Signalling Models*. In F. van Breugel, E. Kashefi, C. Palamidessi & J. Rutten, editors: *Horizons of the Mind. A Tribute to Prakash Panangaden*, *Lect. Notes Comp. Sci.* 8464, Springer, Berlin, pp. 59–75, doi:10.1007/978-3-319-06880-0_3.

[2] A. Blass & Y. Gurevich (2021): *Negative probabilities: what are they for?* *Journal of Physics A: Mathematical and Theoretical* 54(31), p. 315303, doi:10.1088/1751-8121/abef4d.

[3] K. Cho & B. Jacobs (2019): *Disintegration and Bayesian inversion via string diagrams*. *Math. Struct. in Comp. Sci.* 29(7), pp. 938–971, doi:10.1017/s0960129518000488.

[4] F. Clerc, F. Dahlqvist, V. Danos & I. Garnier (2017): *Pointless learning*. In J. Esparza & A. Murawski, editors: *Foundations of Software Science and Computation Structures*, *Lect. Notes Comp. Sci.* 10203, Springer, Berlin, pp. 355–369, doi:10.1007/978-3-662-54458-7_21.

[5] P. Diaconis (1977): *Finite forms of De Finetti's theorem on exchangeability*. *Synthese* 36(2), pp. 271–281, doi:10.1007/BF00486116.

[6] W. Dong-Bing (1993): *Dual bases of a Bernstein polynomial basis on simplices*. *Computer aided geometric design* 10(6), pp. 483–489, doi:10.1016/0167-8396(93)90025-X.

[7] R. Feynman (1987): *Negative Probability*. In B. Hiley & F. Peat, editors: *Quantum Implications, Essays in Honor of David Bohm*, Routledge and Kegan Paul, London, pp. 235–246, doi:10.1063/1.2811503.

[8] T. Fritz (2020): *A synthetic approach to Markov kernels, conditional independence, and theorems on sufficient statistics*. *Advances in Math.* 370, p. 107239, doi:10.1016/J.AIM.2020.107239.

[9] J. Hoschek & D. Lasser (1993): *Fundamentals of computer aided geometric design*. AK Peters, Ltd.

[10] B. Jacobs (2016): *Affine Monads and Side-Effect-Freeness*. In I. Hasuo, editor: *Coalgebraic Methods in Computer Science (CMCS 2016)*, *Lect. Notes Comp. Sci.* 9608, Springer, Berlin, pp. 53–72, doi:10.1007/978-3-319-40370-0_5.

[11] B. Jacobs (2018): *From Probability Monads to Commutative Effectuses*. *Journ. of Logical and Algebraic Methods in Programming* 94, pp. 200–237, doi:10.1016/j.jlamp.2016.11.006.

[12] B. Jacobs (2020): *A Channel-Based Perspective on Conjugate Priors*. *Math. Struct. in Comp. Sci.* 30(1), pp. 44–61, doi:10.1017/S0960129519000082.

[13] B. Jacobs (2021): *From Multisets over Distributions to Distributions over Multisets*. In: *Logic in Computer Science*, IEEE, Computer Science Press, doi:10.1109/lics52264.2021.9470678.

[14] B. Jacobs (2022): *Urns & Tubes*. *Compositionality* 4(4), doi:10.32408/compositionality-4-4.

[15] B. Jacobs & S. Staton (2020): *De Finetti's construction as a categorical limit*. In D. Petrişan & J. Rot, editors: *Coalgebraic Methods in Computer Science (CMCS 2020)*, *Lect. Notes Comp. Sci.* 12094, Springer, Berlin, pp. 90–111, doi:10.1007/978-3-030-57201-3_6.

[16] E. Jaynes (1982): *Some applications and extensions of the De Finetti representation theorem*. *Bayesian Inference and Decision Techniques with Applications: Essays in Honor of Bruno de Finetti. North-Holland Publishers.*

[17] B. Jüttler (1998): *The dual basis functions for the Bernstein polynomials*. *Adv. Computational Mathematics* 8, pp. 345–352, doi:10.1023/A:1018912801267.

[18] G. Kerns & G. Székely (2006): *De Finetti's Theorem for abstract finite exchangeable sequences*. *Journal of Theoretical Probability* 19(3), pp. 589–608, doi:10.1007/s10959-006-0028-z.

[19] G. Lorentz (2013): *Bernstein polynomials*. American Mathematical Soc.

[20] G. Meissner & M. Burgin (2011): *Negative Probabilities in Financial Modeling*. *Wilmott Magazine*, doi:10.2139/ssrn.1773077.

[21] P. Panangaden (2009): *Labelled Markov Processes*. Imperial College Press, London, doi:10.1142/p595.

[22] G. Székely (2005): *Half of a Coin: Negative Probabilities*. Wilmott Magazine, pp. 66–68.

[23] H. Tijms & K. Staats (2007): *Negative probabilities at work in the M/D/1 queue*. Probability in the Engineering and Informational Sciences 21(1), pp. 67–76, doi:10.1017/S0269964807070040.

[24] K. Zhao & J. Sun (1988): *Dual bases of multivariate Bernstein-Bézier polynomials*. Computer aided geometric design 5(2), pp. 119–125, doi:10.1016/0167-8396(88)90026-X.

Appendix

A Examples

We elaborate two examples: first we illustrate how to actually compute a dual basis, and then how to compute signed hypergeometric distributions.

Example 19 *We take $n = 3$ and $K = 2$ so that we have $\left(\!\binom{n}{K}\!\right) = \frac{4!}{2!\cdot 2!} = 6$ multisets in $\mathcal{M}[2](\mathbf{3})$, namely: $2|0\rangle$, $1|0\rangle + 1|1\rangle$, $2|1\rangle$, $1|0\rangle + 1|2\rangle$, $1|1\rangle + 1|2\rangle$, $2|2\rangle$. Using this order of multisets we get a 6×6 matrix:*

$$FS \;=\; \Big((\varphi+\psi)\,!\Big)_{\varphi,\psi\in\mathcal{M}[2](\mathbf{3})} \;=\; \begin{pmatrix} 24 & 6 & 4 & 6 & 2 & 4 \\ 6 & 4 & 6 & 2 & 2 & 2 \\ 4 & 6 & 24 & 2 & 6 & 4 \\ 6 & 2 & 2 & 4 & 2 & 6 \\ 2 & 2 & 6 & 2 & 4 & 6 \\ 4 & 2 & 4 & 6 & 6 & 24 \end{pmatrix} \qquad \textit{so} \qquad FS^{-1} \;=\; \frac{1}{60}\begin{pmatrix} 6 & -8 & 1 & -8 & 2 & 1 \\ -8 & 44 & -8 & -6 & -6 & 2 \\ 1 & -8 & 6 & 2 & -8 & 1 \\ -8 & -6 & 2 & 44 & -6 & -8 \\ 2 & -6 & -8 & -6 & 44 & -8 \\ 1 & 2 & 1 & -8 & -8 & 6 \end{pmatrix}$$

Notice that negative values appear in this inverse matrix, without clear pattern.

We can now compute the dual basis vectors by combining (10) *and* (12)*. We elaborate the case of the first multiset $2|0\rangle$. For $(r_0,r_1,r_2) \in \Delta^3$, that is, for $r_0,r_1,r_2 \in [0,1]$ with $r_0+r_1+r_2 = 1$ we get $\mathfrak{d}_{2|0\rangle} \in P_2(\Delta^3)$ determined as:*

$$\begin{aligned} &\mathfrak{d}_{2|0\rangle}(r_0,r_1,r_2) \\ &\overset{(10)}{=} \sum_{\varphi\in\mathcal{M}[2](\mathbf{3})} c_{\varphi,2|0\rangle}\cdot(r_0,r_1,r_2)^{\varphi} \overset{(12)}{=} \sum_{\varphi\in\mathcal{M}[2](\mathbf{3})} \tfrac{6!}{(2|0\rangle)}\cdot\Big(FS^{-1}\Big)_{\varphi,2|0\rangle}\cdot r_0^{\varphi(0)}\cdot r_1^{\varphi(1)}\cdot r_2^{\varphi(2)} \\ &= \; 12\Big(6r_0^2 - 8r_0^1r_1^1 + 1r_1^2 - 8r_0^1r_2^1 + 2r_1^1r_2^1 + 1r_2^2\Big) \;=\; 72r_0^2 - 96r_0r_1 + 12r_1^2 - 96r_0r_2 + 24r_1r_2 + 12r_2^2. \end{aligned}$$

Similarly, for the other multisets in $\mathcal{M}[2](\mathbf{3})$,

$$\begin{aligned} \mathfrak{d}_{1|0\rangle+1|1\rangle}(r_0,r_1,r_2) \;&=\; -48r_0^2 + 264r_0r_1 - 48r_1^2 - 36r_0r_2 - 36r_1r_2 + 12r_2^2 \\ \mathfrak{d}_{2|1\rangle}(r_0,r_1,r_2) \;&=\; 12r_0^2 - 96r_0r_1 + 72r_1^2 + 24r_0r_2 - 96r_1r_2 + 12r_2^2 \\ \mathfrak{d}_{1|0\rangle+1|2\rangle}(r_0,r_1,r_2) \;&=\; -48r_0^2 - 36r_0r_1 + 12r_1^2 + 264r_0r_2 - 36r_1r_2 - 48r_2^2 \\ \mathfrak{d}_{1|1\rangle+1|2\rangle}(r_0,r_1,r_2) \;&=\; 12r_0^2 - 36r_0r_1 - 48r_1^2 - 36r_0r_2 + 264r_1r_2 - 48r_2^2 \\ \mathfrak{d}_{2|2\rangle}(r_0,r_1,r_2) \;&=\; 12r_0^2 + 24r_0r_1 + 12r_1^2 - 96r_0r_2 - 96r_1r_2 + 72r_2^2. \end{aligned}$$

Then indeed, $\big\langle \mathfrak{m}_\varphi, \mathfrak{d}_\psi \big\rangle = \delta_{\varphi,\psi}$ for all $\varphi,\psi \in \mathcal{M}[2](\mathbf{3})$.

We check the claim of Proposition 11, that the sum of the dual base vectors is a particular constant. Let's abbreviate the (pointwise) sum of the above dual basis functions as: $\mathfrak{d} := \sum_{\varphi \in \mathcal{M}[2](\mathbf{3})} \mathfrak{d}_{\varphi}$. *Then, using the above descriptions, for* $(r_0, r_1, r_2) \in \Delta^3$,

$$\begin{aligned}\mathfrak{d}(r_0,r_1,r_2) &= 12r_0^2 + 24r_0r_1 + 12r_1^2 + 24r_0r_2 + 24r_1r_2 + 12r_2^2 \\ &= 12\Big((r_0+r_1)^2 + 2(r_0+r_1)r_2 + r_2^2\Big) = 12\big(r_0+r_1+r_2\big)^2 = 12.\end{aligned}$$

Hence the sum of the dual basis functions is indeed a constant and equals $\frac{(K+n-1)!}{K!} = \frac{4!}{2!} = 12$.

Example 20 *Let's take as urn* $\upsilon = 1|0\rangle + 1|1\rangle + 1|2\rangle$ *with one ball of each color in* $\mathbf{3} = \{0,1,2\}$. *By drawing two balls we remain within the world of ordinary hypergeometric distributions, by Theorem 13 and get the following distribution over draws.*

$$shg[2](\upsilon) = hg[2](\upsilon) = \tfrac{1}{3}\big|1|0\rangle + 1|1\rangle\big\rangle + \tfrac{1}{3}\big|1|0\rangle + 1|2\rangle\big\rangle + \tfrac{1}{3}\big|1|1\rangle + 1|2\rangle\big\rangle.$$

We can also draw three balls; in that case, the outcome is certain to be the whole urn:

$$shg[3](\upsilon) = hg[3](\upsilon) = 1\big|1|0\rangle + 1|1\rangle + 2|1\rangle\big\rangle = 1\big|\upsilon\big\rangle.$$

Drawing four balls, more than in the urn, is only possible with the signed hypergeometric. It leads to negative probabilities in:

$$\begin{aligned}shg[4](\upsilon) = {}& \tfrac{7}{126}\big|4|0\rangle\big\rangle - \tfrac{14}{126}\big|3|0\rangle + 1|1\rangle\big\rangle - \tfrac{21}{126}\big|2|0\rangle + 2|1\rangle\big\rangle - \tfrac{14}{126}\big|1|0\rangle + 3|1\rangle\big\rangle + \tfrac{7}{126}\big|4|1\rangle\big\rangle \\ & - \tfrac{14}{126}\big|3|0\rangle + 1|2\rangle\big\rangle + \tfrac{84}{126}\big|2|0\rangle + 1|1\rangle + 1|2\rangle\big\rangle + \tfrac{84}{126}\big|1|0\rangle + 2|1\rangle + 1|2\rangle\big\rangle \\ & - \tfrac{14}{126}\big|3|1\rangle + 1|2\rangle\big\rangle - \tfrac{21}{126}\big|2|0\rangle + 2|2\rangle\big\rangle + \tfrac{84}{126}\big|1|0\rangle + 1|1\rangle + 2|2\rangle\big\rangle \\ & - \tfrac{21}{126}\big|2|1\rangle + 2|2\rangle\big\rangle - \tfrac{14}{126}\big|1|0\rangle + 3|2\rangle\big\rangle - \tfrac{14}{126}\big|1|1\rangle + 3|2\rangle\big\rangle + \tfrac{7}{126}\big|4|2\rangle\big\rangle.\end{aligned}$$

There is no apparent 'logic' in these probabilities, for instance in terms of draw probabilities. In particular, an intuitive explanation is missing for why certain probabilities are negative. But, as explained above, these distributions are constructed in a systematic manner, via dual bases.

Clarification: the outcomes of the signed hypergeometric given above are obtained via elementary Python scripts that perform matrix inversion — as in (12) *— to obtain representations of dual basis vectors and to perform integration over simplices. Our Python scripts produce real numbers as probabilities, but they are so close to the above fractions in* $shg[4](\upsilon)$ *that we write these fractions instead, for the sake of readability.*

B Missing proofs

In the body of the article we left out the proofs of several basis properties of ordinary draws.

Proof (of Theorem 5)

1. Via (7), for arbitrary $\varphi \in \mathcal{M}[K](\boldsymbol{n})$,

$$
\begin{aligned}
\big(mn[K] \gg\!= Dir(\upsilon)\big)(\varphi) &= \int_{\boldsymbol{r}\in\Delta^n} mn[K](\boldsymbol{r})(\varphi) \cdot dir(\upsilon)(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \\
&= \int_{\boldsymbol{r}\in\Delta^n} (\varphi) \cdot \prod_{i\in\boldsymbol{n}} r_i^{\varphi(i)} \cdot \frac{(L-1)!}{(\upsilon-\mathbf{1})\underset{\circ}{!}} \cdot \prod_{i\in\boldsymbol{n}} r_i^{\upsilon(i)-1}\, \mathrm{d}\boldsymbol{r} \\
&= \int_{\boldsymbol{r}\in\Delta^n} \frac{K! \cdot (L-1)!}{\varphi\underset{\circ}{!} \cdot (\upsilon-\mathbf{1})\underset{\circ}{!}} \cdot \prod_{i\in\boldsymbol{n}} r_i^{\upsilon(i)+\varphi(i)-1}\, \mathrm{d}\boldsymbol{r} \\
&= \frac{\left(\!\binom{\upsilon}{\varphi}\!\right)}{\left(\!\binom{L}{K}\!\right)} \cdot \int_{\boldsymbol{r}\in\Delta^n} \frac{(L+K-1)!}{(\upsilon+\varphi-\mathbf{1})\underset{\circ}{!}} \cdot \prod_{i\in\boldsymbol{n}} r_i^{(\upsilon+\varphi)(i)-1}\, \mathrm{d}\boldsymbol{r} \\
&\overset{(6)}{=} pol[K](\upsilon)(\varphi).
\end{aligned}
$$

2. This can be computed in a similar manner, but the details are beyond the scope of this paper.

3. We have already seen that the set $\mathcal{M}[K](\boldsymbol{n})$ has $\left(\!\binom{n}{K}\!\right)$ elements, so that the uniform distribution $uf_{\mathcal{M}[K](\boldsymbol{n})}$ on this set is $\sum_{\varphi\in\mathcal{M}[K](\boldsymbol{n})} \frac{1}{\left(\!\binom{n}{K}\!\right)} \big|\varphi\big\rangle$. This uniform distribution equals $pol[K](\mathbf{1})$, where $\mathbf{1} = \sum_{i\in\boldsymbol{n}} 1|i\rangle$. Hence we can reason diagrammatically, as on the right in Figure 1. The uniform distribution $uf_{\mathcal{D}(\boldsymbol{n})}$ on the left in this chain of equations is $Dir(\mathbf{1})$ on Δ^n, with density $\boldsymbol{r} \mapsto (n-1)!$, since $\int_{\boldsymbol{r}\in\Delta^n} 1\, \mathrm{d}\boldsymbol{r} = \frac{1}{(n-1)!}$ by (6).

4. Consider an urn υ of size L and a number $j \in \boldsymbol{n}$. We write $\upsilon_j := \upsilon + 1|j\rangle$ of size $L+1$. Then, using (7),

$$
\begin{aligned}
\big(sam \gg\!= Dir(\upsilon)\big)(j) &= \int_{\boldsymbol{r}\in\Delta^n} r_j \cdot \frac{(L-1)!}{(\upsilon-\mathbf{1})\underset{\circ}{!}} \cdot \prod_{i\in\boldsymbol{n}} r_i^{\upsilon(i)-1}\, \mathrm{d}\boldsymbol{r} \\
&= \frac{\upsilon(j)}{L} \cdot \int_{\boldsymbol{r}\in\Delta^n} \frac{L!}{(\upsilon_j-\mathbf{1})\underset{\circ}{!}} \cdot \prod_{i\in\boldsymbol{n}} r_i^{\upsilon_j(i)-1}\, \mathrm{d}\boldsymbol{r} \\
&\overset{(6)}{=} \frac{\upsilon(j)}{L} \cdot 1 \\
&= flrn(\upsilon).
\end{aligned}
$$
◀

C The bivariate case

In the previous sections we have elaborated the multivariate case, involving multiple variables. The bivariate case, for $n = 2$, can be done slightly differently, using the isomorphisms $\mathcal{D}(\mathbf{2}) \cong [0,1]$ and $\mathcal{M}[K](\mathbf{2}) \cong \{0,1,\ldots,K\} \cong \boldsymbol{K+1}$. The binomial channel $bn[K]\colon [0,1] \rightsquigarrow \{0,\ldots,K\}$ is thus related via the general multinomial one via the following square.

$$
\begin{tikzcd}
{[0,1]} \arrow[r, "bn{[K]}"] \arrow[d, "\cong"] & \{0,\ldots,K\} \\
\mathcal{D}(\mathbf{2}) \arrow[r, "mn{[K]}"] & \mathcal{M}[K](\mathbf{2}) \arrow[u, "\cong"']
\end{tikzcd}
$$

Explicitly, for $r \in [0,1]$, we have

$$bn[K](r) \;=\; \sum_{0 \le i \le K} \binom{K}{i} \cdot r^i \cdot (1-r)^{K-i} \big| i \big\rangle.$$

The polynomials involved are known as *Bernstein polynomials*, namely

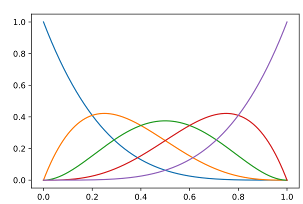
$$\begin{aligned} \mathfrak{b}_i(r) \;\coloneqq\; \binom{K}{i} \cdot r^i \cdot (1-r)^{K-i} \;&=\; \sum_{0 \le j \le K-i} \binom{K}{i} \cdot \binom{K-i}{j} \cdot r^i \cdot 1^j \cdot (-r)^{K-i-j} \\ &=\; \sum_{0 \le j \le K-i} \binom{K}{i} \cdot \binom{K-i}{j} \cdot (-1)^{K-i-j} \cdot r^{K-j}. \end{aligned}$$

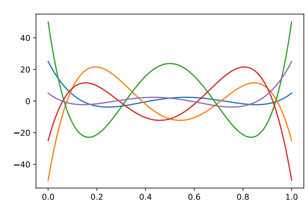
These polynomials are widely studied in Computer Graphics and Computer Aided Geometric Design [9], but also in areas such as approximation theory [19] and probability. They appear not only as probability mass functions for the binomial distributions (as described above), but also as density function of the (continuous) Beta distributions, rescaled by a normalisation factor.

The Hilbert space $P_K(\Delta^2)$ can be identified with the space of univariate polynomials, as functions $[0,1] \to \mathbb{R}$, spanned by the monomial basis $(r^i : 0 \le i \le K)$. The $(n{+}1) \times (n{+}1)$ matrix B that represents the above polynomials $\mathfrak{b}_i$ in this basis has entries

$$B_{i,K-j} \;=\; \binom{K}{i} \cdot \binom{K-i}{j} \cdot (-1)^{K-i-j}.$$

Below we plot several Bernstein polynomials (on the left) and their dual bases (on the right).





We write S for the matrix with inner products of the base vectors, so:

$$S_{i,j} \;\coloneqq\; \langle r^i, r^j \rangle \;=\; \int_{r \in [0,1]} r^i \cdot r^j \,\mathrm{d}r \;=\; \tfrac{1}{i+j+1}.$$

The dual basis of the binomial polynomials $\mathfrak{b}_i$ are then given by the matrix inverse $(B^T \cdot S)^{-1}$. It has been studied in a Computer Graphics context [6] [24], and formulas for computing the dual basis are known, see [17] for an overview.

This explicit formulation of the dual basis allows us to describe the *bivariate signed hypergeometric* channel $bshg[L,K] \colon \{0,\ldots,L\} \to \mathscr{S}\big(\{0,\ldots,K\}\big)$, in the following commuting diagram.

$$\begin{tikzcd} \{0,\ldots,L\}_{\cong} \arrow[r, "bshg{[L,K]}", "\circ" marking] \arrow[d] & \{0,\ldots,K\} \\ \mathscr{M}[L](\mathbf{2}) \arrow[r, "shg{[K]}", "\circ" marking] & \mathscr{M}[K](\mathbf{2})_{-\cong} \arrow[u] \end{tikzcd}$$

We include the parameter L in writing $bshg[L,K]$ since it cannot be derived from an input $j \in \{0,\ldots,L\}$. In contrast, writing this parameter explicitly is not needed in the multivariate case, since the size of the urn can be computed from the urn itself.

The explicit formula for this bivariate signed hypergeometric $bshg[L,K]$ is:

$$\begin{aligned} & bshg[L,K](j) \\ & := \sum_{0\leq i\leq K} \frac{\binom{K}{i}}{(K+L+1)\cdot\binom{L}{j}} \left(\sum_{0\leq \ell\leq L} \frac{(-1)^{j+\ell}}{\binom{K+L}{i+\ell}} \sum_{0\leq k\leq \min(j,\ell)} (2k+1)\binom{L+k+1}{L-j}\binom{L-k}{L-j}\binom{L+k+1}{L-\ell}\binom{L-k}{L-\ell}\right) \big| i \big\rangle \end{aligned} \tag{13}$$

This $bshg[L,K]$ is defined on $0 \leq j \leq L$, corresponding to urn $j|0\rangle + (L-j)|1\rangle$. The complicated character of this formula is not helpful for an operational interpretation in terms of draw probabilities. But it does allow us to compute (truely) exact distributions.

Example 21 *Let's take an urn size $L = 3$ with 2 balls of colour 0. Thus, in a multivariate scenario we would write this as urn $\upsilon = 2|0\rangle + 1|1\rangle$. We first look at a draw of size $K = 4$. Using the formula from Figure C we get a bivariate signed hypergeometric distribution of the form:*

$$bshg[3,4](2) = \tfrac{17}{210}\big|0\big\rangle - \tfrac{34}{105}\big|1\big\rangle + \tfrac{17}{35}\big|2\big\rangle + \tfrac{106}{105}\big|3\big\rangle - \tfrac{53}{210}\big|4\big\rangle.$$

The number i in $|i\rangle$ refers to the number of balls of colour 0 that are drawn (out of K in total), with corresponding (positive or negative) probability.

Similarly, for $K = 5$ we have:

$$bshg[3,5](2) = \tfrac{1}{6}\big|0\big\rangle - \tfrac{3}{7}\big|1\big\rangle + \tfrac{1}{21}\big|2\big\rangle + \tfrac{16}{21}\big|3\big\rangle + \tfrac{37}{42}\big|4\big\rangle - \tfrac{3}{7}\big|5\big\rangle.$$

Effectful Semantics in 2-Dimensional Categories: Premonoidal and Freyd Bicategories

Hugo Paquet
LIPN, Université Sorbonne Paris Nord
Paris, France
paquet@lipn.univ-paris13.fr

Philip Saville
University of Oxford
Oxford, UK
philip.saville@cs.ox.ac.uk

Premonoidal categories and Freyd categories provide an encompassing framework for the semantics of call-by-value programming languages. Premonoidal categories are a weakening of monoidal categories in which the interchange law for the tensor product may not hold, modelling the fact that effectful programs cannot generally be re-ordered. A Freyd category is a pair of categories with the same objects: a premonoidal category of general programs, and a monoidal category of 'effect-free' programs which do admit re-ordering.

Certain recent innovations in semantics, however, have produced models which are not categories but bicategories. Here we develop the theory to capture such examples by introducing premonoidal and Freyd structure in a bicategorical setting. The second dimension introduces new subtleties, so we verify our definitions with several examples and a correspondence theorem—between Freyd bicategories and certain actions of monoidal bicategories—which parallels the categorical framework.

1 Introduction

A fundamental aspect of call-by-value functional programming languages is the distinction between *values* and *computations*. While values are 'pure' program fragments that can be passed around safely, computations may interact with their environment in the form of *effects* (such as raising exceptions, interacting with state, or behaving probabilistically), and must therefore be manipulated with care.

Values and computations obey different algebraic properties, and in particular only computations are sensitive to the evaluation order. For instance `print "a"; print "b"` is not equivalent to `print "b"; print "a"`. This is reflected in the denotational semantics of call-by-value languages, which consists of a pair of categories: a monoidal category of values, and a *premonoidal* category of computations. These are related by an identity-on-objects functor coercing values into effect-free computations, and the resulting structure is called a *Freyd category* ([38, 26]).

In this paper we generalize these notions from categories to bicategories. The resulting theory includes models of programming languages in which the morphisms are themselves objects with structure—spans, strategies, parameter spaces, profunctors, open systems, *etc.*—for which the notion of composition uses a universal construction, such as a pullback or a pushout. In these models, the 2-cells play a central role in characterizing the composition operation for morphisms, and additionally provide refined semantic information (see *e.g.* [17, 8, 48, 34, 22]).

1.1 Bicategorical models

A bicategory is a 2-dimensional category in which the associativity and unit laws for the composition of morphisms are replaced by invertible 2-cells satisfying coherence axioms [2]. Bicategories have recently found prominence as models of computational processes: see *e.g.* [31, 6, 12, 1]. We illustrate this with

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 190–209, doi:10.4204/EPTCS.397.12

two simple examples: spans of sets, and graded monads. For reasons of space we have omitted definitions of the basic notions in bicategory theory, such as pseudofunctors, pseudonatural transformations, and modifications. For a textbook account, see *e.g.* [2].

Bicategories of spans. The bicategory **Span**(**Set**) has objects sets and 1-cells $A \rightsquigarrow B$ spans of functions $A \longleftarrow S \longrightarrow B$. We can compose pairs of morphisms $A \longleftarrow S \longrightarrow B$ and $B \longleftarrow R \longrightarrow C$ using a pullback in the category of sets, as on the left below:

$$
\begin{tikzcd}[column sep=small]
& & R \circ S \arrow[dl] \arrow[dr] \arrow[dd, phantom, "\vee", very near start] & & \\
& S \arrow[dl] \arrow[dr] & & R \arrow[dl] \arrow[dr] & \\
A & & B & & C
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=tiny]
& & & \bullet \arrow[dl] \arrow[ddrr] \arrow[d, phantom, "\vee", very near start] & & & \\
& & \bullet \arrow[dl] \arrow[dr] \arrow[d, phantom, "\vee", very near start] & & & & \\
& \bullet \arrow[dl] \arrow[dr] & & \bullet \arrow[dl] \arrow[dr] & & \bullet \arrow[dl] \arrow[dr] & \\
\bullet & & \bullet & & \bullet & & \bullet
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=tiny]
& & & \bullet \arrow[ddll] \arrow[dr] \arrow[d, phantom, "\vee", very near start] & & & \\
& & & & \bullet \arrow[dl] \arrow[dr] \arrow[d, phantom, "\vee", very near start] & & \\
& \bullet \arrow[dl] \arrow[dr] & & \bullet \arrow[dl] \arrow[dr] & & \bullet \arrow[dl] \arrow[dr] & \\
\bullet & & \bullet & & \bullet & & \bullet
\end{tikzcd}
$$

This composition correctly captures a notion of 'plugging together' spans, but is only associative in a weak sense, since the two ways of taking pullbacks (on the right above) are not generally equal. But, by the universal property of pullbacks, they are canonically isomorphic as spans.

Kleisli bicategories for graded monads. For another example we consider monads graded by monoidal categories. Formally, a graded monad on a category $\mathbb{C}$ consists of a monoidal category $(\mathbb{E}, \bullet, I)$ of *grades* and a lax monoidal functor $T : \mathbb{E} \to [\mathbb{C}, \mathbb{C}]$ (see *e.g.* [43, 30, 21]). In particular, this gives a functor $T_e : \mathbb{C} \to \mathbb{C}$ for every $e \in \mathbb{E}$, and natural transformations $\mu_{e,e'} : T_{e'} \circ T_e \Rightarrow T_{e \bullet e'}$ and $\eta : \mathrm{id} \Rightarrow T_I$ corresponding to a multiplication and unit.

Previous Kleisli-like constructions for graded monads have used presheaf-enriched categories (*e.g.* [10, 28]), but there is also a natural bicategorical construction. The objects are those of $\mathbb{C}$ and 1-cells $A \rightsquigarrow B$ consist of a grade e and a map $f : A \to T_e B$ in $\mathbb{C}$. The 2-cells $(e, f) \Rightarrow (e', f')$ are re-gradings: maps $\gamma : e \to e'$ in $\mathbb{E}$ such that $T_\gamma(B) \circ f = f'$. The composition and identities use the multiplication and unit, as for a Kleisli category. But, unless $\mathbb{E}$ is strict monoidal, this operation is only weakly associative and unital.

A concrete instance of this is the **coPara** construction on a monoidal category $\mathbb{C}$ ([9, 5]), equivalently defined as the Kleisli bicategory for the monad graded by $\mathbb{C}$ itself and given by $T_C(A) = A \otimes C$.

The broader context for this work is the recent occurrence of bicategories in the semantics of programming languages. Bicategories of profunctors are now prominent in the analysis of linear logic and the λ-calculus ([7, 11, 22]), and game semantics employs a variety of span-like constructions that compose weakly ([31, 3]). These models have also influenced the development of 2-dimensional type theories ([8, 34]). This paper supports these developments from the perspective of call-by-value languages. (The connection to linear logic explains our insistence on monoidal rather than cartesian Freyd bicategories.)

1.2 Monoidal bicategories

A monoidal bicategory is a bicategory equipped with a unit object and a tensor product which is only weakly associative and unital. In the categorical setting 'weakly' typically means 'up to isomorphism'; in bicategory theory it typically means 'up to *equivalence*'.

Definition 1. *An* equivalence *between objects A and B in a bicategory $\mathscr{B}$ is a pair of 1-cells $f : A \to B$ and $f^\bullet : B \to A$ together with invertible 1-cells $\mathsf{u} : \mathsf{Id}_A \Rightarrow f^\bullet \circ f$ and $\mathsf{c} : f \circ f^\bullet \Rightarrow \mathsf{Id}_B$. This is an* adjoint equivalence *if the witnessing 2-cells u and c satisfy the usual triangle laws for an adjunction (see* e.g. *[24]).*

$$
\begin{tikzcd}
((AB)C)D \arrow[r, "\alpha"] \arrow[d, "\alpha D"'] & (AB)(CD) \arrow[r, "\alpha"] \arrow[d, phantom, "\Uparrow \mathfrak{p}"] & A(B(CD)) \\
(A(BC))D \arrow[rr, "\alpha"'] & & A((BC)D) \arrow[u, "A\alpha"']
\end{tikzcd}
\quad
\begin{tikzcd}
& AB \arrow[d, phantom, "\mathfrak{m}\Uparrow"] & \\
(AI)B \arrow[ur, "\rho B"] \arrow[rr, "\alpha"'] & {} & A(IB) \arrow[ul, "A\lambda"']
\end{tikzcd}
\quad
\begin{tikzcd}
(IA)B \arrow[r, "\alpha"] \arrow[d, "\lambda B"'] \arrow[dr, phantom, "\overset{\mathfrak{l}}{\Rrightarrow}"] & I(AB) \arrow[dl, bend left, "\lambda"] \\
AB &
\end{tikzcd}
\quad
\begin{tikzcd}
(AB)I \arrow[r, "\alpha"] \arrow[dr, bend right, "\rho"'] \arrow[dr, phantom, "\overset{\mathfrak{r}}{\Rrightarrow}"] & A(BI) \arrow[d, "A\rho"] \\
& AB
\end{tikzcd}
$$

Figure 1: The structural modifications of a monoidal bicategory

It is common in bicategory theory for definitions to ask for adjoint equivalences: these are easier to work with and no stronger than asking for just equivalences (see *e.g.* [24, Proposition 1.5.7]).

The bicategorical version of a natural isomorphism is a *pseudonatural (adjoint) equivalence*: a pseudonatural transformation in which each 1-cell component has the structure of an (adjoint) equivalence.

Definition 2 (*e.g.* [45])**.** *A* monoidal bicategory *is a bicategory $\mathscr{B}$ equipped with a pseudofunctor $\otimes : \mathscr{B} \times \mathscr{B} \to \mathscr{B}$ and an object $I \in \mathscr{B}$, together with:*

- *pseudonatural adjoint equivalences α, λ and ρ with components $\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$ (the* associator*), $\lambda_A : I \otimes A \to A$, and $\rho_A : A \otimes I \to A$ (the* unitors*); and*
- *invertible modifications $\mathfrak{p}, \mathfrak{l}, \mathfrak{m}$ and $\mathfrak{r}$ with components as in Figure 1, subject to coherence axioms.*

Monoidal bicategories have a technical algebraic definition but nonetheless arise naturally. For example, the cartesian product on the category **Set** induces a monoidal structure on the bicategory **Span**(**Set**). Many other examples appear in a similar fashion: see [49].

Coherence theorems. A careful reader might observe that the diagrams in Figure 1 are not, strictly speaking, well-typed: for example, the anti-clockwise route around the diagram for $\mathfrak{p}$ could denote $(A\alpha \circ \alpha) \circ \alpha D$ or $A\alpha \circ (\alpha \circ \alpha D)$. This is justified by a suitable *coherence* theorem.

Typically, coherence theorems show that any two parallel 2-cells built out of the structural data are equal. Appropriate coherence theorems apply to bicategories [27], pseudofunctors [14], and (symmetric) monoidal bicategories ([13, 14, 15]). These results justify writing simply $\cong$ for composites of structural data in commutative diagrams of 2-cells, in much the same way as one does for monoidal categories.

As is common in the field, we rely heavily on the coherence of bicategories and pseudofunctors when writing pasting diagrams of 2-cells. We omit all compositors and unitors for pseudofunctors, and ignore the weakness of 1-cell composition. Thus, even though our diagrams do not strictly type-check, coherence guarantees the resulting 2-cell is the same no matter how one fills in the structural details. For example, for a pseudofunctor T on a monoidal bicategory we may write $T\mathfrak{l}$ as a 2-cell of type $T(\lambda B) \Rightarrow T\lambda \circ T\alpha$. For a detailed justification see *e.g.* [14, Remark 4.5] or [40, §2.2].

1.3 Premonoidal categories and Freyd categories

Premonoidal categories generalize monoidal categories in that the tensor product $\otimes$ is only functorial in each argument separately [37]. The lack of a monoidal "interchange law" reflects the fact that one cannot generally re-order the statements of an effectful program, even if the data flow permits it. As a consequence, one can directly model effectful programs in a premonoidal category, in the sense that a typed program $(\Gamma \vdash M : A)$ is modelled directly as an arrow $\Gamma \to A$ and the result of substituting M into another effectful program $(\Delta, x : A \vdash N : B)$ is modelled by the composite $\Delta \otimes \Gamma \xrightarrow{\Delta \otimes M} \Delta \otimes A \xrightarrow{N} B$. Thus, the composition of morphisms in a premonoidal category should be understood as encoding control flow. This is illustrated in Figure 2 using the graphical calculus for premonoidal categories ([20, 39]), where the dashed red line indicates control flow. This direct interpretation contrasts with monadic approaches ([32, 33]), which rely on a monad whose structure may not be reflected in the syntax.

We can axiomatize the morphisms for which interchange does hold. Let $\mathbb{D}$ be a category equipped with functors $A \rtimes (-) : \mathbb{D} \longrightarrow \mathbb{D}$ and $(-) \ltimes B : \mathbb{D} \longrightarrow \mathbb{D}$ for every $A, B \in \mathbb{D}$, such that $A \rtimes B = A \ltimes B$. We write $A \otimes B$, or just AB, for their joint value, and $(\mathbb{D}, \rtimes, \ltimes)$ is called a *binoidal category*. A map $f : A \to A'$ in $\mathbb{D}$ is *central* if the two diagrams

$$\begin{tikzcd} AB \arrow[r, "A \rtimes g"] \arrow[d, "f \ltimes B"'] & AB' \arrow[d, "f \ltimes B'"] \\ A'B \arrow[r, "A' \rtimes g"'] & A'B' \end{tikzcd} \qquad \begin{tikzcd} BA \arrow[r, "g \ltimes A"] \arrow[d, "B \rtimes f"'] & B'A \arrow[d, "B' \rtimes f"] \\ BA' \arrow[r, "g \ltimes A'"'] & B'A' \end{tikzcd} \tag{1}$$

commute for every $g : B \to B'$. Semantically, f corresponds to a computation which may be run at any point without changing the observable result.

A premonoidal category is a binoidal category $(\mathbb{D}, \rtimes, \ltimes)$ with central structural isomorphisms α, λ and ρ similar to those in a monoidal category. Unlike with monoidal categories, however, the associator α cannot be a natural transformation in all arguments simultaneously, because $\otimes$ is not a functor on $\mathbb{D}$. Instead, we must ask for naturality in each argument separately, so the following three diagrams commute:

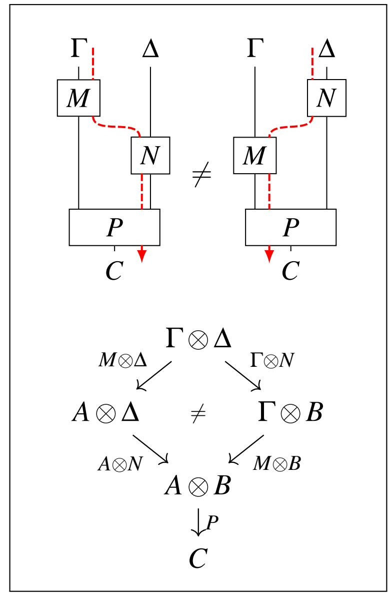


Figure 2: Failure of interchange in a premonoidal category.

$$\begin{tikzcd} (AB)C \arrow[r, "(f \ltimes B) \ltimes C"] \arrow[d, "\alpha"'] & (A'B)C \arrow[d, "\alpha"] \\ A(BC) \arrow[r, "f \ltimes (BC)"'] & A'(BC) \end{tikzcd} \quad \begin{tikzcd} (AB)C \arrow[r, "(A \rtimes g) \ltimes C"] \arrow[d, "\alpha"'] & (AB')C \arrow[d, "\alpha"] \\ A(BC) \arrow[r, "A \rtimes (g \ltimes C)"'] & A(B'C) \end{tikzcd} \quad \begin{tikzcd} (AB)C \arrow[r, "(AB) \rtimes h"] \arrow[d, "\alpha"'] & (AB)C' \arrow[d, "\alpha"] \\ A(BC') \arrow[r, "A \rtimes (B \rtimes h)"'] & A(BC') \end{tikzcd} \tag{2}$$

Definition 3 ([37])**.** *A* premonoidal category *is a binoidal category $(\mathbb{D}, \rtimes, \ltimes)$ equipped with a unit object I and central isomorphisms $\rho_A : AI \to A$, $\lambda_A : IA \to A$ and $\alpha_{A,B,C} : (AB)C \to A(BC)$ for every $A, B, C \in \mathbb{D}$, natural in each argument separately and satisfying the axioms for a monoidal category.*

One important contribution of this paper is to bicategorify the notion of central morphism. We will see that as we move from categories to bicategories centrality evolves from property to structure (Definition 5).

Freyd categories. When modelling call-by-value languages in premonoidal categories, it is natural to think of the values as effect-free computations. Semantically, this is captured by Freyd categories [38], which are premonoidal categories together with a choice of effect-free maps.

Precisely, a Freyd category consists of a monoidal category $\mathbb{V}$ (often *cartesian* monoidal), a premonoidal category $\mathbb{C}$, and an identity-on-objects functor $\mathsf{J} : \mathbb{V} \to \mathbb{C}$ that strictly preserves the tensor product and structural morphisms, and such that every morphism $\mathsf{J}(f)$ is central in $\mathbb{C}$.

Although every premonoidal category $\mathbb{D}$ canonically induces a Freyd category $\mathscr{Z}(\mathbb{D}) \hookrightarrow \mathbb{D}$, where $\mathscr{Z}(\mathbb{D})$ is the subcategory of central maps (called the *centre*), there are several reasons to consider Freyd categories directly. First, it does not always make sense to regard all central maps as values: for instance, in a language with commutative effects (*e.g.* probability), *all* computations are central. Second, functors between binoidal categories do not in general preserve central maps, whereas morphisms of Freyd categories include a functor between the categories of values specifying how values are sent to values.

Relationship to monad models. Freyd categories encompass the strong monad semantics of call-by-value proposed by Moggi ([32, 33]). Indeed, if $(\mathbb{C}, \otimes, I)$ is symmetric monoidal, then any strength for a monad (T, μ, η) on $\mathbb{C}$ induces a premonoidal structure on the Kleisli category $\mathbb{C}_T$, and $\eta \circ (-) : \mathbb{C} \to \mathbb{C}_T$ becomes a Freyd category. Conversely, a Freyd category corresponds to a monad whenever J has a right adjoint [37]. This adjoint is necessary if the programming language has higher-order functions, but some 'first-order' Freyd categories are not known to arise from a monad (*e.g.* [47, 36, 44]).

1.4 Contributions and outline

The central aim of this paper is to introduce definitions of premonoidal bicategories (Definition 6) and Freyd bicategories (Definition 16). Premonoidal structure relies on an adequate notion of centrality for 1-cells and 2-cells in a bicategory (Definition 5). Freyd bicategories then require a coherent assignment of centrality data, which leads to subtle compatibility issues, outlined in Section 2 and Section 3.

As ever with bicategorical definitions (see *e.g.* [40, §2.1]), the main difficulty is in ensuring the right axioms on the 2-cells. We therefore give further justification for our definitions. On the one hand, we show that our definitions are not too strict: they capture natural examples, presented in Section 2.1 and Section 3.1. On the other hand, we show that our definitions are not too weak: the well-known correspondence between Freyd categories and actions [25] lifts to our setting (Section 4). We note that our definition of action is extracted from standard higher-categorical constructions, and so our work connects to an already-existing and well-understood body of theory.

The definition of premonoidal bicategory presented here is based on that in the ArXiv preprint [35]. For reasons of space, we sketch only the proof of the main theorem Theorem 23 here. For more proofs, see the longer version of this paper, available on the authors' webpages.

2 Premonoidal bicategories

Just as in the categorical setting (*e.g.* [37]), our starting point is *binoidal* structure.

Definition 4. *A* binoidal bicategory $(\mathscr{B}, \rtimes, \ltimes)$ *is a bicategory $\mathscr{B}$ with pseudofunctors $A \rtimes (-)$ and $(-) \ltimes B$ for every $A, B \in \mathscr{B}$, such that $A \rtimes B = A \ltimes B$. We write $A \otimes B$, or just AB, for the joint value on objects.*

As is standard when moving from categories to bicategories, the category-theoretic property of centrality becomes extra structure in a binoidal bicategory. For the definition, we observe that the diagrams defining centrality (1) amount to requiring that f induces two natural transformations:

$$
\begin{aligned}
\mathsf{lc}^f : A \rtimes (-) \Rightarrow A' \rtimes (-) \qquad &, \qquad \mathsf{lc}^f_B := \left(A \rtimes B = A \ltimes B \xrightarrow{f \ltimes B} A' \ltimes B = A' \rtimes B\right) \\
\mathsf{rc}^f : (-) \ltimes A \Rightarrow (-) \ltimes A' \qquad &, \qquad \mathsf{rc}^f_B := \left(B \ltimes A = B \rtimes A \xrightarrow{B \rtimes f} B \rtimes A' = B \ltimes A'\right)
\end{aligned}
\tag{3}
$$

This lifts naturally to the bicategorical setting, and gives an immediate notion of centrality for 2-cells.

Definition 5. *Let $(\mathscr{B}, \rtimes, \ltimes)$ be a binoidal bicategory. A* central 1-cell *is a 1-cell $f : A \to A'$ equipped with invertible 2-cells as on the right for every $g : B \to B'$, such that the 1-cells in (3) are the components of pseudonatural transformations $\mathsf{lc}^f : A \rtimes (-) \Rightarrow A' \rtimes (-)$ and $\mathsf{rc}^f : (-) \ltimes A \Rightarrow (-) \ltimes A'$. A* central 2-cell *$\sigma$ between central 1-cells $(f, \mathsf{lc}^f, \mathsf{rc}^f)$ and $(f', \mathsf{lc}^{f'}, \mathsf{rc}^{f'})$ is a 2-cell $\sigma : f \Rightarrow f'$ such that the 2-cells $\sigma \ltimes B$ and $B \rtimes \sigma$ (for $B \in \mathscr{B}$) define modifications $\mathsf{lc}^f \Rightarrow \mathsf{lc}^{f'}$ and $\mathsf{rc}^f \Rightarrow \mathsf{rc}^{f'}$, respectively.*

$$
\begin{tikzcd}
AB \arrow[r, "A \rtimes g"] \arrow[d, "f \ltimes B"'] \arrow[dr, phantom, "\mathsf{lc}^f_g"] & AB' \arrow[d, "f \ltimes B'"] \\
A'B \arrow[r, "A' \rtimes g"'] & A'B'
\end{tikzcd}
\qquad
\begin{tikzcd}
BA \arrow[r, "g \ltimes A"] \arrow[d, "B \rtimes f"'] \arrow[dr, phantom, "\mathsf{rc}^f_g"] & B'A \arrow[d, "B' \rtimes f"] \\
BA' \arrow[r, "g \ltimes A'"'] & B'A'
\end{tikzcd}
$$

Every monoidal bicategory $(\mathscr{B}, \otimes, I)$ has a canonical binoidal structure, with $\rtimes$ and $\ltimes$ directly induced from the monoidal structure by fixing one argument. Every 1-cell f in $\mathscr{B}$ is canonically central, with lc_g^f given by the interchange isomorphism induced by the pseudofunctor structure of $\otimes$, and rc_g^f by $(\mathsf{lc}_f^g)^{-1}$:

$$\mathsf{lc}_g^f := \big((f \otimes B') \circ (A \otimes g) \overset{\cong}{\Longrightarrow} (f \otimes g) \overset{\cong}{\Longrightarrow} (A' \otimes g) \circ (f \otimes B)\big). \tag{4}$$

By the functoriality of $\otimes$, every 2-cell is central with respect to this structure.

We will define premonoidal bicategories as binoidal bicategories with central structural equivalences. As in Definition 3, the associator α for the tensor product can only be pseudonatural in each argument separately, because $\otimes$ is not a functor of two arguments. We therefore need a family of equivalences $\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$ together with invertible 2-cells $\overline{\alpha}_{f,B,C}, \overline{\alpha}_{A,g,C}$ and $\overline{\alpha}_{A,B,h}$ filling the three squares in (5), so that we get three families of pseudonatural transformations:

$$\begin{aligned}
(\alpha_{-,B,C}, \overline{\alpha}_{-,B,C}) &: (- \ltimes B) \ltimes C \Rightarrow (-) \ltimes (B \otimes C) \\
(\alpha_{A,-,C}, \overline{\alpha}_{A,-,C}) &: (A \rtimes -) \ltimes C \Rightarrow A \rtimes (- \ltimes C) \\
(\alpha_{A,B,-}, \overline{\alpha}_{A,B,-}) &: (A \otimes B) \rtimes (-) \Rightarrow A \rtimes (B \rtimes -)
\end{aligned} \tag{5}$$

A premonoidal bicategory also involves structural modifications corresponding to those of Figure 1. Here the 2-dimensional structure introduces new subtleties. For example, one side of modification $\mathfrak{l}$ in Figure 1 uses the pseudonatural transformation with components $\lambda_A \otimes B : (IA)B \to AB$. For $g : B \to B'$, the 2-cell witnessing pseudonaturality of this transformation is the canonical isomorphism that interchanges λ_A and g. This 2-cell does not exist in a premonoidal bicategory, so instead we must use the centrality witness $\mathsf{lc}_g^{\lambda_A}$ for λ_A. Thus, we define $\mathfrak{l}$ to be a family of 2-cells $\mathfrak{l}_{A,B} : (\lambda_A \ltimes B) \Rightarrow \lambda_{A \otimes B} \circ \alpha_{I,A,B}$, pictured on the left below, inducing modifications in $\mathrm{Hom}(\mathscr{B}, \mathscr{B})$ of both types on the right below:

$$\begin{tikzcd}[column sep=small]
(IA)B \arrow[rr, "\lambda \ltimes B"] \arrow[dr, "\alpha_{I,A,B}"'] & \arrow[d, phantom, "\mathfrak{l}_{A,B}"] & AB \\
& I(AB) \arrow[ur, "\lambda_{A \otimes B}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
(I \rtimes -) \rtimes B \arrow[rr, "\lambda \ltimes B"] \arrow[dr, "\alpha_{I,-,B}"'] & \arrow[d, phantom, "\mathfrak{l}_{-,B}"] & (- \ltimes B) \\
& I \rtimes (- \ltimes B) \arrow[ur, "\lambda_{- \ltimes B}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
(IA) \rtimes (-) \arrow[rr, "\mathsf{lc}^{\lambda}"] \arrow[dr, "\alpha_{I,A,-}"'] & \arrow[d, phantom, "\mathfrak{l}_{A,-}"] & (A \rtimes -) \\
& I \rtimes (A \ltimes -) \arrow[ur, "\lambda_{A \rtimes -}"'] &
\end{tikzcd}$$

Notice that the middle diagram appears exactly as in the definition of a monoidal bicategory; no adjustments are necessary because each transformation is pseudonatural in the open argument without any assumptions of centrality.

Modulo the subtleties just outlined, our main definition is a natural extension of the categorical one. We abuse notation by saying "f is central" to mean f comes with chosen lc^f and rc^f making $(f, \mathsf{lc}^f, \mathsf{rc}^f)$ a central 1-cell and saying "the pseudonatural transformation η is central" to mean each 1-cell component η_A is central.

Definition 6. A premonoidal bicategory *is a binoidal bicategory* $(\mathscr{B}, \rtimes, \ltimes)$ *equipped with a unit object* $I \in \mathscr{B}$*, together with the following data:*

1. *For every* $A \in \mathscr{B}$*, central pseudonatural adjoint equivalences* $\lambda_A : I \rtimes A \to A$ *and* $\rho_A : A \ltimes I \to A$*;*
2. *For every* $A, B, C \in \mathscr{B}$*, an adjoint equivalence* $\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$ *with 2-cells as in* (5) *inducing central pseudonatural equivalences in each component separately;*
3. *For each* $A, B, C, D \in \mathscr{B}$*, invertible central 2-cells* $\mathfrak{p}_{A,B,C,D}, \mathfrak{m}_{A,B}, \mathfrak{l}_{A,B}$ *and* $\mathfrak{r}_{A,B}$*, forming modifications in each argument as in Figure 3 or, if not shown there, as in a monoidal bicategory.*

This data is subject to the same equations between 2-cells as in a monoidal bicategory.

$$
\begin{tikzcd}
 & (-B)(CD) \arrow[dr, "\alpha_{-,B,CD}"] & \\
((-B)C)D \arrow[ur, "\alpha_{-B,C,D}"] \arrow[d, "\alpha_{-,B,C}D"'] & \arrow[u, phantom, "\mathfrak{p}_{-,B,C,D}\Uparrow"] & (-)(B(CD)) \\
(-(BC))D \arrow[rr, "\alpha_{-,BC,D}"'] & & (-)((BC)D) \arrow[u, "\mathsf{rc}^{\alpha}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
 & (AB)(C-) \arrow[dr, "\alpha_{A,B,C-}"] & \\
((AB)C)(-) \arrow[ur, "\alpha_{AB,C,-}"] \arrow[d, "\mathsf{lc}^{\alpha}"'] & \arrow[u, phantom, "\mathfrak{p}_{A,B,C,-}\Uparrow"] & A(B(C-)) \\
(A(BC))(-) \arrow[rr, "\alpha_{A,BC,-}"'] & & A((BC)(-)) \arrow[u, "A\alpha_{B,C,-}"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & (-)B & \\
(-I)B \arrow[ur, bend left, "\rho B"] \arrow[rr, "\alpha_{-,I,B}"'] & \arrow[u, phantom, "\mathfrak{m}_{-,B}\Uparrow"] & (-)(IB) \arrow[ul, bend right, "\mathsf{rc}^{\lambda}"']
\end{tikzcd}
\quad
\begin{tikzcd}
 & A(-) & \\
(AI)(-) \arrow[ur, bend left, "\mathsf{lc}^{\rho}"] \arrow[rr, "\alpha_{A,I,-}"'] & \arrow[u, phantom, "\mathfrak{m}_{A,-}\Uparrow"] & A(I-) \arrow[ul, bend right, "A\lambda"']
\end{tikzcd}
\quad
\begin{tikzcd}
(IA)(-) \arrow[r, "\alpha_{I,A,-}"] \arrow[d, "\mathsf{lc}^{\lambda}"'] \arrow[dr, phantom, "\Rrightarrow \mathfrak{l}_{A,-}"] & I(A-) \arrow[dl, bend left, "\lambda"] \\
A(-) &
\end{tikzcd}
\quad
\begin{tikzcd}
(-B)I \arrow[r, "\alpha_{-,B,I}"] \arrow[dr, bend right, "\rho_{-B}"'] \arrow[dr, phantom, "\mathfrak{r}_{-,B}\Rrightarrow"] & (-)(BI) \arrow[d, "\mathsf{rc}^{\rho}"] \\
 & AB
\end{tikzcd}
$$

Figure 3: Modification axioms for the structural 2-cells of a premonoidal bicategory, where they differ from those of a monoidal bicategory. (To save space we suppress $\rtimes$ and $\ltimes$: these can be inferred.)

Note that we cannot ask for the 2-cell components of the structural transformations to be central: for example, $\overline{\rho}_f$ has type $\rho_{A'} \circ (f \ltimes I) \Rightarrow f \circ \rho_A$, but f may not be a central map. Also note that, although we have changed the conditions for $\mathfrak{p}, \mathfrak{m}, \mathfrak{l}$ and $\mathfrak{r}$ to be modifications, their type as 2-cells has not changed, and thus the equations for a monoidal bicategory are still well-typed.

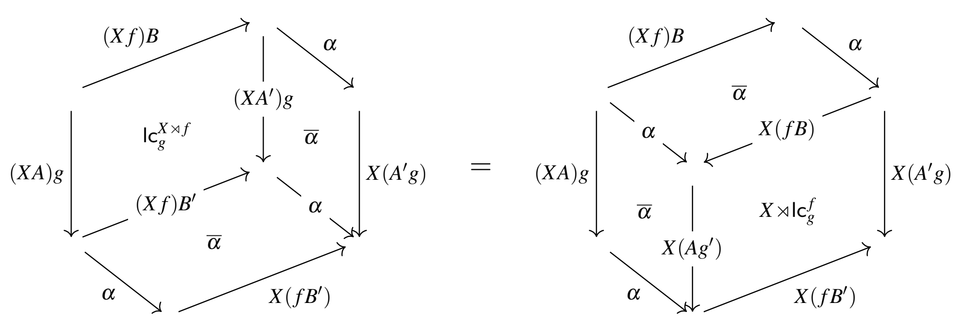
Just as every premonoidal category has a centre, so does every premonoidal bicategory.

Definition 7. *For a premonoidal category $(\mathscr{B}, \rtimes, \ltimes, I)$, denote by $\mathscr{Z}(\mathscr{B})$ the bicategory with the same objects, whose 1-cells and 2-cells are the central 1-cells and central 2-cells in $\mathscr{B}$. Composition is defined using composition in* $\mathrm{Hom}(\mathscr{B},\mathscr{B})$*, and the identity on A is Id_A with the identity transformations.*

The pseudofunctors $A \rtimes (-)$ and $(-) \ltimes B$ lift to the centre. Because $A \rtimes (-)$ is a pseudofunctor, then for any central 1-cell $(f, \mathsf{lc}^f, \mathsf{rc}^f)$ we already have pseudonatural transformations $A \rtimes \mathsf{lc}^f$ and $A \rtimes \mathsf{rc}^f$ in $\mathscr{B}$. To disambiguate between these transformations and the action of $A \rtimes (-)$ on central 1-cells, we denote the latter by $A \rtimes (f, \mathsf{lc}^f, \mathsf{rc}^f) := (A \rtimes f, \mathsf{lc}^{A \rtimes f}, \mathsf{rc}^{A \rtimes f})$, and likewise for $(-) \ltimes B$.

Proposition 8. *Let $(\mathscr{B}, \rtimes, \ltimes, I)$ be a premonoidal bicategory. For every $A, B \in \mathscr{B}$ the operations $A \rtimes (-)$ and $(-) \ltimes B$ induce pseudofunctors on $\mathscr{Z}(\mathscr{B})$.*

Proof sketch. We only sketch the action of $X \rtimes (-)$ and $(-) \ltimes X$ on a central 1-cell $(f, \mathsf{lc}^f, \mathsf{rc}^f) : A \to A'$. For $g : B \to B'$, the 2-cell $\mathsf{lc}_g^{X \rtimes f}$ is uniquely determined by the equation



(6)

in which, for clarity, we have omitted object names and left implicit the functors $\rtimes, \ltimes$, which can be inferred. This is a valid definition for $\mathsf{lc}_g^{X \rtimes f}$ because α is an equivalence, and all 2-cells involved are invertible. We similarly construct 2-cells $\mathsf{lc}_g^{f \ltimes X}$, $\mathsf{rc}_g^{X \rtimes f}$, $\mathsf{rc}_g^{f \ltimes X}$. The rest of the proof consists of routine verifications.

$\square$

2.1 Examples of premonoidal bicategories

State-passing style. Power & Robinson motivate their definition of premonoidal categories by considering an uncurried version of the State monad [37]: for a symmetric monoidal category $(\mathbb{C}, \otimes, I)$ and an object $S \in \mathbb{C}$ modelling a set of states, one can model a program from A to B interacting with the state as a morphism $S \otimes A \to S \otimes B$. The same applies bicategorically.

Lemma 9 (*c.f.* [37], [26, Example A.1])**.** *Let $(\mathscr{B}, \otimes, I)$ be a symmetric monoidal bicategory (*e.g.* [45]) and $S \in \mathscr{B}$. Define a bicategory $\mathscr{K}$ with the same objects as $\mathscr{B}$, hom-categories $\mathscr{K}(A,B) := \mathscr{B}(S \otimes A, S \otimes B)$, and composition and identities as in $\mathscr{B}$. Then $\mathscr{K}$ admits a canonical premonoidal structure.*

For the binoidal structure, one whiskers with the canonical pseudonatural equivalences:

$$\begin{aligned} f \ltimes B &:= \big(S(AB) \xrightarrow{\simeq} (SA)B \xrightarrow{f \otimes B} (SA')B \xrightarrow{\simeq} S(A'B)\big) \\ A \rtimes g &:= \big(S(AB) \xrightarrow{\simeq} A(SB) \xrightarrow{A \otimes g} A(SB') \xrightarrow{\simeq} S(AB')\big) \end{aligned} \tag{7}$$

The structural transformations are then given by composing the structural transformations in $\mathscr{B}$ with the naturality 2-cells for the equivalences in (7).

Bistrong graded monads. It is well-known that if a monad T on a monoidal category $(\mathbb{C}, \otimes, I)$ is *bistrong*, meaning that it is equipped with a left strength $t_{A,B} : A \otimes TB \to T(A \otimes B)$ and a right strength $s_{A,B} : T(A) \otimes B \to T(A \otimes B)$, and these strengths are compatible in the sense that the two canonical maps $(A \otimes T(B)) \otimes C \to T\big(A \otimes (B \otimes C)\big)$ are equal, then $\mathbb{C}_T$ is premonoidal (see *e.g.* [29]). (This definition is obscured in the symmetric setting, because if $\mathbb{C}$ is symmetric every strong monad is canonically bistrong.) A similar fact applies to the Kleisli bicategory $\mathscr{K}_T$ for a graded monad defined in Section 1.1. To state this we need to define bistrong graded monads: we make a small adjustment to Katsumata's definition of strong graded monads [21, Definition 2.5]. An endofunctor $T : \mathbb{C} \to \mathbb{C}$ equipped with two strengths t and s which are compatible in the sense above is called *bistrong* (see *e.g.* [29]).

Definition 10. *A* bistrong graded monad *on a monoidal category $(\mathbb{C}, \otimes, I)$ consists of a monoidal category $(\mathbb{E}, \bullet, I)$ of grades and a lax monoidal functor $T : \mathbb{E} \to [\mathbb{C}, \mathbb{C}]_{bistrong}$, where $[\mathbb{C}, \mathbb{C}]_{bistrong}$ is the category of bistrong endofunctors and natural transformations that commute with both strengths (see* e.g. *[29]).*

Thus, a bistrong graded monad is a graded monad equipped with natural transformations $t^e_{A,B} : A \otimes T_e(B) \to T_e(A \otimes B)$ and $s^e_{A,B} : T_e(A) \otimes B \to T_e(A \otimes B)$ for every grade e, compatible with the graded monad structure and with maps between grades. One then obtains strict pseudofunctors $A \rtimes (-), (-) \ltimes B : \mathscr{K}_T \to \mathscr{K}_T$ for every $A, B \in \mathscr{K}_T$, defined similarly to the premonoidal structure on a Kleisli category:

$$A \rtimes g \;=\; \big(AB \xrightarrow{A \otimes g} AT_e(B') \xrightarrow{t^e} T_e(AB')\big) \quad , \quad f \ltimes B \;=\; \big(AB \xrightarrow{f \otimes B} T_e(A')B \xrightarrow{s^e} T_e(A'B)\big).$$

Moreover, every $f \in \mathbb{C}(A, A')$ determines a 'pure' 1-cell in $\mathscr{K}_T$, as $\widetilde{f} := \big(A \xrightarrow{f} A' \xrightarrow{\eta_{A'}} T_I A'\big)$. This 1-cell canonically determines a central 1-cell, with $\mathsf{lc}^{\widetilde{f}}$ and $\mathsf{rc}^{\widetilde{f}}$ given by the canonical isomorphism in $\mathbb{C}$; in particular, $\mathsf{lc}^{\tilde{f}}_{\tilde{g}} = (\mathsf{rc}^{\tilde{g}}_{\tilde{f}})^{-1}$ for every $g \in \mathbb{C}(B, B')$. The structural transformations are then all of the form $\widetilde{\sigma}$ for σ a structural transformation in $\mathbb{E}$, and the structural modifications are all canonical isomorphisms of the form $I^{\otimes i} \xrightarrow{\cong} I^{\otimes j}$ for $i, j \in \mathbb{N}$. Summarizing, we have the following.

Proposition 11. *Let (T, μ, η) be a bistrong graded monad on $(\mathbb{C}, \otimes, I)$ with grades $(\mathbb{E}, \bullet, I)$. Then the bicategory $\mathscr{K}_T$ has a canonical choice of premonoidal structure.*

***Un*natural transformations.** For any category $\mathbb{C}$ the category $[\mathbb{C},\mathbb{C}]_u$ of functors and *un*natural transformations (*i.e.* families of maps $\sigma_C : FC \to GC$ with no further conditions) is strictly premonoidal. This is almost by definition, because Power & Robinson define a strict premonoidal category to be a monoid with respect to the funny tensor product $\otimes$ on the category **Cat** [37]. A version holds bicategorically.

Lemma 12 (*c.f.* [37])**.** *For any bicategory $\mathscr{B}$, let $[\mathscr{B},\mathscr{B}]_u$ denote the bicategory with objects pseudofunctors $F : \mathscr{B} \to \mathscr{B}$, 1-cells $F \to G$ families of maps $\{\sigma_B : FB \to GB \mid B \in \mathscr{B}\}$, and 2-cells $\sigma \Rightarrow \tau$ families of 2-cells $\{m_B : \sigma_B \Rightarrow \tau_B \mid B \in \mathscr{B}\}$. Then $[\mathscr{B},\mathscr{B}]_u$ admits a premonoidal structure given by composition.*

3 Freyd bicategories

We build up to our definition of Freyd bicategories in stages. Although the bicategories of values and computations have the same objects and their structures are tightly connected, bicategories offer a range of levels of strictness, so we must make careful choices.

We begin with a useful technical notion for relating two pseudofunctors which agree on objects:

Definition 13 ([23])**.** *For pseudofunctors $F, G : \mathscr{B} \to \mathscr{C}$ which agree on objects, an* icon $\theta : F \to G$ *is an oplax natural transformation whose 1-cell components are all identity. More explicitly, θ is a family of 2-cells $\theta_f : F(f) \to G(f)$ indexed by 1-cells of $\mathscr{B}$, subject to naturality, identity and composition laws.*

Using this, we define a notion of strict morphism between binoidal bicategories.

Definition 14. *Let $(\mathscr{V}, \rtimes, \ltimes)$ and $(\mathscr{B}, \rtimes, \ltimes)$ be binoidal bicategories. A* 0-strict binoidal pseudofunctor *is a pseudofunctor $\mathsf{J} : \mathscr{V} \to \mathscr{B}$ together with families of invertible icons θ^A and ζ^A (for $A \in \mathscr{B}$) as on the right; their existence implicitly requires that $\mathsf{J}(A \otimes B) = \mathsf{J}A \otimes \mathsf{J}B$.*

$$\begin{tikzcd}
\mathscr{V} \arrow[r, "(-)\ltimes A"] \arrow[d, "\mathsf{J}"'] \arrow[dr, phantom, "\overset{\zeta^A}{\cong}"] & \mathscr{V} \arrow[d, "\mathsf{J}"] \arrow[dr, phantom, "\overset{\theta^A}{\cong}"] & \mathscr{V} \arrow[l, "A\rtimes(-)"'] \arrow[d, "\mathsf{J}"] \\
\mathscr{B} \arrow[r, "(-)\ltimes \mathsf{J}A"'] & \mathscr{B} & \mathscr{B} \arrow[l, "\mathsf{J}A\rtimes(-)"]
\end{tikzcd}$$

It is crucial that we take preservation up to icons, and not up to identity. In the context of Lemma 9, for instance, we get a 0-strict binoidal pseudofunctor $S \otimes (-) : \mathscr{B} \to \mathscr{K}$ with icons θ and ζ constructed using the pseudonaturality of the equivalences in (7). However, these icons do strictly commute with the premonoidal structure of $\mathscr{K}$ by the coherence of symmetric monoidal bicategories [15]. This suggests the following; for simplicity we focus on the case where J is identity-on-objects.

Definition 15. *Let $(\mathscr{V}, \rtimes, \ltimes, I)$ and $(\mathscr{B}, \rtimes, \ltimes, I)$ be premonoidal bicategories with the same objects and unit I. An* identity-on-objects, 0-strict premonoidal pseudofunctor *$\mathscr{V} \to \mathscr{B}$ is a 0-strict binoidal pseudofunctor $(\mathsf{J}, \theta, \zeta)$ such that J is identity-on-objects and the following axioms hold:*

1. *J strictly preserves the components of the structural transformations: for each $A, B, C \in \mathscr{B}$ we have $\mathsf{J}\alpha_{A,B,C} = \alpha_{A,B,C}, \mathsf{J}\lambda_A = \lambda_A$, and $\mathsf{J}\rho_A = \rho_A$;*
2. *J preserves structural 2-cells up to the icons θ and ζ, according to the axioms in Figure 4.*

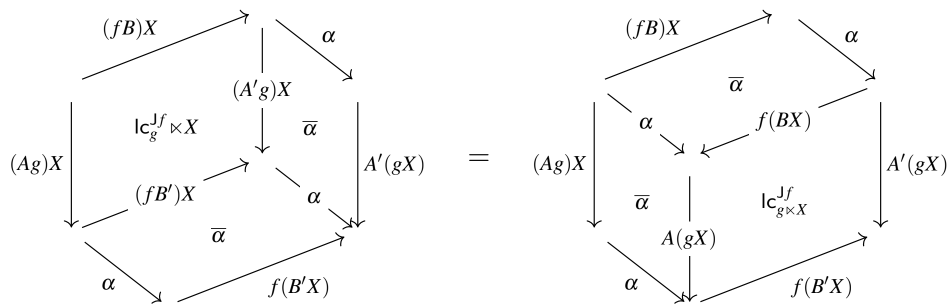
A Freyd bicategory is an identity-on-objects 0-strict premonoidal pseudofunctor from a monoidal bicategory of values to a premonoidal bicategory of computations, together with a choice of centrality witnesses for every value. This choice must be functorial, coherent, and compatible with the interchange law whenever two values are being interchanged. We formalize this in terms of a strict factorization through the centre $\mathscr{Z}(\mathscr{B})$, as is done for Freyd categories [26]. (Unlike for Freyd categories, this factorization is additional structure and not a property of the premonoidal pseudofunctor.)

Definition 16. *A* Freyd bicategory $\mathscr{F}$ *consists of a monoidal bicategory* $(\mathscr{V},\otimes,I)$*, a premonoidal bicategory,* $(\mathscr{B},\rtimes,\ltimes,I)$*, an identity-on-objects, 0-strict premonoidal pseudofunctor* $\mathsf{J}:\mathscr{V}\to\mathscr{B}$*, and a binoidal pseudofunctor* $\mathsf{J}_{\mathscr{Z}}$ *factoring* J *through the centre of* $\mathscr{B}$*, as pictured below,*

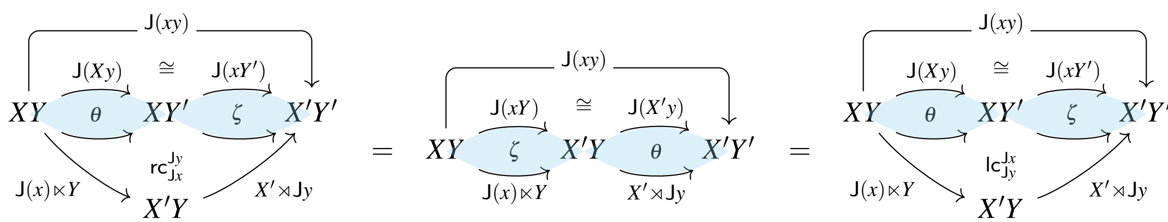
$$\begin{tikzcd} \mathscr{V} \arrow[rr, "\mathsf{J}"] \arrow[dr, dashed, "\mathsf{J}_{\mathscr{Z}}"'] & & \mathscr{B} \\ & \mathscr{Z}(\mathscr{B}) \arrow[ur, "\text{forget}"'] & \end{tikzcd}$$

such that the following axioms hold, where we write $(\mathsf{J}f,\mathsf{lc}^{\mathsf{J}f},\mathsf{rc}^{\mathsf{J}f})$ *for* $\mathsf{J}_{\mathscr{Z}}(f)$*:*

1. *The chosen centrality witnesses for the structural 1-cells agree with those in the premonoidal structure of* $\mathscr{B}$*.*
2. *For each value* $f:A\to A'$*, the chosen* $\mathsf{lc}^{\mathsf{J}f}$ *satisfies the following compatibility law for every* $g\in\mathscr{B}(B,B')$ *and* $X\in\mathscr{B}$*. (This complements the constructions of Proposition 8.)*



3. *For values* $x:X\to X'$ *and* $y:Y\to Y'$*,* $\mathsf{lc}^{\mathsf{J}x}_{\mathsf{J}y}$ *and* $\mathsf{rc}^{\mathsf{J}y}_{Jx}$ *are determined by the interchange law in* $\mathscr{V}$ *(4):*



3.1 Examples of Freyd bicategories

Two of the examples of premonoidal bicategories from Section 2.1 naturally yield Freyd bicategories. First, in the context of Lemma 9, we have a pseudofunctor $S\otimes(-):\mathscr{B}\to\mathscr{K}$ and icons θ and ζ constructed using the equivalences defining the binoidal structure (recall (7)). Moreover, coherence for symmetric monoidal bicategories [15] gives a unique choice of 2-cell for each $\mathsf{lc}^{S\otimes f}_g$ and $\mathsf{rc}^{S\otimes f}_g$, so $S\otimes(-)$ factors through the centre, yielding the following.

Lemma 17. *Let* $(\mathscr{B},\otimes,I)$ *be a symmetric monoidal bicategory and let* $\mathscr{K}$ *be the premonoidal bicategory defined in Lemma 9. Then the pseudofunctor* $S\otimes(-)$ *defines a Freyd bicategory* $\mathscr{B}\to\mathscr{K}$*.*

Similarly, for a bistrong graded monad T, we can think of morphisms in the base monoidal category $\mathbb{C}$ as parameterized maps with trivial parameter space, to construct a Freyd bicategory. The identity-on-objects pseudofunctor has action on morphisms determined by $\mathsf{J}(f):=\widetilde{f}=\eta\circ f$. The structural icons θ and ζ are the identity, and J factors strictly through the centre because every $\widetilde{f}$ has a canonical choice of centrality data.

Proposition 18. *Let (T,η,μ) be a bistrong graded monad on $(\mathbb{C},\otimes,I)$ with grades $(\mathbb{E},\bullet,I)$. Then, writing* $\mathrm{d}\mathbb{C}$ *for the monoidal category $\mathbb{C}$ viewed as a locally-discrete monoidal 2-category, there exists a canonical choice of pseudofunctor J making $\mathsf{J} : \mathrm{d}\mathbb{C} \to \mathscr{K}_T$ a Freyd bicategory.*

Finally, recall the unnatural transformations discussed in Section 2.1: although one could expect the inclusion $\iota : [\mathscr{B},\mathscr{B}] \hookrightarrow [\mathscr{B},\mathscr{B}]_u$ to be a Freyd bicategory, this is not true even in the categorical setting: it is not the case that every natural transformation is central, so ι does not factor through the centre.

4 Freyd bicategories and actions

Freyd categories may equivalently be defined as certain actions of monoidal categories (*e.g.* [25]). In this section we show that this is also possible in the two-dimensional setting.

We first define actions of monoidal bicategories. As observed in [19], a left action on a category is equivalently a bicategory with two objects and certain hom-categories taken to be trivial. We therefore define a left action on a bicategory so it is equivalently a *tricategory* (see [13]) with two objects and certain hom-bicategories taken to be trivial. It follows from the coherence of tricategories ([13, 14]) that every diagram of 2-cells constructed using the structural data of an action must commute.

Definition 19. *A* left action *of a monoidal bicategory $(\mathscr{V},\otimes,I)$ on a bicategory $\mathscr{B}$ consists of a pseudofunctor $\triangleright : \mathscr{V} \times \mathscr{B} \to \mathscr{B}$, together with the following data:*

- *Pseudonatural adjoint equivalences $\widetilde{\lambda}_A : I \triangleright A \to A$ and $\widetilde{\alpha}_{X,Y,C} : (X \otimes Y) \triangleright C \to X \triangleright (Y \triangleright C)$;*
- *Invertible modifications as shown below, satisfying the same coherence axioms as $\mathfrak{p},\mathfrak{m}$, and $\mathfrak{l}$ in a monoidal bicategory (*e.g. *[45]):*

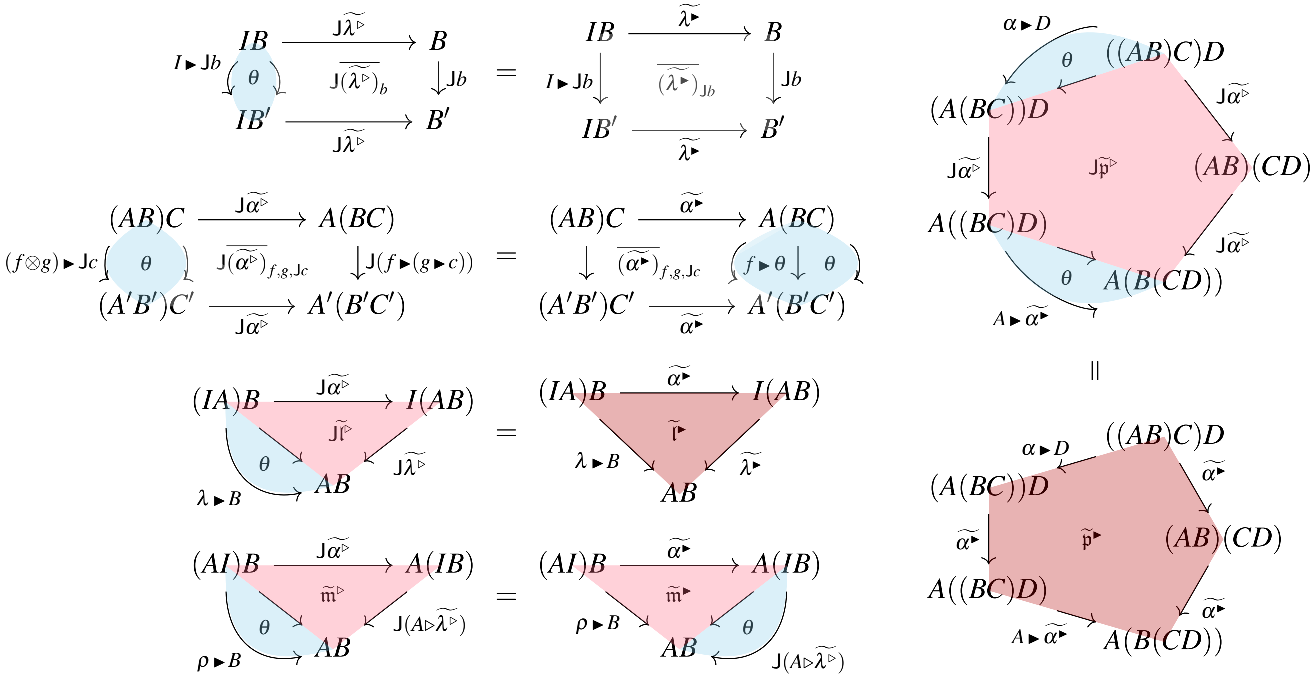
$$\begin{tikzcd}
((XY)Z)\triangleright D \arrow[r, "\widetilde{\alpha}"] \arrow[d, "\alpha\triangleright D"'] & (XY)\triangleright(Z\triangleright D) \arrow[r, "\widetilde{\alpha}"] \arrow[d, phantom, "\widetilde{\mathfrak{p}}\Uparrow"] & X\triangleright(Y\triangleright(Z\triangleright D)) \\
(X(YZ))\triangleright D \arrow[rr, "\widetilde{\alpha}"'] & & X\triangleright((YZ)\triangleright D) \arrow[u, "X\triangleright\widetilde{\alpha}"']
\end{tikzcd}
\quad
\begin{tikzcd}
& X\triangleright C \arrow[d, phantom, "\widetilde{\mathfrak{m}}\Uparrow"] & \\
(XI)\triangleright C \arrow[ur, "\rho\triangleright C"] \arrow[rr, "\widetilde{\alpha}"'] & & X\triangleright(I\triangleright C) \arrow[ul, "X\triangleright\widetilde{\lambda}"']
\end{tikzcd}
\quad
\begin{tikzcd}
(IY)\triangleright C \arrow[r, "\widetilde{\alpha}"] \arrow[d, "\lambda\triangleright C"'] \arrow[dr, phantom, "\overset{\widetilde{\mathfrak{l}}}{\Rrightarrow}"] & I\triangleright(Y\triangleright C) \arrow[dl, bend left, "\widetilde{\lambda}"] \\
Y\triangleright C &
\end{tikzcd}$$

A *right* action $\triangleleft : \mathscr{B} \times \mathscr{V} \to \mathscr{B}$ can be defined analogously, with a right unitor $\widetilde{\rho}_A : A \triangleleft I \to A$, an associator $\widetilde{\alpha}_{A,X,Y} : (A \triangleleft X) \triangleleft Y \to A \triangleleft (X \otimes Y)$, and 2-dimensional structural data.

Every monoidal bicategory $\mathscr{V}$ has canonical left and right actions on itself given by the monoidal data. As we will see, a Freyd bicategory $\mathsf{J} : \mathscr{V} \to \mathscr{B}$ corresponds to a pair of actions $\triangleright : \mathscr{V} \times \mathscr{B} \to \mathscr{B}$ and $\triangleleft : \mathscr{B} \times \mathscr{V} \to \mathscr{B}$ that extend the canonical actions: this mirrors the categorical situation. To that end, we consider a category $\mathscr{V}$-$\mathbf{act}_{0s}$ of actions of $\mathscr{V}$ and identity-on-objects pseudofunctors that preserve the action strictly on objects, but weakly on morphisms. (This is a very special case of a more canonical notion of map between actions.)

Definition 20. *Let $\mathscr{V}$ be a monoidal bicategory and let $(\mathscr{B},\triangleright)$ and $(\mathscr{B}',\blacktriangleright)$ be left actions of $\mathscr{V}$. A* 0-strict morphism of actions *from $(\mathscr{B},\triangleright)$ to $(\mathscr{B}',\blacktriangleright)$ is an identity-on-objects functor $\mathsf{J} : \mathscr{B} \to \mathscr{B}'$ satisfying $\widetilde{\lambda^{\triangleright}_A} = \mathsf{J}(\widetilde{\lambda^{\blacktriangleright}_A})$ and $\widetilde{\alpha^{\triangleright}}_{A,B,C} = \mathsf{J}(\widetilde{\alpha^{\blacktriangleright}}_{A,B,C})$ for every $A,B,C \in \mathscr{B}$, equipped with an icon as on the right, which relates the structural data for the actions according to the axioms below:*

$$\begin{tikzcd}
\mathscr{V}\times\mathscr{B} \arrow[r, "\triangleright"] \arrow[d, "\mathscr{V}\times\mathsf{J}"'] \arrow[dr, phantom, "\theta\Nearrow"] & \mathscr{V} \arrow[d, "\mathsf{J}"] \\
\mathscr{V}\times\mathscr{B}' \arrow[r, "\blacktriangleright"'] & \mathscr{B}
\end{tikzcd}$$



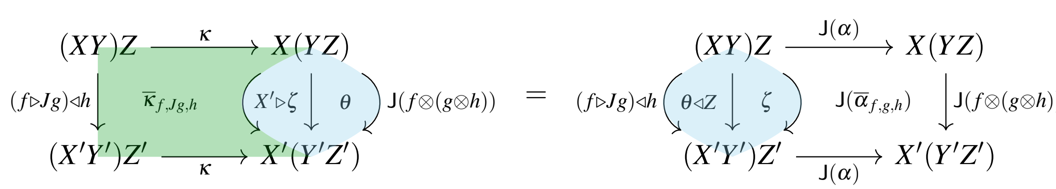
A key example is the following:

Definition 21. *For a monoidal bicategory* $(\mathscr{V},\otimes,I)$*, a* left extension of the canonical action of $\mathscr{V}$ on itself *is a* $\mathscr{V}$*-action* $(\mathscr{B},\triangleright)$*, together with a 0-strict morphism* $(\mathsf{J},\theta):(\mathscr{V},\otimes)\to(\mathscr{B},\triangleright)$ *such that* θ *is invertible. (We say this is an extension* along J*.)*

We define a *right extension* analogously; this involves a right action $\triangleleft : \mathscr{B}\times\mathscr{V}\to\mathscr{B}$ and an invertible icon with components $\zeta_{f,g} : f\triangleleft \mathsf{J}g \Rightarrow \mathsf{J}(f\otimes g)$. The rest of this section is devoted to showing Freyd bicategories may be equivalently presented as pairs of extensions, which we call *Freyd actions*.

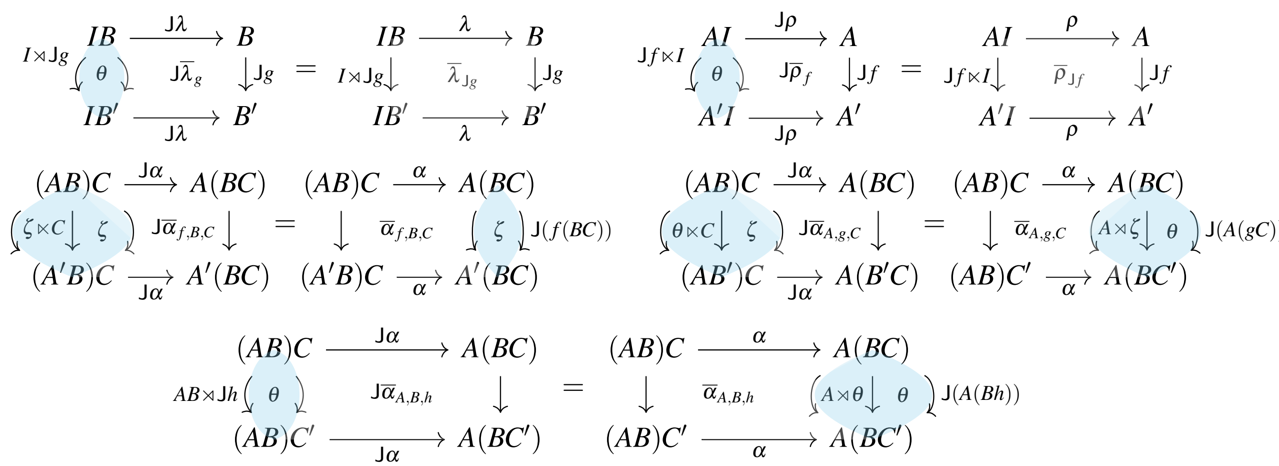
Definition 22. *A* Freyd action *consists of an identity-on-objects pseudofunctor* $\mathsf{J}:\mathscr{V}\to\mathscr{B}$ *from a monoidal bicategory* $(\mathscr{V},\otimes,I)$ *to a bicategory* $\mathscr{B}$*, together with:*

1. *A left extension* $(\triangleright,\theta)$ *and right extension* $(\triangleleft,\zeta)$ *along* J *of the canonical actions of* $\mathscr{V}$ *on itself;*
2. *A pseudonatural adjoint equivalence* κ *with 1-cell components* $\kappa_{X,B,Z}=\mathsf{J}(\alpha_{X,B,Z}):(X\triangleright B)\triangleleft Z\to X\triangleright(B\triangleleft Z)$*, subject to the equation below and additional axioms given in Appendix A:*

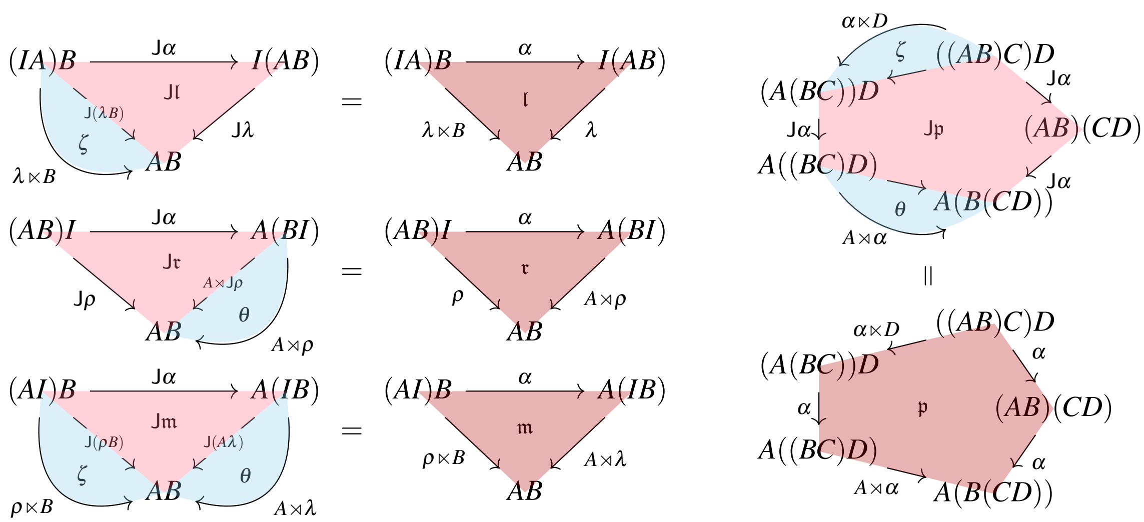


We construct an equivalence of categories between Freyd actions and Freyd bicategories, over a fixed identity-on-objects pseudofunctor $\mathsf{J}:\mathscr{V}\to\mathscr{B}$. (The corresponding categorical result is a bijection, but we must work modulo the structural isomorphisms, and hence lose the strictness.)

On one side, the category $\mathbf{FreydAct}(\mathsf{J})$ has objects Freyd actions $(\triangleright,\theta,\triangleleft,\zeta,\kappa)$ with underlying pseudofunctor J. Morphisms $((\triangleright,\theta),(\triangleleft,\zeta),\kappa)\to((\triangleright',\theta'),(\triangleleft',\zeta),\kappa')$ are pairs of icons $\vartheta:\triangleright\Rightarrow\triangleright'$ and



(a) Compatibility rules for structural transformations



(b) Compatibility rules for structural modifications

Figure 4: Compatibility laws for Definition 15

$\chi : \triangleleft \Rightarrow \triangleleft'$ fitting in the diagram in $\mathscr{V}$-$\mathbf{act}_{0s}$ as on the left below, such that κ is preserved as on the right:

$$\begin{tikzcd}
(\mathscr{B},\triangleright) \arrow[d, "{(\mathrm{id}_{\mathscr{B}},\vartheta)}"'] & (\mathscr{V},\otimes) \arrow[l, "{(J,\theta)}"'] \arrow[r, "{(J,\zeta')}"] \arrow[dl, "{(J,\theta')}"] \arrow[dr, "{(J,\zeta)}"'] & (\mathscr{B},\triangleleft) \arrow[d, "{(\mathrm{id}_{\mathscr{B}},\chi)}"] \\
(\mathscr{B},\triangleright') & & (\mathscr{B}',\triangleleft')
\end{tikzcd}
\qquad
\begin{tikzcd}
(AB)C \arrow[r, "\kappa'"] \arrow[d, phantom, "\vartheta\triangleleft h \; \Downarrow \; \chi"] & A(BC) \arrow[d, "f\triangleright'(b\triangleleft' h)"] \\
(A'B')C' \arrow[r, "\kappa'"'] \arrow[ur, phantom, "\overline{(\kappa')}_{f,b,h}"] & A'(B'C')
\end{tikzcd}
=
\begin{tikzcd}
(AB)C \arrow[r, "\kappa"] \arrow[d, "(f\triangleright b)\triangleleft h"'] & A(BC) \arrow[d, phantom, "f\triangleright\chi \; \Downarrow \; \vartheta"] \\
(A'B')C' \arrow[r, "\kappa"'] \arrow[ur, phantom, "\overline{\kappa}_{f,b,h}"] & A'(B'C')
\end{tikzcd}$$

On the other side, the category $\mathbf{FreydBicat}(\mathsf{J})$ has objects Freyd bicategories whose underlying pseudofunctor is J; these are determined by a premonoidal structure on $\mathscr{B}$ and families of icons $\theta = \{\theta^A \mid A \in \mathscr{B}\}$ and $\{\zeta^A \mid A \in \mathscr{B}\}$ making the pseudofunctor J premonoidal. Morphisms $(\rtimes, \ltimes, \theta, \zeta) \to (\rtimes', \ltimes', \theta', \zeta')$ are families of icons $\vartheta^A : (A \rtimes -) \Rightarrow (A \rtimes' -)$ and $\chi^A : (- \ltimes A) \Rightarrow (- \ltimes' A)$ making the identity pseudofunctor $\mathscr{B} \to \mathscr{B}$ premonoidal and such that $(J, \theta', \zeta') \circ (\mathrm{id}_{\mathscr{B}}, \vartheta, \chi) = (J, \theta, \zeta)$ as premonoidal pseudofunctors.

Our correspondence theorem is then as follows.

Theorem 23. *For any monoidal bicategory $(\mathscr{V}, \otimes, I)$, bicategory $\mathscr{B}$, and identity-on-objects pseudofunctor $\mathsf{J} : \mathscr{V} \to \mathscr{B}$, the categories* $\mathbf{FreydAct}(\mathsf{J})$ *and* $\mathbf{FreydBicat}(\mathsf{J})$ *are equivalent.*

5 Conclusions

Summary. We have introduced bicategorical versions of premonoidal categories (Definition 6) and Freyd categories (Definition 16). Along the way we have observed subtleties that arise only in the 2-dimensional setting, and discussed simple canonical examples. Finally, we have connected our theory to the existing literature by showing our definition is equivalent to certain actions in the expected way.

This paper develops abstract categorical notions, but these are intended to be immediately practical. Specifically, the literature contains no satisfying account of call-by-value languages in bicategories of games ([31, 3]), spans [6], or profunctors [7], and this work offers a technical basis to fill that gap. Our next steps will be in this direction.

Perspectives. This work takes place in a broader line of research on bicategorical semantic structures, and there are several avenues to explore. We expect a tight connection between Freyd bicategories and recently-developed notions of strength for pseudomonads on monoidal bicategories ([46, 35, 42]). Freyd bicategories should also be related to a 2-dimensional notion of *arrows*, based on **Cat**-valued profunctors, yet to be developed ([16, 4]).

In particular, the Kleisli bicategory of a strong pseudomonad should be premonoidal, and the canonical functor from the base category should give a Freyd structure, and conversely, a *closed* Freyd bicategory should be equivalent to a strong pseudomonad together with Kleisli exponentials. From a syntactic perspective, we expect cartesian Freyd bicategories to have an internal language similar to fine-grained call-by-value λ-calculus [26], with the addition of *rewrites* between terms (*c.f.* [41, 17, 18, 8]).

In a more theoretical direction, although the centre of a premonoidal category is always a monoidal category, this does not happen in the bicategorical setting. Roughly speaking, for central f and g, the interchange of f and g is witnessed independently by 2-cells lc_g^f and $(\mathsf{rc}_f^g)^{-1}$. This leads to ambiguity and it is not clear how to define the pseudofunctor $\otimes$; indeed, it is not even clear that these 2-cells are themselves central. In this paper we have shown that the centre is a binoidal bicategory, and in further work we will give a more complete description of its structure, along with an alternative presentation of Freyd bicategories in terms of centrality witnesses.

Acknowledgements. HP was supported by a Royal Society University Research Fellowship and by a Paris Region Fellowship co-funded by the European Union (Marie Skłodowska-Curie grant agreement 945298). PS was supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0038. Both authors thank D. McDermott, N. Arkor, and the Oxford PL group for useful discussions.

References

[1] J. C. Baez, B. Fong & B. S. Pollard (2016): *A compositional framework for Markov processes*. Journal of Mathematical Physics 57(3), doi:10.1063/1.4941578.

[2] J. Bénabou (1967): *Introduction to bicategories*. In: Reports of the Midwest Category Seminar, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–77, doi:10.1007/BFb0074299.

[3] S. Castellan, P. Clairambault, S. Rideau & G. Winskel (2017): *Games and Strategies as Event Structures*. Logical Methods in Computer Science Volume 13, Issue 3, doi:10.23638/LMCS-13(3:35)2017.

[4] Alexander S Corner (2016): *Day convolution for monoidal bicategories*. Ph.D. thesis, University of Sheffield.

[5] G. S. H. Cruttwell, B. Gavranović, N. Ghani, P. Wilson & F. Zanasi (2022): *Categorical Foundations of Gradient-Based Learning*. In: Programming Languages and Systems, doi:10.1007/978-3-030-99336-8_1.

[6] J. L. Fiadeiro & V. Schmitt (2007): *Structured Co-spans: An Algebra of Interaction Protocols*. In: Algebra and Coalgebra in Computer Science, pp. 194–208, doi:10.1007/978-3-540-73859-6_14.

[7] M. Fiore, N. Gambino, M. Hyland & G. Winskel (2007): *The cartesian closed bicategory of generalised species of structures*. Journal of the London Mathematical Society 77(1), pp. 203–220, doi:10.1112/jlms/jdm096.

[8] M. Fiore & P. Saville (2019): *A type theory for cartesian closed bicategories*. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), doi:10.1109/LICS.2019.8785708.

[9] B. Fong, D. Spivak & R. Tuyeras (2019): *Backprop as Functor: A compositional perspective on supervised learning*. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), doi:10.1109/lics.2019.8785665.

[10] M. Gaboardi, S. Katsumata, D. Orchard & T. Sato (2021): *Graded Hoare Logic and its Categorical Semantics*. In: Programming Languages and Systems, pp. 234–263, doi:10.1007/978-3-030-72019-3_9.

[11] Z. Galal (2020): *A Profunctorial Scott Semantics*. In: 5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020), doi:10.4230/LIPICS.FSCD.2020.16.

[12] F. R. Genovese, J. Herold, F. Loregian & D. 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.

[13] R. Gordon, A. J. Power & R. Street (1995): *Coherence for tricategories*. Memoirs of the American Mathematical Society, doi:10.1090/memo/0558.

[14] N. Gurski (2013): *Coherence in Three-Dimensional Category Theory*. Cambridge University Press, doi:10.1017/CBO9781139542333.

[15] N. Gurski & A. Osorno (2013): *Infinite loop spaces, and coherence for symmetric monoidal bicategories*. Advances in Mathematics 246, pp. 1 – 32, doi:10.1016/j.aim.2013.06.028.

[16] C. Heunen & B. Jacobs (2006): *Arrows, like Monads, are Monoids*. In: 22nd Annual Conference on Mathematical Foundations of Programming Semantics (MFPS), doi:10.1016/j.entcs.2006.04.012.

[17] B. P. Hilken (1996): *Towards a proof theory of rewriting: the simply typed 2λ-calculus*. Theoretical Computer Science 170(1), pp. 407–444, doi:10.1016/S0304-3975(96)80713-4.

[18] T. Hirschowitz (2013): *Cartesian closed 2-categories and permutation equivalence in higher-order rewriting*. Logical Methods in Computer Science 9, pp. 1–22, doi:10.2168/LMCS-9(3:10)2013.

[19] G. Janelidze & G. M. Kelly (2001): *A note on actions of a monoidal category*. Theory and Applications of Categories 9(4), pp. 61–91. Available at `tac.mta.ca/tac/volumes/9/n4/n4.pdf`.

[20] A. Jeffrey (1997): *Premonoidal categories and a graphical view of programs*.

[21] S. Katsumata (2014): *Parametric effect monads and semantics of effect systems*. In: 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), doi:10.1145/2535838.2535846.

[22] A. Kerinec, G. Manzonetto & F. Olimpieri (2023): *Why Are Proofs Relevant in Proof-Relevant Models?* Proceedings of the ACM on Programming Languages (POPL), doi:10.1145/3571201.

[23] S. Lack (2008): *Icons*. Applied Categorical Structures 18(3), pp. 289–307, doi:10.1007/s10485-008-9136-5.

[24] T. Leinster (2004): *Higher operads, higher categories*. London Mathematical Society Lecture Note Series 298, Cambridge University Press, doi:10.1017/CBO9780511525896.

[25] P. B. Levy (2003): *Call-By-Push-Value: A Functional/Imperative Synthesis*. Springer Netherlands, doi:10.1007/978-94-007-0954-6.

[26] P. B. Levy, J. Power & H. Thielecke (2003): *Modelling environments in call-by-value programming languages*. Information and Computation 185(2), pp. 182–210, doi:10.1016/s0890-5401(03)00088-9.

[27] S. Mac Lane & R. Paré (1985): *Coherence for bicategories and indexed categories*. Journal of Pure and Applied Algebra 37, pp. 59 – 80, doi:10.1016/0022-4049(85)90087-8.

[28] D. McDermott & T. Uustalu (2022): *Flexibly Graded Monads and Graded Algebras*. In: Lecture Notes in Computer Science, Springer International Publishing, pp. 102–128, doi:10.1007/978-3-031-16912-0_4.

[29] D. McDermott & T. Uustalu (2022): *What Makes a Strong Monad?* Electronic Proceedings in Theoretical Computer Science 360, pp. 113–133, doi:10.4204/eptcs.360.6.

[30] P.-A. Melliès (2012): *Parametric monads and enriched adjunctions*. Available at `https://www.irif.fr/~mellies/tensorial-logic/8-parametric-monads-and-enriched-adjunctions.pdf`.

[31] P.-A. Melliès (2021): *Asynchronous Template Games and the Gray Tensor Product of 2-Categories*. In: 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), doi:10.1109/lics52264.2021.9470758.

[32] E. Moggi (1989): *Computational lambda-calculus and monads*. In: Proceedings, Fourth Annual Symposium on Logic in Computer Science, IEEE Comput. Soc. Press, doi:10.1109/lics.1989.39155.

[33] E. Moggi (1991): *Notions of computation and monads*. Information and Computation 93(1), pp. 55–92, doi:10.1016/0890-5401(91)90052-4.

[34] F. Olimpieri (2021): *Intersection Type Distributors*. In: 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), doi:10.1109/lics52264.2021.9470617.

[35] H. Paquet & P. Saville (2023): *Strong pseudomonads and premonoidal bicategories*. arXiv:2304.11014.

[36] J. Power (2002): *Premonoidal categories as categories with algebraic structure*. Theoretical Computer Science 278(1-2), pp. 303–321, doi:10.1016/s0304-3975(00)00340-6.

[37] J. Power & E. Robinson (1997): *Premonoidal categories and notions of computation*. Mathematical Structures in Computer Science 7(5), pp. 453–468, doi:10.1017/s0960129597002375.

[38] J. Power & H. Thielecke (1997): *Environments, continuation semantics and indexed categories*. In: Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 391–414, doi:10.1007/bfb0014560.

[39] M. Román (2022): *Promonads and String Diagrams for Effectful Categories*. CoRR abs/2205.07664, doi:10.48550/arXiv.2205.07664. arXiv:2205.07664.

[40] C. J. Schommer-Pries (2009): *The Classification of Two-Dimensional Extended Topological Field Theories*. Ph.D. thesis, University of California. Available at `https://arxiv.org/pdf/1112.1000.pdf`.

[41] R. A. G. Seely (1987): *Modelling Computations: A 2-Categorical Framework*. In: 2nd Annual IEEE Symp. on Logic in Computer Science (LICS).

[42] A. Slattery (2023): *Pseudocommutativity and Lax Idempotency for Relative Pseudomonads*. arXiv:2304.14788.

[43] A. L. Smirnov (2008): *Graded monads and rings of polynomials*. Journal of Mathematical Sciences 151(3), pp. 3032–3051, doi:10.1007/s10958-008-9013-7.

[44] S. Staton (2017): *Commutative Semantics for Probabilistic Programming*. In: Programming Languages and Systems, Springer Berlin Heidelberg, pp. 855–879, doi:10.1007/978-3-662-54434-1_32.

[45] M. Stay (2016): *Compact Closed Bicategories*. Theories and Applications of Categories 31(26), pp. 755–798. Available at `http://www.tac.mta.ca/tac/volumes/31/26/31-26.pdf`.

[46] M. Tanaka (2005): *Pseudo-Distributive Laws and a Unified Framework for Variable Binding*. Ph.D. thesis, University of Edinburgh. Available at `https://www.lfcs.inf.ed.ac.uk/reports/04/ECS-LFCS-04-438/ECS-LFCS-04-438.pdf`.

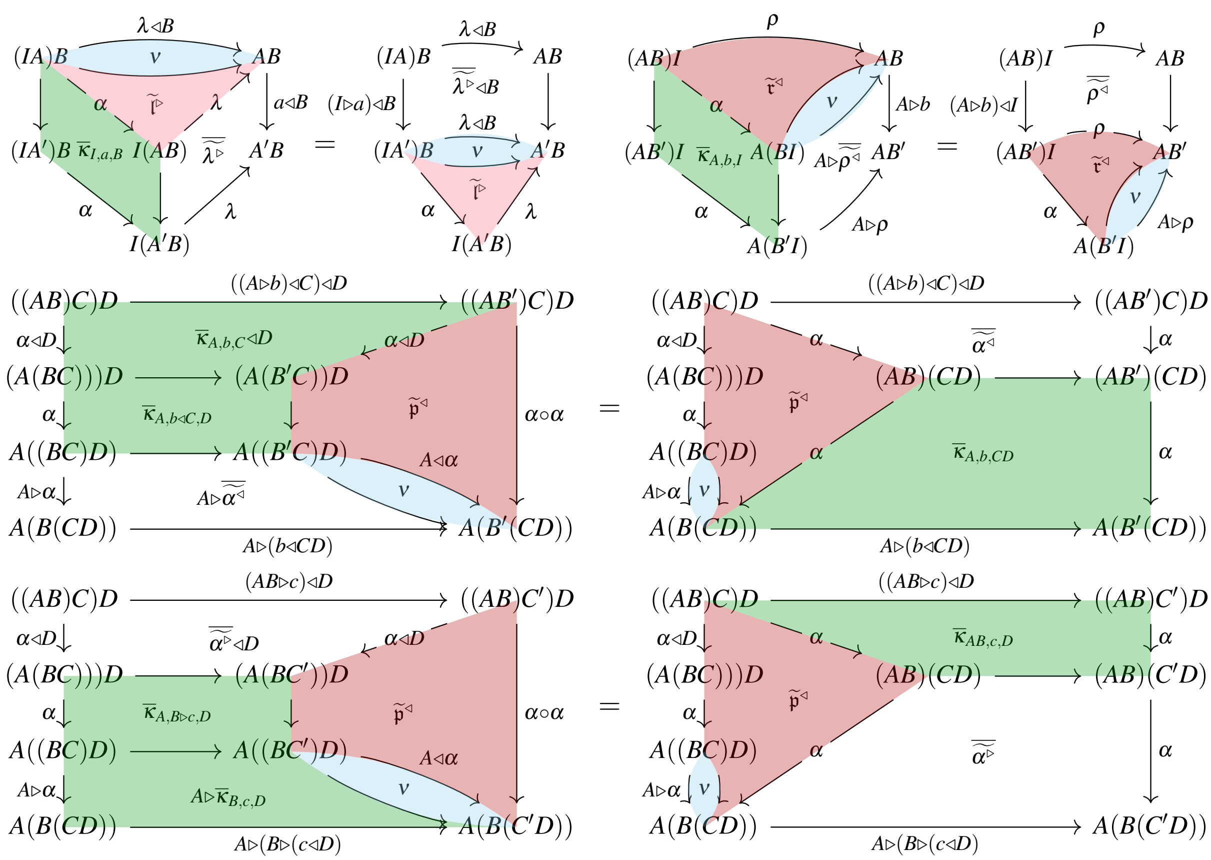
[47] H. Thielecke (1997): *Continuation Semantics and Self-adjointness*. Electronic Notes in Theoretical Computer Science 6, pp. 348–364, doi:10.1016/s1571-0661(05)80149-5.

[48] T. Tsukada, K. Asada & C.-H. L. Ong (2018): *Species, Profunctors and Taylor Expansion Weighted by SMCC*. In: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), doi:10.1145/3209108.3209157.

[49] L. Wester Hansen & M. Shulman (2019): *Constructing symmetric monoidal bicategories functorially*. arXiv:1910.09240.

A Missing coherence axioms for Definition 22

We complete the list of coherence equations for Definition 22. In the diagrams that follow, we consider $b : B \to B'$ and $c : C \to C'$ in $\mathscr{B}$ and, for $f : A \to A'$ and $g : B \to B'$ in $\mathscr{V}$, we write ν for the composite $\mathsf{J}f \triangleleft g \overset{\zeta}{\Rightarrow} \mathsf{J}(f \otimes g) \overset{\theta}{\Rightarrow} f \triangleright \mathsf{J}g$. In Theorem 23, κ represents the pseudonaturality of the associator α in its middle argument, and these axioms enforce the appropriate modification conditions (*c.f.* Figure 3).



B Proofs for Section 4

B.1 From Freyd action to Freyd bicategory

Fix a Freyd action $(\triangleright, \theta, \triangleleft, \zeta, \kappa)$ over $\mathsf{J} : \mathscr{V} \to \mathscr{B}$. We construct a Freyd bicategory with the same underlying pseudofunctor. For the unit of the premonoidal structure we take the unit I for $\mathscr{V}$. Next define $A \rtimes (-) := A \triangleright (-)$ and $(-) \ltimes B := (-) \triangleleft B$. The icons θ and ζ for the Freyd action then determine the required icons component-wise:

$$A \rtimes J(-) = A \triangleright (-) \xRightarrow{\theta_{A,-}} \mathsf{J}(A \otimes -) \qquad , \qquad \mathsf{J}(-) \ltimes B = J(-) \triangleleft B \xRightarrow{\zeta_{-,B}} \mathsf{J}(- \otimes B).$$

The left- and right unitors are given by $\widetilde{\lambda}^{\rhd} : IA \to A$ and $\widetilde{\lambda}^{\lhd} : AI \to A$ respectively, and the associator by $\mathsf{J}(\alpha)$ with 2-cell components given by the witnessing 2-cells for $\widetilde{\alpha}^{\rhd}$, κ, and $\widetilde{\alpha}^{\lhd}$. The compatibility laws of a Freyd action immediately give the compatibility laws of a Freyd bicategory. Similarly, the structural modifications are wholly determined by the definition of a Freyd bicategory: for example, the pentagonator $\mathfrak{p}$ in $\mathscr{B}$ is $\mathsf{J}(\mathfrak{p})$ composed with θ and ζ as in Definition 16. The axioms of a premonoidal bicategory are then checked using the compatibility laws and the corresponding axioms in $\mathscr{V}$.

It remains to show that the 2-cell components of θ and ζ are central and that J factors through the centre. The former is a short direct calculation. For the latter, for $f : X \to X'$ in $\mathscr{V}$ and $a : A \to A'$ in $\mathscr{B}$ we define lc^f_a using θ, ζ and the interchange laws for the pseudofunctors underlying the actions as in the diagram to the right; rc^f_a is similar. We write ν for the composite $\mathsf{J}f \lhd g \overset{\zeta}{\Rightarrow} \mathsf{J}(f \otimes g) \overset{\theta}{\Rightarrow} f \rhd \mathsf{J}g$.

$$
\mathsf{lc}^{\mathsf{J}f}_a := \begin{tikzcd}
XA \arrow[r, bend left, "\mathsf{J}(f)\lhd A"] \arrow[r, bend right, "f\rhd A"'] \arrow[r, phantom, "\nu"] \arrow[d, "X\rhd a"'] & X'A \arrow[d, "X'\rhd a"] \\
XA' \arrow[r, bend left, "f\rhd A'"] \arrow[r, bend right, "\mathsf{J}(f)\lhd A'"'] \arrow[r, phantom, "\nu"] \arrow[ur, phantom, "\cong"] & X'A'
\end{tikzcd}
$$

Thus, we define $\mathsf{J}'(f) := (f, \mathsf{lc}^{\mathsf{J}f}, \mathsf{rc}^{\mathsf{J}f})$. For any 2-cell $\sigma : f \Rightarrow f'$ in $\mathscr{V}$, we get that $\mathsf{J}(\sigma)$ is natural by naturality of all the data defining $\mathsf{lc}^{\mathsf{J}f}$ and $\mathsf{rc}^{\mathsf{J}f}$. Finally, one shows that the unit and compositor for J are central using the identity and composition laws of the icons θ and ζ.

In summary, we have the following:

Proposition 24. *Every Freyd action with underlying pseudofunctor* $\mathsf{J} : \mathscr{V} \to \mathscr{B}$ *determines a Freyd bicategory with the same underlying pseudofunctor.*

B.2 From Freyd bicategory to Freyd action

Let $\mathscr{F} = (\mathscr{V} \xrightarrow{\mathsf{J}} \mathscr{B})$ be a Freyd bicategory. First we shall show how to construct a left action $\rhd : \mathscr{V} \times \mathscr{B} \to B$; the right action is constructed similarly. Thereafter we shall show how to construct the rest of the data for a Freyd action.

From Freyd bicategory to a left action. We get a left action $\rhd : \mathscr{V} \times \mathscr{B} \to \mathscr{V}$ as follows. On objects, we set $X \rhd B := X \otimes B$. The action on 1-cells is $f \rhd b := \left(XB \xrightarrow{\mathsf{J}(f)\ltimes B} X'B \xrightarrow{X'\rtimes b} X'B'\right)$ with the evident action on 2-cells. The unitor is constructed from the unitors for the premonoidal structure, but the compositor relies on centrality. We define $\phi_{x,b}$ as on the right, where we write just $\cong$ for the compositors.

$$
\begin{tikzcd}
XB \arrow[d, "\mathsf{J}(f)\ltimes B"'] \arrow[dr, "\mathsf{J}(f'f)\ltimes B"] \arrow[dr, phantom, "\cong"'] & & \\
X'B \arrow[r, "\mathsf{J}(f')\ltimes B"] \arrow[d, "X'\ltimes b"'] \arrow[dr, phantom, "\mathsf{lc}^{\mathsf{J}f'}_b"] & X''B \arrow[d, "X''\rtimes b"] \arrow[dr, "X''\rtimes(b'b)"] \arrow[dr, phantom, "\cong"'] & \\
X'B' \arrow[r, "\mathsf{J}(f')\ltimes B'"'] & X''B' \arrow[r, "X''\rtimes b"'] & X''B''
\end{tikzcd}
$$

Next note that $X \rhd b = (X \rtimes b) \circ (\mathsf{JId}_X \ltimes B)$ so the unitor also gives a canonical structural isomorphism $(X \rhd b) \cong (X \rtimes b)$ yielding an icon $(X \rhd -) \Rightarrow (X \rtimes -)$. So we may define the unitor to be the composite $\widetilde{\lambda} := (I \rhd -) \overset{\cong}{\Rightarrow} (I \rtimes -) \overset{\lambda}{\Rightarrow} \mathrm{id}$. For the associator, we take the 1-cell components to be as for the premonoidal structure in $\mathscr{B}$, so that $\widetilde{\alpha}_{X,Y,C} := \alpha_{X,Y,C}$, and define the 2-cell components using θ, ζ, and the associator for the premonoidal structure:

$$
\overline{\overline{\alpha}}_{f,g,c} := \begin{tikzcd}
(XY)C \arrow[rr, "\mathsf{J}(f\otimes g)\ltimes C"] \arrow[dr, bend right, "(\mathsf{J}(f)Y)C"'] \arrow[dr, phantom, "\zeta C"] \arrow[d, "\alpha"'] & \arrow[d, phantom, "\cong"] & (X'Y')C \arrow[r, "(X'Y')\rtimes a"] \arrow[dd] \arrow[ddr, phantom, "\overline{\alpha}_{X',Y',c}"] & (X'Y')C' \arrow[dd, "\alpha"] \\
\arrow[r, phantom, "\overline{\alpha}_{\mathsf{J}f,Y,C}"] & (X'Y)C \arrow[ur, bend left, "(X\mathsf{J}(g))C"'] \arrow[ur, phantom, "\theta C"] \arrow[d] \arrow[r, phantom, "\overline{\alpha}_{X',\mathsf{J}g,C}"] & & \\
X(YC) \arrow[r, "\mathsf{J}(f)\ltimes(YC)"'] & X'(YC) \arrow[r, "X'\rtimes(\mathsf{J}(g)\ltimes C)"'] & X'(Y'C) \arrow[r, "X'\ltimes(Y'\ltimes a)"'] & X'(Y'C')
\end{tikzcd}
$$

The compatibility laws on $\overline{\widetilde{\lambda}}$ and $\overline{\widetilde{\alpha}}$ hold by the corresponding compatibility laws of a Freyd bicategory. Turning now to the structural modifications, because the structural transformations agree with those of $\mathscr{B}$ on 1-cells, we take the corresponding modifications for the premonoidal structure. Showing these are indeed modifications relies on the condition that $\mathsf{lc}^{\mathsf{J}f}_{\mathsf{J}g} = (\mathsf{rc}^{\mathsf{J}g}_{\mathsf{J}f})^{-1}$. Consider the case of $\widetilde{\mathfrak{m}}$. As 2-cells, $\widetilde{\mathfrak{m}}_{A,B} = \mathfrak{m}_{A,B}$ but $\widetilde{\mathfrak{m}}$ is required to be a modification in two arguments, while the axioms of a premonoidal bicategory make $\mathfrak{m}_{A,B}$ a modification in each argument separately: in one argument, using rc^{λ}, and in the other argument using lc^{ρ}. Unpacking the equations for showing $\widetilde{\mathfrak{m}}$ is a modification at maps $a : A \to A'$ and $x : X \to X'$, we get an instance of $\mathsf{lc}^{\mathsf{J}a}_{\lambda}$ arising from the compositor for $\rhd$. To apply the modification law for $\mathfrak{m}$, therefore, we first need to pass through the equality $\mathsf{lc}^{\mathsf{J}a}_{\lambda} = \mathsf{lc}^{\mathsf{J}a}_{\mathsf{J}\lambda} = (\mathsf{rc}^{\mathsf{J}\lambda}_{\mathsf{J}a})^{-1}$.

The axioms of an action hold immediately from the axioms of a premonoidal bicategory. The proof for the right action case is analogous, except one sets $a \lhd g := \left(AY \xrightarrow{a \ltimes Y} A'Y \xrightarrow{A' \rtimes \mathsf{J}g} A'Y'\right)$ and defines the compositor using right centrality. In summary, therefore, we have the following.

Proposition 25. *Every Freyd bicategory ($\mathscr{V} \xrightarrow{\mathsf{J}} \mathscr{B}$) determines a left action $\rhd : \mathscr{V} \times \mathscr{B} \to \mathscr{B}$ and a right action $\lhd : \mathscr{B} \times \mathscr{V} \to \mathscr{V}$.*

From Freyd bicategory to Freyd action. It remains to show the actions just constructed extend the canonical action of $\mathscr{V}$ on itself, and show they are compatible. First we define icons θ' and ζ' by noting that $f \rhd \mathsf{J}(g) = (X' \rtimes \mathsf{J}(g)) \circ (\mathsf{J}(f) \ltimes Y) = \mathsf{J}(f) \lhd g$ so that we can set $\theta'_{f,g}$ and $\zeta'_{f,g}$ both to be the composite diagram on the right. In particular, $\theta'_{f,Y}$ and $\zeta'_{X,g}$ are just θ_f and ζ_g, respectively, composed with structural isomorphisms.

$$
\begin{tikzcd}
& X'Y \arrow[dr, bend left, "X' \rtimes \mathsf{J}(g)"] \arrow[dr, ""{name=B}] \arrow[d, phantom, "\cong"] & \\
XY \arrow[ur, bend left, "\mathsf{J}(f) \ltimes Y"] \arrow[ur, ""{name=A}] \arrow[rr, "\mathsf{J}(f \otimes g)"'] & {} & X'Y'
\arrow[from=1-2, to=2-3, phantom, bend left=15, "\theta"]
\arrow[from=2-1, to=1-2, phantom, bend left=15, "\zeta"]
\end{tikzcd}
$$

Now we define κ. On 1-cells we take just α, but on 2-cells we take a definition similar to the proof of naturality in the 1-dimensional case: for $f : X \to X'$ and $h : Z \to Z'$ in $\mathscr{V}$ and $b : B \to B'$ in $\mathscr{B}$ we take:

$$
\overline{\kappa}_{f,b,h} :=
\begin{tikzcd}
(XB)Z \arrow[rr, "(f \rhd b) \ltimes Z"] \arrow[rrr, bend left=30, "(f \rhd b) \lhd h"] \arrow[dr, "(\mathsf{J}(f)B)Z"] \arrow[dd, "\alpha"'] & {} \arrow[d, phantom, "\cong"] & (X'B')Z \arrow[r, "(X'B') \rtimes \mathsf{J}h"] \arrow[dd] \arrow[ddr, phantom, "\overline{\alpha}_{X',B',\mathsf{J}h}"] & (X'B')Z' \arrow[dd, "\alpha"] \\
{} \arrow[r, phantom, "\overline{\alpha}_{\mathsf{J}f,B,Z}"] & (X'B)A \arrow[ur, "(X \rtimes b)Z"] \arrow[d] \arrow[r, phantom, "\overline{\alpha}_{X',b,Z}"] & {} & \\
X(BZ) \arrow[r, "\mathsf{J}(f) \ltimes (BZ)"'] \arrow[rrr, bend right=40, "f \rhd (b \lhd h)"'] & X'(BZ) \arrow[r, "X' \rtimes (b \ltimes Z)"'] \arrow[rr, bend right=30, "X' \rtimes (b \lhd h)"'] & X'(B'Z) \arrow[r, "X' \ltimes (B' \ltimes \mathsf{J}h)"'] \arrow[d, phantom, "\cong"] & X'(B'Z') \\
& & {} &
\end{tikzcd}
$$

The rest of the equations to check for the Freyd action are proven by applying the various compatibility laws to massage the statement into the corresponding axiom given by the definition of a Freyd bicategory. This completes the proof of the following.

Proposition 26. *Every Freyd bicategory ($\mathscr{V} \xrightarrow{\mathsf{J}} \mathscr{B}$) determines a Freyd action with the same underlying pseudofunctor.*

B.3 The correspondence theorem

Theorem 23. *For any monoidal bicategory $(\mathscr{V}, \otimes, I)$, bicategory $\mathscr{B}$, and identity-on-objects pseudofunctor $\mathsf{J} : \mathscr{V} \to \mathscr{B}$, the categories* $\mathbf{FreydAct}(\mathsf{J})$ *and* $\mathbf{FreydBicat}(\mathsf{J})$ *are equivalent.*

Proof. We define functors $F : \mathbf{FreydAct} \leftrightarrows \mathbf{FreydBicat} : G$ given on objects by the constructions in Proposition 24 and Proposition 26 respectively. So suppose (ϑ, χ) is a map in **FreydAct**. Then $F(\vartheta, \chi) := (F\vartheta, F\chi)$ is defined by taking

$$(F\vartheta)^A_f := \big((A \rtimes f) = (\mathsf{Id}_A \triangleright f) \xRightarrow{\vartheta_{\mathsf{Id}_A, f}} (\mathsf{Id}_A \triangleright' f) = (A \rtimes' f)\big)$$

$$(F\chi)^A_f := \big((f \ltimes A) = (f \triangleleft \mathsf{Id}_A) \xRightarrow{\chi_{f, \mathsf{Id}_A}} (f \triangleleft' \mathsf{Id}_A) = (f \ltimes A)\big)$$

Conversely, given a map (ϑ, χ) in **FreydBicat** we define $G(\vartheta, \chi) := (G\vartheta, G\chi)$ to be

$$(G\vartheta)_{f,b} := \begin{tikzcd} XB \arrow[r, bend left, "\mathsf{J}f \ltimes B"] \arrow[r, bend right, "\mathsf{J}f \ltimes' B"'] \arrow[r, phantom, "\Downarrow \chi^B_{\mathsf{J}f}"] \arrow[rr, bend left=60, "f \triangleright b"] \arrow[rr, bend right=60, "f \triangleright' b"'] & X'B \arrow[u, phantom, "\cong"] \arrow[d, phantom, "\cong"] \arrow[r, bend left, "X' \rtimes b"] \arrow[r, bend right, "X' \rtimes' b"'] \arrow[r, phantom, "\Downarrow \vartheta^{X'}_b"] & X'B' \end{tikzcd} \quad , \quad (G\chi)_{a,g} := \begin{tikzcd} AX \arrow[r, bend left, "a \ltimes X"] \arrow[r, bend right, "a \ltimes' X"'] \arrow[r, phantom, "\Downarrow \chi^X_a"] \arrow[rr, bend left=60, "a \triangleleft g"] \arrow[rr, bend right=60, "a \triangleleft' g"'] & A'X \arrow[u, phantom, "\cong"] \arrow[d, phantom, "\cong"] \arrow[r, bend left, "A' \rtimes \mathsf{J}g"] \arrow[r, bend right, "A' \rtimes' \mathsf{J}g"'] \arrow[r, phantom, "\Downarrow \vartheta^{A'}_{\mathsf{J}g}"] & A'X' \end{tikzcd}$$

One shows both F and G are well-defined by a long calculation using the compatibility properties on one side to show the required compatibility condition on the other side.

We now show that $GF \cong \mathrm{id}$ and $FG \cong \mathrm{id}$. Given an action $\mathscr{A} := (\triangleright, \theta, \triangleleft, \zeta, \kappa)$, the composite $GF(\mathscr{A})$ has left action $\triangleright'$ given by $f \triangleright' b = (\mathsf{Id}_{X'} \triangleright b) \circ (\mathsf{J}f \triangleleft \mathsf{Id}_B)$ and right action $\triangleleft'$ given by $a \triangleleft' g = (\mathsf{Id}_{A'} \triangleright \mathsf{J}g) \circ (a \triangleleft \mathsf{Id}_Y)$ so we get an obvious choice of icons $\triangleright' \Rightarrow \triangleright$ and $\triangleleft' \Rightarrow \triangleleft$ given by

$$\begin{tikzcd} XB \arrow[r, bend left=50, "\mathsf{J}f \triangleleft \mathsf{Id}_B"] \arrow[r] \arrow[r, bend right=50, "f \triangleright \mathsf{Id}_B"'] \arrow[r, phantom, shift left=2, "\zeta"] \arrow[r, phantom, shift right=2, "\theta"] \arrow[rr, bend right=70, "f \triangleright b"'] & X'B \arrow[d, phantom, "\cong"] \arrow[r, "\mathsf{Id}_{X'} \triangleright b"] & X'B' \end{tikzcd} \qquad \begin{tikzcd} AY \arrow[r, "a \triangleleft \mathsf{Id}_Y"'] \arrow[rr, bend right=70, "a \triangleleft g"'] & A'Y \arrow[d, phantom, "\cong"] \arrow[r, bend left=50, "\mathsf{Id}_{A'} \triangleright \mathsf{J}g"] \arrow[r] \arrow[r, bend right=50, "\mathsf{Id}_{A'} \triangleleft g"'] \arrow[r, phantom, shift left=2, "\theta"] \arrow[r, phantom, shift right=2, "\zeta"] & A'Y' \end{tikzcd}$$

These commute with all the data because θ and ζ do, and forms a natural isomorphism $GF(\mathscr{A}) \cong \mathscr{A}$ because morphisms in **FreydAct** commute with the icons of the actions.

Finally, to show that $FG \cong \mathrm{id}$ consider a Freyd bicategory $\mathscr{F} := (\rtimes, \theta, \ltimes, \zeta)$. Then $FG(\mathscr{F})$ has $a \ltimes' B := (X' \rtimes \mathsf{J}\mathsf{Id}_B) \circ (a \ltimes B)$ and $A \rtimes' b := (A \rtimes b) \circ (\mathsf{J}\mathsf{Id}_A \ltimes B)$ so we have evident structural isomorphisms $(a \ltimes' B) \cong (a \ltimes B)$ and $(A \rtimes' b) \cong (A \rtimes b)$. These commute with all the data and define a natural isomorphism $FG(\mathscr{F}) \cong \mathscr{F}$ by straightforward applications of coherence. $\square$

Structured and Decorated Cospans from the Viewpoint of Double Category Theory

Evan Patterson
Topos Institute
evan@epatters.org

Structured and decorated cospans are broadly applicable frameworks for building bicategories or double categories of open systems. We streamline and generalize these frameworks using central concepts of double category theory. We show that, under mild hypotheses, double categories of structured cospans are cocartesian (have finite double-categorical coproducts) and are equipments. The proofs are simple as they utilize appropriate double-categorical universal properties. Maps between double categories of structured cospans are studied from the same perspective. We then give a new construction of the double category of decorated cospans using the recently introduced double Grothendieck construction. Besides its conceptual value, this reconstruction leads to a natural generalization of decorated cospans, which we illustrate through an example motivated by statistical theories and other theories of processes.

1 Introduction

A central theme of applied category theory is the mathematical modeling of open systems: physical or computational systems that interact with each other along boundaries or interfaces. Within this tradition, mathematical models of open systems are most commonly based on spans or cospans, an idea now at least twenty-five years old [17, 22]. Two general frameworks for building open systems using cospans have emerged: structured cospans [5, 11] and decorated cospans [6, 12]. Complementing the mathematical theory, structured cospans have been implemented in the programming framework Catlab.jl and used to create software tools for epidemiological modeling based on open Petri nets [7, 18] and open stock and flow diagrams [3]. Structured and decorated cospans are now essential tools of applied and computational category theory.

The categorical description of open systems based on cospans has evolved over time. Some early works studied categories of cospans, which compose by taking pushouts. Because pushouts are defined only up to isomorphism, the morphisms of these categories must be *isomorphism classes* of cospans. This is unfaithful to implementation, where one always computes with representatives of an equivalence class, rather than the equivalence class itself. More fundamentally, systems generally have morphisms of their own—for example, Petri nets come with homomorphisms between them—and these are lost if open systems are taken to be morphisms, rather than objects, of a category.

Both problems are solved by passing from categories to a two-dimensional categorical structure, of which the best-studied are bicategories. Yet this presents its own difficulties. In addition to composing along their boundaries, open systems generally admit a symmetric monoidal product that juxtaposes two of them "in parallel." One then needs to construct not just a bicategory but a symmetric monoidal bicategory of open systems. Monoidal bicategories are inherently complicated because they are properly a three-dimensional categorical structure (namely, tricategories with one object). It was noticed that rather than constructing a monoidal bicategory directly, it can be easier to first construct a monoidal *double*

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 210–225, doi:10.4204/EPTCS.397.13

category and then obtain the monoidal bicategory from the globular cells of the double category [16, 24]. But since a double category is at least as good as a bicategory, one may as well consider double categories of open systems. That is now what is typically done. In recent work, both structured cospans [5] and decorated cospans [6] have been assembled into symmetric monoidal double categories.

The thesis of this paper is that viewing open systems as double categories is not merely a technical device or a means to constructing bicategories, but a source of mathematical insights that cannot be obtained at the 1-categorical or even bicategorical levels. To understand why, consider the philosophy behind the modern theory of double categories, as developed principally by Grandis and Paré, beginning with an account of double limits and colimits [15], and exposited recently by Grandis [14]. Another important expression of this viewpoint is Shulman's theory of equipments [23].

A **double category** is succinctly defined as a pseudocategory in **Cat**.[1] Thus, a double category $\mathbb{D}$ consists of a category of objects, $\mathbb{D}_0$; a category of morphisms, $\mathbb{D}_1$; source and target functors, $\mathrm{src}, \mathrm{tgt} : \mathbb{D}_1 \rightrightarrows \mathbb{D}_0$; and external composition and identity operations, $\odot : \mathbb{D}_1 \times_{\mathbb{D}_0} \mathbb{D}_1 \to \mathbb{D}_1$ and $\mathrm{id} : \mathbb{D}_0 \to \mathbb{D}_1$, which obey the category axioms up to coherent globular isomorphisms in $\mathbb{D}_1$. The objects of the category $\mathbb{D}_0$ are called the **objects** of the double category $\mathbb{D}$, the morphisms of $\mathbb{D}_0$ the **arrows** of $\mathbb{D}$, the objects of $\mathbb{D}_1$ the **proarrows** of $\mathbb{D}$, and the morphisms of $\mathbb{D}_1$ the **cells** of $\mathbb{D}$. In particular, on this definition, the proarrows of a double category are first and foremost the *objects* of a category, which happen to have a source and target. In important examples of double categories, such as those of spans, cospans, relations, matrices, profunctors, and bimodules, the proarrows are best thought of in precisely this way, as objects that happen to have a source and target. Crucially, this also applies to double categories of open systems, which are systems that happen to have boundaries. Shulman calls such double categories (or rather their underlying bicategories) "$\mathscr{M}od$-like" after the bicategory of bimodules between rings [23].

Whether one thinks of proarrows primarily as objects or morphisms may seem a small matter of perspective, but it gains significance through the modern theory of double categories, where proarrows play the role of objects in all of the main concepts, such as natural transformations, limits and colimits, commas, adjunctions, and the Grothendieck construction. The theory is thus well suited to describe open systems, including those based on spans and cospans. In this paper, we study structured and decorated cospans from the viewpoint of double category theory.[2]

After reviewing their structure as a double category, we show that structured cospans form a cocartesian double category, a statement that is stronger yet easier to prove than being a symmetric monoidal double category. We also show that structured cospans are an equipment, so altogether form a cocartesian equipment (Section 2). Here we see the advantages of double-categorical universal properties. We then turn to decorated cospans (Section 3), reconstructing the double category of decorated cospans as an application of the double Grothendieck construction [10]. As a byproduct, we also generalize decorated cospans in several directions, which we illustrate through an example motivated by categorical statistics.

2 Structured Cospans as a Cocartesian Equipment

Structured cospans represent open systems as cospans whose feet are restricted compared with the apex [5, 11]. A simple example is open graphs with boundaries restricted to be *discrete* graphs. Compared with other techniques, structured cospans have the advantage of being particularly easy to use, as the

[1] Some authors call this structure a **pseudo double category** but since all double categories in this paper are pseudo, we prefer to omit the adjective. Likewise, our double functors are pseudo by default. For complete definitions of these concepts, see [14].

[2] This paper synthesizes a series of blog posts by the author: "Grothendieck construction for double categories" (2022), "Decorated cospans via the Grothendieck construction" (2022), "Structured cospans as a cocartesian equipment" (2023).

hypotheses for the construction are often easy to check in examples. However, proofs that the construction itself works are more involved because the mathematical object being constructed—a symmetric monoidal double category—is complicated, involving a large number of coherence conditions. Three different correctness proofs for structured cospans have been given: the first one by direct but lengthy verification of the axioms [9] and two later ones by more conceptual routes that however import other sophisticated concepts. These concepts are pseudocategories in the 2-category of symmetric monoidal categories [5] and symmetric monoidal bifibrations [6].

Such difficulties can be bypassed by viewing structured cospans in a different light, as forming a *cocartesian* double category, even a cocartesian equipment. Just as a cartesian or cocartesian category can be given the structure of a symmetric monoidal category by making a choice of finite products or coproducts, so can a cartesian or cocartesian double category be given the structure of a symmetric monoidal double category. It is, however, much easier to prove cocartesianness than to directly construct the symmetric monoidal product. This circumstance highlights a recurring tension in category theory: that between universal properties and algebraic structures. Although algebraic structure is arguably more flexible, universal properties, when they can be found, are extremely powerful because many consequences and coherences flow directly from the defining existence and uniqueness statement, which is often easy to verify in particular situations. Both cocartesian double categories and equipments are defined by universal properties, whereas a symmetric monoidal product is a structure on a double category.

2.1 Double Category of Structured Cospans

We begin by reviewing the definition of structured cospans and their structure as a double category [5].

Proposition 2.1. *Let $L : \mathsf{A} \to \mathsf{X}$ be a functor into a category X with pushouts. Then there is a double category ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})$ that has*

- *as objects, the objects of A;*
- *as arrows, the morphisms of A;*
- *as proarrows $a \nrightarrow b$, **L-structured cospans** with feet a and b, which are cospans in X of the form $La \to x \leftarrow Lb$;*
- *as cells* $\begin{array}{ccc} a & \overset{x}{\nrightarrow} & b \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ c & \underset{y}{\nrightarrow} & d \end{array}$, ***morphisms of L-structured cospans** with foot maps f and g, which are morphisms of cospans in X of the form*

$$\begin{tikzcd} L(a) \arrow[r] \arrow[d, "Lf"'] & x \arrow[d, "h"] & L(b) \arrow[l] \arrow[d, "Lg"] \\ L(d) \arrow[r] & y & L(c) \arrow[l] \end{tikzcd} .$$

Composition in the categories ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})_0$ and ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})_1$ is by composition in A and $\mathbb{C}\mathsf{sp}(\mathsf{X})_1$, respectively, and external composition in ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})$ is given by that in $\mathbb{C}\mathsf{sp}(\mathsf{X})$, i.e., by pushout in X.

We take this result as given. The proof is straightforward because the double category structure of L-structured cospans is inherited from that of cospans in X. For details, see [5, Theorem 2.3].

2.2 Cocartesian Equipment of Structured Cospans

We now prove that structured cospans form an equipment, then a cocartesian double category, and hence a cocartesian equipment. The reader may find it helpful to review the definitions of cocartesian double categories and equipments in Appendix A.

Proposition 2.2. *Let $L : \mathsf{A} \to \mathsf{X}$ be a functor into a category X with pushouts. Then the double category of L-structured cospans is an equipment.*

Proof. To restrict an L-structured cospan $(c, Lc \to y \leftarrow Ld, d)$ along arrows $f : a \to c$ and $g : b \to d$ in A, simply restrict the underlying cospan in X along Lf and Lg, using the fact that $\mathbb{C}\mathsf{sp}(\mathsf{X})$ is an equipment (Example A.3). The universal property holds as a special case of the universal property in $\mathbb{C}\mathsf{sp}(\mathsf{X})$:

$$
\begin{tikzcd}
La' \arrow[r] \arrow[d, "Lf'"'] & x \arrow[dd, "h"] & Lb' \arrow[l] \arrow[d, "Lg'"] \\
La \arrow[d, "Lf"'] & & Lb \arrow[d, "Lg"] \\
Lc \arrow[r, "\ell"'] & y & Ld \arrow[l, "r"]
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
La' \arrow[r] \arrow[d, "Lf'"'] & x \arrow[d, dashed, "h"'] & Lb' \arrow[l] \arrow[d, "Lg'"] \\
La \arrow[r, "\ell\circ Lf"'] \arrow[d, "Lf"'] & y \arrow[d, equal] & Lb \arrow[l, "r\circ Lg"] \arrow[d, "Lg"] \\
Lc \arrow[r, "\ell"'] & y & Ld \arrow[l, "r"]
\end{tikzcd}
$$
□

For the double category of L-structured cospans to be cocartesian, extra assumptions are needed. Clearly, the category A must itself have finite coproducts. Also, these must preserved by the functor $L : \mathsf{A} \to \mathsf{X}$. The latter is often is easy to verify in examples by exhibiting L as a left adjoint.

Theorem 2.3. *Suppose A is a category with finite coproducts, X is a category with finite colimits, and $L : \mathsf{A} \to \mathsf{X}$ is a functor that preserves finite coproducts. Then the double category of L-structured cospans is cocartesian, hence also a cocartesian equipment.*

Proof. Because the categories A and X have finite coproducts, there are canonical comparison maps

$$L_{a,a'} := [L(\iota_a), L(\iota_{a'})] : L(a) + L(a') \to L(a + a'), \qquad a, a' \in \mathsf{A},$$

and $L_0 := !_{L(0)} : 0_{\mathsf{X}} \to L(0_{\mathsf{A}})$. For any maps $f : a \to c$ and $f' : a' \to c$ in A, the comparisons satisfy

$$
\begin{tikzcd}
L(a)+L(a') \arrow[rr, "L_{a,a'}"] \arrow[dr, "{[Lf,Lf']}"'] & & L(a+a') \arrow[dl, "{L([f,f'])}"] \\
& L(c) &
\end{tikzcd}
$$

as shown by precomposing both sides with the coprojections ι_{La} and $\iota_{La'}$ to obtain Lf and Lf', respectively. Since by assumption L preserves finite coproducts, the comparisons $L_{a,a'}$ and L_0 are, in fact, isomorphisms.

We now prove that the categories underlying ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})$ are cocartesian. By assumption, the category ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})_0 = \mathsf{A}$ has finite coproducts. Since the comparison L_0 is an isomorphism, $L(0_{\mathsf{A}})$ is initial in X and the initial L-structured cospan is $(0_{\mathsf{A}}, \mathrm{id}_{L(0_{\mathsf{A}})}, 0_{\mathsf{A}})$. Furthermore, the coproduct of two L-structured cospans $(a, La \to x \leftarrow Lb, b)$ and $(a', La' \to x' \leftarrow Lb', b')$, denoted

$$(a + a',\ L(a+a') \to x + x' \leftarrow L(b+b'),\ b + b'),$$

is obtained from the pointwise coproduct of cospans in X by restriction along the inverse comparisons $L_{a,a'}^{-1}$ and $L_{b,b'}^{-1}$. The universal property of coproducts in ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})_1$ then takes the form:

$$
\begin{tikzcd}
L(a+a') \arrow[r, "L_{a,a'}^{-1}"] \arrow[d, "{L([f,f'])}"'] & L(a)+L(a') \arrow[r] \arrow[d, "{[Lf,Lf']}"'] & x+x' \arrow[d, "{[h,h']}"'] & L(b)+L(b') \arrow[l] \arrow[d, "{[Lg,Lg']}"] & L(b+b') \arrow[l, "L_{b,b'}^{-1}"'] \arrow[d, "{L([g,g'])}"] \\
L(c) \arrow[r, equal] & L(c) \arrow[r] & y & L(d) \arrow[l] & L(d) \arrow[l, equal]
\end{tikzcd}
.
$$

We have shown that both categories underlying ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})$ have finite coproducts, and it is immediate that the source and target functors preserve them.

Finally, the comparison cells in ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})$ interchanging finite coproducts with external composition and identity (Definition A.1) are all isomorphisms because they are defined by the same maps in X as the comparison cells in $\mathbb{C}\mathsf{sp}(\mathsf{X})$, which we already know to be isomorphisms (Example A.3). □

As a corollary, every double category of L-structured cospans satisfying the hypotheses of the theorem can be given the structure of a symmetric monoidal double category, by making choices of coproducts in both underlying categories. This follows abstractly because any cocartesian object in a 2-category with finite 2-products is a symmetric pseudomonoid in a canonical way [24, Remark 2.11]. Cocartesian double categories are cocartesian objects in the 2-category **Dbl**, whereas symmetric monoidal double categories are symmetric pseudomonoids in **Dbl**.

2.3 Maps Between Structured Cospan Double Categories

We complete the essential theory of structured cospans by showing how to construct maps between cocartesian equipments of structured cospans. These maps are cocartesian double functors (Definition A.4). Compared with the original results [5, Theorems 4.2 and 4.3], the theorem below is slightly more general, treating the lax case as well as the pseudo one, and slightly stronger, yielding cocartesian double functors instead of symmetric monoidal ones.

Theorem 2.4. *Suppose we have a diagram in* **Cat** *of the form*

$$\begin{tikzcd} \mathsf{A} \arrow[r, "L"] \arrow[d, "F_0"'] & \mathsf{X} \arrow[d, "F_1"] \\ \mathsf{A}' \arrow[r, "L'"'] \arrow[ur, phantom, "\overset{\alpha}{\nearrow}"] & \mathsf{X}' \end{tikzcd},$$

where the categories X and X' have pushouts. Then there is a lax double functor $\mathbb{F} : {}_L\mathbb{C}\mathsf{sp}(\mathsf{X}) \to {}_{L'}\mathbb{C}\mathsf{sp}(\mathsf{X}')$ that has underlying functor $\mathbb{F}_0 = F_0$ and acts on proarrows as

$$(a, L(a) \xrightarrow{\ell} x \xleftarrow{r} L(b), b) \quad \mapsto \quad (F_0(a), L'(F_0(a)) \xrightarrow{\alpha_a} F_1(L(a)) \xrightarrow{F_1(\ell)} F_1(x) \xleftarrow{F_1(r)} F_1(L(b)) \xleftarrow{\alpha_b} L'(F_0(b)), F_0(b))$$

and on cells as

$$\begin{tikzcd} L(a) \arrow[r, "\ell"] \arrow[d, "Lf"'] & x \arrow[d, "h"] & L(b) \arrow[l, "r"'] \arrow[d, "Lg"] \\ L(a') \arrow[r, "\ell'"'] & x' & L(b') \arrow[l, "r'"] \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} L'(F_0(a)) \arrow[r, "F_1(\ell)\circ\alpha_a"] \arrow[d, "L'(F_0(f))"'] & F_1(x) \arrow[d, "F_1(h)"] & L'(F_0(b)) \arrow[l, "F_1(r)\circ\alpha_b"'] \arrow[d, "L'(F_0(g))"] \\ L'(F_0(a')) \arrow[r, "F_1(\ell')\circ\alpha_{a'}"'] & F_1(x') & L'(F_0(b')) \arrow[l, "F_1(r')\circ\alpha_{b'}"] \end{tikzcd}.$$

Moreover, $\mathbb{F}$ is a pseudo double functor whenever F_1 preserves pushouts and α is a natural isomorphism.

Suppose further that all of the categories in question have finite coproducts and that L and L' preserve them, so that both double categories ${}_L\mathbb{C}\mathsf{sp}(\mathsf{X})$ and ${}_{L'}\mathbb{C}\mathsf{sp}(\mathsf{X}')$ are cocartesian. Then the lax double functor $\mathbb{F}$ is cocartesian if and only if both functors F_0 and F_1 are cocartesian. In particular, $\mathbb{F}$ is a cocartesian pseudo double functor whenever F_0 preserves finite coproducts, F_1 preserves finite colimits, and α is a natural isomorphism.

As a substantial application of the theorem, we have formulated the generalized Lokta-Volterra model as a cocartesian lax double functor from open signed graphs to open parameterized dynamical systems [1].

We prove the theorem by decomposing the lax double functor $\mathbb{F}$ into three simpler ones. Taken together, the lemmas also implicitly give formulas for the laxators and unitors of $\mathbb{F}$, which we omitted in the theorem statement.

Lemma 2.5. *Let* X *be a category with pushouts and let* $\mathsf{A}_0 \xrightarrow{F_0} \mathsf{A} \xrightarrow{L} \mathsf{X}$ *be functors. Then there is a* strict *double functor* ${}_{F_0}\mathbb{C}\mathsf{sp}(\mathsf{X}) : {}_{L\circ F_0}\mathbb{C}\mathsf{sp}(\mathsf{X}) \to {}_{L}\mathbb{C}\mathsf{sp}(\mathsf{X})$ *given by* F_0 *on objects and arrows and by the identity on the cospans and maps of cospans underlying proarrows and cells.*

Furthermore, the double functor ${}_{F_0}\mathbb{C}\mathsf{sp}(\mathsf{X})$ *is cocartesian whenever* A_0, A, *and* X *have finite coproducts and the functors* L *and* F_0 *preserve them.*

The proof is immediate from the definitions. The next lemma is slightly more involved.

Lemma 2.6. *Let* X *and* X' *be categories with pushouts and let* $\mathsf{A} \xrightarrow{L} \mathsf{X} \xrightarrow{F_1} \mathsf{X}'$ *be functors. Then there is a* normal *lax double functor* ${}_{L}\mathbb{C}\mathsf{sp}(F_1) : {}_{L}\mathbb{C}\mathsf{sp}(\mathsf{X}) \to {}_{F_1\circ L}\mathbb{C}\mathsf{sp}(\mathsf{X}')$ *that is the identity on objects and arrows and acts on proarrows and cells by postcomposing the underlying diagrams in* X *with* $F_1 : \mathsf{X} \to \mathsf{X}'$.

The laxators are given by the universal property of pushouts in X', *and* ${}_{L}\mathbb{C}\mathsf{sp}(F_1)$ *is pseudo if and only if* F_1 *preserves pushouts. Furthermore, when* A, X, *and* X' *have finite coproducts and* L *preserves them,* ${}_{L}\mathbb{C}\mathsf{sp}(F_1)$ *is cocartesian if and only if* F_1 *is cocartesian.*

Proof. The proposed lax double functor ${}_{L}\mathbb{C}\mathsf{sp}(F_1) : {}_{L}\mathbb{C}\mathsf{sp}(\mathsf{X}) \to {}_{F_1\circ L}\mathbb{C}\mathsf{sp}(\mathsf{X}')$ acts on cospans and maps of cospans in exactly the same way as the lax double functor $\mathbb{C}\mathsf{sp}(F_1) : \mathbb{C}\mathsf{sp}(\mathsf{X}) \to \mathbb{C}\mathsf{sp}(\mathsf{X}')$ reviewed in Example A.5. The proof thus carries over directly. □

In the final lemma, we isolate the maps between structured cospan double categories induced by natural transformations between the structuring functors.

Lemma 2.7. *Let* X *be a category with pushouts and let* $\alpha : L' \Rightarrow L : \mathsf{A} \to \mathsf{X}$ *be a natural transformation. Then there is a lax double functor* $\alpha^* : {}_{L}\mathbb{C}\mathsf{sp}(\mathsf{X}) \to {}_{L'}\mathbb{C}\mathsf{sp}(\mathsf{X})$ *that acts*

- *on objects and arrows, as the identity;*
- *on proarrows* $a \nrightarrow b$, *by restricting the underlying cospan* $L(a) \to x \leftarrow L(b)$ *along the components* $\alpha_a : L'(a) \to L(a)$ *and* $\alpha_b : L'(b) \to L(b)$;
- *on cells* $\begin{array}{ccc} a & \overset{m}{\nrightarrow} & b \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ c & \underset{n}{\nrightarrow} & d \end{array}$, *by pasting the naturality squares for* f *and* g:

$$\begin{tikzcd} L'(a) \arrow[r, "\alpha_a"] \arrow[d, "L'(f)"'] & L(a) \arrow[r] \arrow[d, "L(f)"'] & x \arrow[d, "h"'] & L(b) \arrow[l] \arrow[d, "L(g)"] & L'(b) \arrow[l, "\alpha_b"'] \arrow[d, "L'(g)"] \\ L'(c) \arrow[r, "\alpha_c"'] & L(c) \arrow[r] & y & L(d) \arrow[l] & L'(d) \arrow[l, "\alpha_d"] \end{tikzcd} .$$

The laxator $\alpha^*_{m,n} : \alpha^*(m) \odot \alpha^*(n) \to \alpha^*(m \odot n)$ *for proarrows* $m = (a,\, L(a) \to x \leftarrow L(b),\, b)$ *and* $n = (b,\, L(b) \to y \leftarrow L(c),\, c)$ *has apex map given by the universal property of the pushout over* $L'(b)$:

$$\begin{tikzcd} & & y \arrow[d, "\iota'_y"'] \arrow[dr, "\iota_y"] & \\ L'(b) \arrow[r, "\alpha_b"] & L(b) \arrow[ur] \arrow[dr] & x +_{L'(b)} y \arrow[r, dashed] & x +_{L(b)} y \\ & & x \arrow[u, "\iota'_x"] \arrow[ur, "\iota_x"'] & \end{tikzcd} .$$

The unitor $\alpha_a^*: \mathrm{id}'_a \to \alpha^*(\mathrm{id}_a)$ for object $a \in \mathsf{A}$ has apex map $\alpha_a : L'(a) \to L(a)$. The lax double functor α^* is pseudo whenever α is a natural isomorphism, and it is automatically cocartesian whenever the structured cospan double categories are cocartesian.

Proof. The laxators and unitors obey the coherence axioms by the uniqueness part of the universal property. Importantly, the last statement about cocartesianness holds because natural transformations automatically commute with coproducts. That is, if A and X have finite coproducts, then, using the notation of the proof of Theorem 2.3, the following diagrams commute for all objects $a,b \in \mathsf{A}$:

$$\begin{tikzcd} L'(a)+L'(b) \arrow[r, "L'_{a,b}"] \arrow[d, "\alpha_a+\alpha_b"'] & L'(a+b) \arrow[d, "\alpha_{a+b}"] \\ L(a)+L(b) \arrow[r, "L_{a,b}"'] & L(a+b) \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} & L'(0_{\mathsf{A}}) \arrow[d, "\alpha_0"] \\ 0_{\mathsf{X}} \arrow[ur, "L'_0"] \arrow[dr, "L_0"'] & \\ & L(0_{\mathsf{A}}) \end{tikzcd} .$$

Restricting along the components of α thus commutes with restricting along the inverse comparison maps and so also commutes with coproducts of structured cospans. $\square$

Proof of Theorem 2.4. Using the three lemmas, the lax double functor $\mathbb{F} : {}_L\mathbb{C}\mathsf{sp}(\mathsf{X}) \to {}_{L'}\mathbb{C}\mathsf{sp}(\mathsf{X}')$ is realized as the composite

$$\mathbb{F} : {}_L\mathbb{C}\mathsf{sp}(\mathsf{X}) \xrightarrow{{}_L\mathbb{C}\mathsf{sp}(F_1)} {}_{F_1\circ L}\mathbb{C}\mathsf{sp}(\mathsf{X}') \xrightarrow{\alpha^*} {}_{L'\circ F_0}\mathbb{C}\mathsf{sp}(\mathsf{X}') \xrightarrow{{}_{F_0}\mathbb{C}\mathsf{sp}(\mathsf{X}')} {}_{L'}\mathbb{C}\mathsf{sp}(\mathsf{X}'). \qquad \square$$

3 Decorated Cospans as a Double Grothendieck Construction

Decorated cospans represent open systems as cospans with apexes decorated by extra data [6, 12]. For example, open dynamical systems comprise a cospan of finite sets along with a dynamical system whose set of state variables is the apex set [7]. In contrast to structured cospans, the symmetric monoidal product of decorated cospans need not satisfy a universal property such as cocartesianness. Decorated cospans are therefore applicable in certain situations where structured cospans are not, at the expense of requiring more data to construct.

The Grothendieck construction $\int F$ of a functor $F : \mathsf{A} \to \mathbf{Cat}$ can be thought to decorate the objects of A with data from F, inasmuch as the objects of $\int F$ consist of an object $a \in \mathsf{A}$ together with an object $x \in F(a)$ (the "decoration"). So one might suppose that decorated cospans arise from a Grothendieck construction. For that to be the case, the cospans being decorated must be the *objects* of a category. Fortunately, as we emphasized in Section 1, that is precisely how cospans are seen by the modern theory of double categories. In this section, we reconstruct and generalize the double category of decorated cospans using the double-categorical analogue of the Grothendieck construction.

3.1 Double Grothendieck construction

In their study of double fibrations [10], Cruttwell, Lambert, Pronk, and Szyld introduced a Grothendieck construction for double categories, taking as input a lax double functor into $\mathbb{S}\mathsf{pan}(\mathbf{Cat})$.[3]

[3] In its most general form, the double Grothendieck construction takes as input a lax double *pseudo* functor into $\mathbb{S}\mathsf{pan}(\mathbf{Cat})$, analogous to how the Grothendieck construction takes a *pseudo*functor into $\mathbf{Cat}$. For simplicity, we eschew this aspect but see [10, Definition 3.12].

Before stating the construction, we unpack some of the considerable amount of data contained in a lax double functor $F : \mathbb{A} \to \mathbb{S}\mathsf{pan}(\mathbf{Cat})$. First, there are natural transformations

$$\sigma : \mathrm{apex} \circ F_1 \Rightarrow F_0 \circ \mathrm{src} : \mathbb{A}_1 \to \mathbf{Cat} \qquad \text{and} \qquad \tau : \mathrm{apex} \circ F_1 \Rightarrow F_0 \circ \mathrm{tgt} : \mathbb{A}_1 \to \mathbf{Cat}$$

whose components are the functors σ_m and τ_m defined by

$$F_1(m) =: \Big(F_0(a) = \mathrm{ft}_L(F_1(m)) \xleftarrow{\sigma_m} \mathrm{apex}(F_1(m)) \xrightarrow{\tau_m} \mathrm{ft}_R(F_1(m)) = F_0(b) \Big)$$

for each proarrow $m : a \nrightarrow b$ in $\mathbb{A}$. The naturality squares for σ and τ are precisely the maps of spans

$$F_1(\alpha) = \left(\begin{tikzcd} F_0(a) \arrow[r, equal] \arrow[d, "F_0(f)"'] & \mathrm{ft}_L(F_1(m)) \arrow[d, "\mathrm{ft}_L(F_1(\alpha))"'] & \mathrm{apex}(F_1(m)) \arrow[l, "\sigma_m"'] \arrow[r, "\tau_m"] \arrow[d, "\mathrm{apex}(F_1(\alpha))"'] & \mathrm{ft}_R(F_1(m)) \arrow[r, equal] \arrow[d, "\mathrm{ft}_R(F_1(\alpha))"] & F_0(b) \arrow[d, "F_0(g)"] \\ F_0(c) \arrow[r, equal] & \mathrm{ft}_L(F_1(n)) & \mathrm{apex}(F_1(n)) \arrow[l, "\sigma_n"] \arrow[r, "\tau_n"'] & \mathrm{ft}_R(F_1(n)) \arrow[r, equal] & F_0(d) \end{tikzcd} \right)$$

for each cell $\begin{tikzcd} a \arrow[r, "m", "\shortmid"{marking}] \arrow[d, "f"'] \arrow[dr, phantom, "\alpha"] & b \arrow[d, "g"] \\ c \arrow[r, "n"', "\shortmid"{marking}] & d \end{tikzcd}$ in $\mathbb{A}$. Writing $F_{m,n} : F(m) \odot F(n) \to F(m \odot n)$ and $F_a : \mathrm{id}_{Fa} \to F(\mathrm{id}_a)$ for the laxators and unitors of F, there are also natural families of functors

$$\Phi_{m,n} := \mathrm{apex}(F_{m,n}) : \mathrm{apex}(F(m)) \,{}_{\tau_m}\!\times_{\sigma_n} \mathrm{apex}(F(n)) \to \mathrm{apex}(F(m \odot n))$$

and $\Phi_a := \mathrm{apex}(F_a) : F(a) \to \mathrm{apex}(F(\mathrm{id}_x))$, indexed by proarrows $a \overset{m}{\nrightarrow} b \overset{n}{\nrightarrow} c$ and objects a in $\mathbb{A}$.

Using this notation, the double Grothendieck construction [10, Theorem 3.51] appears as:

Theorem 3.1. *Given a lax double functor $F : \mathbb{A} \to \mathbb{S}\mathsf{pan}(\mathbf{Cat})$, there is a double category $\int F$, the* ***double Grothendieck construction*** *of F, with underlying categories $(\int F)_0 = \int F_0$ and $(\int F)_1 = \int (\mathrm{apex} \circ F_1)$. Explicitly, the double category $\int F$ has*

- *as objects, pairs (a,x) where a is an object of $\mathbb{A}$ and x is an object of $F(a)$;*
- *as arrows $(a,x) \to (b,y)$, pairs (f,ϕ) where $f : a \to b$ is an arrow of $\mathbb{A}$ and $\phi : F(f)(x) \to y$ is a morphism of $F(b)$;*
- *as proarrows $(a,x) \nrightarrow (b,y)$, pairs (m,s) where $m : a \nrightarrow b$ is a proarrow of $\mathbb{A}$ and s is an object of* $\mathrm{apex}(F(m))$ *such that $\sigma_m(s) = x$ and $\tau_m(s) = y$;*
- *as cells* $\begin{tikzcd} (a,x) \arrow[r, "(m,s)", "\shortmid"{marking}] \arrow[d, "(f,\phi)"'] & (b,y) \arrow[d, "(g,\psi)"] \\ (c,w) \arrow[r, "(n,t)"', "\shortmid"{marking}] & (d,z) \end{tikzcd}$*, pairs (α,ν) such that* $\begin{tikzcd} a \arrow[r, "m", "\shortmid"{marking}] \arrow[d, "f"'] \arrow[dr, phantom, "\alpha"] & b \arrow[d, "g"] \\ c \arrow[r, "n"', "\shortmid"{marking}] & d \end{tikzcd}$ *is a cell in $\mathbb{A}$ and $\nu :$* $\mathrm{apex}(F(\alpha))(s) \to t$ *is a morphism of* $\mathrm{apex}(F(n))$ *such that $\sigma_n(\nu) = \phi$ and $\tau_n(\nu) = \psi$.*

External composition and identities in $\int F$ are as follows.

- *The composite of proarrows $(a,x) \overset{(m,s)}{\nrightarrow} (b,y) \overset{(n,t)}{\nrightarrow} (c,z)$ is $(m \odot n, \Phi_{m,n}(s,t)) : (a,x) \nrightarrow (b,y)$.*
- *The external composite of cells is*

$$\begin{tikzcd} (a,x) \arrow[r, "(m,s)", "\shortmid"{marking}] \arrow[d, "(f,\phi)"'] \arrow[dr, phantom, "(\alpha,\mu)"] & (b,y) \arrow[r, "(n,t)", "\shortmid"{marking}] \arrow[d, "(g,\psi)"] \arrow[dr, phantom, "(\beta,\nu)"] & (c,z) \arrow[d, "(h,\eta)"] \\ (a',x') \arrow[r, "(m',s')"', "\shortmid"{marking}] & (b',y') \arrow[r, "(n',t')"', "\shortmid"{marking}] & (c',z') \end{tikzcd} \quad := \quad \begin{tikzcd} (a,x) \arrow[r, "(m \odot n, \Phi_{m,n}(s,t))", "\shortmid"{marking}] \arrow[d, "(f,\phi)"'] \arrow[dr, phantom, "(\alpha \odot \beta, \Phi_{m',n'}(\mu,\nu))"] & (c,z) \arrow[d, "(h,\eta)"] \\ (a',x') \arrow[r, "(m' \odot n', \Phi_{m',n'}(s',t'))"', "\shortmid"{marking}] & (c',z') \end{tikzcd} .$$

- *The identity proarrow at object* (a,x) *is* $(\mathrm{id}_a, \Phi_a(x))$.
- *The identity cell at arrow* $(f,\phi) : (a,x) \to (b,y)$ *is* $(\mathrm{id}_f, \Phi_b(\phi))$.

Moreover, there is a canonical ***projection*** $\pi_F : \int F \to \mathbb{A}$*, which is a strict double functor.*

3.2 A Modular Reconstruction of Decorated Cospans

To define decorated cospans, we apply the double Grothendieck construction in the case that the base double category $\mathbb{A}$ is a double category of cospans. Specifically, let A be a category with pushouts and let $F : \mathbb{C}\mathsf{sp}(\mathsf{A}) \to \mathbb{S}\mathsf{pan}(\mathbf{Cat})$ be a lax double functor. Then the **double category of** F**-decorated cospans**, denoted $F\mathbb{C}\mathsf{sp}$, is the double Grothendieck construction $\int F$.

This notion of decorated cospan is more general than the established one [6, §2] in two different ways. First, the decorations assigned to a cospan may depend on the whole cospan, not just on its apex. Second, the feet of the cospans receive their own decorations, which can be extracted from the cospan decorations using the transformations denoted σ and τ above. For two decorated cospans to be composable, not only must the feet of the cospans be compatible, so must be the decorations on the feet. We will see an application that takes advantage of this extra generality shortly. Before that, we show how to recover the original notion of decorated cospan based on lax monoidal functors into $(\mathbf{Cat}, \times)$.

Corollary 3.2. *Let* A *be a category with finite colimits and let* $F : (\mathsf{A},+) \to (\mathbf{Cat},\times)$ *be a lax monoidal functor. Then there is a double category* $F\mathbb{C}\mathsf{sp}$ *that has*

- *as objects, the objects of* A*;*
- *as arrows, the morphisms of* A*;*
- *as proarrows* $a \nrightarrow b$*,* F***-decorated cospans*** *with feet* a *and* b*, which are cospans* $p = (a \to m \leftarrow b)$ *in* A *together with a* ***decoration*** $s \in F(m)$*;*
- *as cells* $\begin{array}{ccc} a & \overset{(p,s)}{\nrightarrow} & b \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ c & \underset{(q,t)}{\nrightarrow} & d \end{array}$ *where* $p = (a \to m \leftarrow b)$ *and* $q = (c \to n \leftarrow d)$*,* ***morphisms of*** F***-decorated cospans*** *with foot maps* f *and* g*, which are morphisms of cospans in* A *of the form*

$$\begin{tikzcd} a \arrow[r] \arrow[d, "f"'] & m \arrow[d, "h"'] & c \arrow[l] \arrow[d, "g"] \\ b \arrow[r] & n & d \arrow[l] \end{tikzcd}$$

together with a ***decoration morphism*** $\nu : F(h)(s) \to t$ *in* $F(n)$*.*

The composite of proarrows $a \overset{(p,s)}{\nrightarrow} b \overset{(q,t)}{\nrightarrow} c$*, where* $p = (a \to m \leftarrow b)$ *and* $q = (b \to n \leftarrow c)$*, is the proarrow* $(p \odot q, \Phi_{m,n}(s,t))$*, where the cospan* $p \odot q$ *is given by pushout in* A *and the functor* $\Phi_{m,n}$ *is the composite*

$$\Phi_{m,n} : F(m) \times F(n) \xrightarrow{F_{m,n}} F(m+n) \xrightarrow{F([\iota_m,\iota_n])} F(m +_b n).$$

The identity proarrow at $a \in \mathsf{A}$ *is* (id_a, Φ_a)*, where* Φ_a *is the composite* $1 \xrightarrow{F_0} F(0) \xrightarrow{F(!)} F(a)$*.*

Moreover, there is a canonical ***projection*** $\pi_F : F\mathbb{C}\mathsf{sp} \to \mathbb{C}\mathsf{sp}(\mathsf{A})$*, which is a strict double functor.*

Proof. We construct the double category $F\mathbb{C}\mathsf{sp}$ in a modular fashion by applying the double Grothendieck construction to a lax double functor $\tilde{F} : \mathbb{C}\mathsf{sp}(\mathsf{A}) \to \mathbb{S}\mathsf{pan}(\mathbf{Cat})$ that is itself the composite of three simpler lax double functors:

$$\tilde{F} : \mathbb{C}\mathsf{sp}(\mathsf{A}) \xrightarrow{\mathrm{Apex}} \mathbb{B}(\mathsf{A},+) \xrightarrow{\mathbb{B}F} \mathbb{B}(\mathbf{Cat},\times) \xrightarrow{\mathrm{Apex}_*} \mathbb{S}\mathsf{pan}(\mathbf{Cat}).$$

Let us explain each of these. First, any monoidal category $(\mathsf{C}, \otimes, I)$ can be regarded as a double category $\mathbb{D}$ whose category of objects is trivial, $\mathbb{D}_0 = 1$; whose category of morphisms is $\mathbb{D}_1 = \mathsf{C}$; and whose external composition and identity are the monoidal product and unit [14, §3.3.4]. Lax monoidal functors then induce lax double functors between such degenerate double categories, and monoidal natural transformations induce natural transformations of those, so altogether there is a 2-functor $\mathbb{B} : \mathbf{MonCat}_{\mathrm{lax}} \to \mathbf{Dbl}_{\mathrm{lax}}$. In particular, the lax monoidal functor $F : (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ induces a lax double functor $\mathbb{B}F$.

Next, given a category A with finite colimits, the lax double functor $\mathrm{Apex} : \mathbb{C}\mathsf{sp}(\mathsf{A}) \to \mathbb{B}(\mathsf{A}, +)$ has the unique map $\mathrm{Apex}_0 : \mathsf{A} \xrightarrow{!} 1$ between categories of objects and the functor $\mathrm{Apex}_1 := \mathsf{A}^{\{\bullet \to \bullet \leftarrow \bullet\}} \xrightarrow{\text{apex}} \mathsf{A}$ between categories of morphisms. The laxators

$$\mathrm{Apex}_{p,q} : \mathrm{Apex}(p) + \mathrm{Apex}(q) = m + n \xrightarrow{[\iota_m, \iota_n]} m +_b n = \mathrm{Apex}(p \odot n)$$

for proarrows $p = (a \to m \leftarrow b)$ and $q = (b \to n \leftarrow c)$, and the unitors $\mathrm{Apex}_a : 0 \xrightarrow{!} a$ for objects $a \in \mathsf{A}$, are all given by the universal properties of the colimits involved.

Finally, given a category C with finite limits, the double functor $\mathrm{Apex}_* : \mathbb{B}(\mathsf{C}, \times) \to \mathbb{S}\mathsf{pan}(\mathsf{C})$ has underlying functors $(\mathrm{Apex}_*)_0 : 1 \to \mathsf{C}$ picking out the terminal object 1 of C and $(\mathrm{Apex}_*)_1 : \mathsf{C} \to \mathsf{C}^{\{\bullet \leftarrow \bullet \to \bullet\}}$ sending each object $c \in \mathsf{C}$ to the span $1 \xleftarrow{!} c \xrightarrow{!} 1$. This double functor is pseudo because products are isomorphic to pullbacks over the terminal object. By making reasonable choices of products and pullbacks, we can even assume that the double functor is strict.

The double category $F\mathbb{C}\mathsf{sp}$ is precisely the double Grothendieck construction of $\tilde{F}$ (Theorem 3.1). This follows from the formulas for the laxators and unitors of a composite lax double functor [14, Equation 3.63]. In terms of the notation in the corollary statement, the laxators and unitors of the composite $\tilde{F}$ are $\tilde{F}_{p,q} = \mathrm{Apex}_*(\Phi_{\mathrm{Apex}(p),\mathrm{Apex}(q)})$ and $\tilde{F}_a = \mathrm{Apex}_*(\Phi_a)$. $\square$

This result was first proved in [6, Theorem 2.1]. Our reconstruction solves a lingering conceptual puzzle about the composition law for decorated cospans: why does it involve two operations, instead of just one? As the proof shows, the reason is that decorated cospans implicitly use a *composite* of lax double functors. Specifically, laxators from the lax monoidal functor F combine with laxators from the lax double functor Apex to give the distinctive formula for composing decorations of decorated cospans.

3.3 Application: Double Category of Process Theories

An early and recurring theme of applied category theory is the mathematical modeling of physical or computational processes by monoidal categories, often with extra structure [4]. To describe a process syntactically, one can define, say by generators and relations, a small category T with the relevant structure, and then choose a particular morphism p in T. The category T defines the basic material for the process and the morphism p specifies the process itself. Regarding the category T as a theory in the sense of the categorical logic, the pair (T, p) might be called a *theory of a process*, or *process theory* for short. For example, in the author's thesis [20], a *statistical theory* is defined to be a small Markov category [13] equipped with extra linear algebraic structure, together with a distinguished morphism $p : \theta \to x$ representing the data generating process for a statistical model.

To be more precise, process theories are defined relative to a **concrete 2-category**, by which we mean a 2-category $\mathbf{C}$ equipped with a 2-functor $|-| : \mathbf{C} \to \mathbf{Cat}$, giving the **underlying category** of $\mathbf{C}$. This 2-functor will often satisfy additional properties, such as being locally faithful, but we need not assume that. Given a morphism $F : \mathsf{X} \to \mathsf{Y}$ in a concrete 2-category, we will write $F(x) := |F|(x)$ and $F(f) := |F|(f)$ for the action of the underlying functor of F on the objects and morphisms of $|\mathsf{X}|$. As an example, statistical theories are based on the concrete 2-category of small linear algebraic Markov categories, structure-preserving monoidal functors, and monoidal natural transformations [20].

Process theories can be composed once their underlying theories are made open. In the context of statistics, this composition corresponds to making hierarchical statistical models, where samples from one model become parameters of the next. To express this mathematically, we construct a double category of process theories. We need two main ingredients: the double Grothendieck construction, and an extension of the familiar construction of comma categories to a lax double functor. We now review the latter, which is interesting in its own right.

There is a lax double functor Comma : $\mathbb{C}\mathsf{sp}(\mathbf{Cat}) \to \mathbb{S}\mathsf{pan}(\mathbf{Cat})$ that is the identity on objects and arrows and sends a cospan of categories $(\mathsf{A} \xrightarrow{i} \mathsf{X} \xleftarrow{o} \mathsf{B})$ to the span of categories $(\mathsf{A} \xleftarrow{\pi_\mathsf{A}} i/o \xrightarrow{\pi_\mathsf{B}} \mathsf{B})$ comprising the comma category i/o with its canonical projections.[4] It acts on maps of cospans as

$$\begin{tikzcd} \mathsf{A} \arrow[r, "i"] \arrow[d, "H"'] & \mathsf{X} \arrow[d, "F"'] & \mathsf{B} \arrow[l, "o"'] \arrow[d, "K"] \\ \mathsf{A}' \arrow[r, "i'"'] & \mathsf{X}' & \mathsf{B}' \arrow[l, "o'"] \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} \mathsf{A} \arrow[d, "H"'] & i/o \arrow[l, "\pi_\mathsf{A}"'] \arrow[r, "\pi_\mathsf{B}"] \arrow[d, "\tilde{F}"'] & \mathsf{B} \arrow[d, "K"] \\ \mathsf{A}' & i'/o' \arrow[l, "\pi_{\mathsf{A}'}"] \arrow[r, "\pi_{\mathsf{B}'}"'] & \mathsf{B}' \end{tikzcd},$$

where the functor denoted $\tilde{F}$ sends an object $(a,\, i(a) \xrightarrow{f} o(b),\, b)$ of the comma category i/o to

$$(H(a),\, i'(H(a)) = F(i(a)) \xrightarrow{F(f)} F(o(b)) = o'(H(b)),\, H(b))$$

and a morphism (h,k) to $(H(h),K(k))$.

To describe the laxators, let $m = (\mathsf{A} \xrightarrow{i} \mathsf{X} \xleftarrow{o} \mathsf{B})$ and $n = (\mathsf{B} \xrightarrow{j} \mathsf{Y} \xleftarrow{p} \mathsf{C})$ be composable cospans of categories and let $\iota_\mathsf{X} : \mathsf{X} \to \mathsf{X} +_\mathsf{B} \mathsf{Y}$ and $\iota_\mathsf{Y} : \mathsf{Y} \to \mathsf{X} +_\mathsf{B} \mathsf{Y}$ be the inclusions into the pushout of categories. Then the apex map of the laxator $\mathrm{Comma}_{m,n}$ is the functor

$$(i/o) \times_\mathsf{B} (j/p) \to (\iota_\mathsf{X} \circ i)/(\iota_\mathsf{Y} \circ p)$$

that sends a pair of objects (a,f,b) and (b,g,c) with $o(b) = j(b)$ to $(a, \iota_\mathsf{Y}(g) \circ \iota_\mathsf{X}(f), c)$, which is well-defined since $\iota_\mathsf{X}(o(b)) = \iota_\mathsf{Y}(j(b))$ in $\mathsf{X} +_\mathsf{B} \mathsf{Y}$. This functor sends a pair of maps (h,k) and (k,ℓ) to the map (h,ℓ). Finally, given a category A, the apex map of the unitor $\mathrm{Comma}_\mathsf{A}$ is the functor $\mathsf{A} \to 1_\mathsf{A}/1_\mathsf{A}$ that sends an object $a \in \mathsf{A}$ to $(a, 1_a, a)$ and a morphism h to (h,h).

Proposition 3.3. *Let* $\mathbf{C}$ *be a concrete 2-category with pushouts. Then there is a double category that has*

- *as objects, an object* A *in* $\mathbf{C}$ *together with an object* $a \in |\mathsf{A}|$*;*
- *as arrows* $(\mathsf{A},a) \to (\mathsf{A}',a')$*, a morphism* $H : \mathsf{A} \to \mathsf{A}'$ *in* $\mathbf{C}$ *together with a morphism* $h' : H(a) \to a'$ *in* $|\mathsf{A}'|$*;*
- *as proarrows* $(\mathsf{A},a) \nrightarrow (\mathsf{B},b)$*, a cospan in* $\mathbf{C}$ *of form* $m = (\mathsf{A} \xrightarrow{i} \mathsf{X} \xleftarrow{o} \mathsf{B})$ *along with a morphism* $f : i(a) \to o(b)$ *in* $|\mathsf{X}|$*;*

[4] The lax double functor Comma : $\mathbb{C}\mathsf{sp}(\mathbf{C}) \to \mathbb{S}\mathsf{pan}(\mathbf{C})$ even generalizes from $\mathbf{C} = \mathbf{Cat}$ to any 2-category $\mathbf{C}$ with comma objects, pushouts, and pullbacks [14, §4.5.9], although we will not use that.

- *as cells* $\begin{array}{ccc} (\mathsf{A},a) & \overset{(m,f)}{\nrightarrow} & (\mathsf{B},b) \\ {\scriptstyle (H,h')}\downarrow & & \downarrow{\scriptstyle (K,k')} \\ (\mathsf{A}',a') & \underset{(m',f')}{\nrightarrow} & (\mathsf{B}',b') \end{array}$, *a morphism* $F : \mathsf{X} \to \mathsf{X}'$ *forming a map of cospans* $\begin{array}{ccccc} \mathsf{A} & \xrightarrow{i} & \mathsf{X} & \xleftarrow{o} & \mathsf{B} \\ {\scriptstyle H}\downarrow & & {\scriptstyle F}\downarrow & & \downarrow{\scriptstyle K} \\ \mathsf{A}' & \underset{i'}{\rightarrow} & \mathsf{X}' & \underset{o'}{\leftarrow} & \mathsf{B}' \end{array}$

  in $\mathbf{C}$ *and making the following square in* $|\mathsf{X}'|$ *commute:*

$$\begin{array}{ccc} i'(H(a)) = F(i(a)) \overset{F(f)}{\longrightarrow} F(o(b)) = o'(K(b)) \\ {\scriptstyle i'(h')}\downarrow \qquad\qquad\qquad\qquad \downarrow{\scriptstyle o'(k')} \\ i'(a') \xrightarrow{\qquad\qquad f' \qquad\qquad} o'(b') \end{array} .$$

Two proarrows $(\mathsf{A},a) \overset{(m,f)}{\nrightarrow} (\mathsf{B},b) \overset{(n,g)}{\nrightarrow} (\mathsf{C},c)$, *with* $m = (\mathsf{A} \xrightarrow{i} \mathsf{X} \xleftarrow{o} \mathsf{B})$ *and* $n = (\mathsf{B} \xrightarrow{j} \mathsf{Y} \xleftarrow{p} \mathsf{C})$, *have composite* $(m \odot n, h) : (\mathsf{A},a) \nrightarrow (\mathsf{C},c)$, *where* $m \odot n$ *is the composite cospan in* $\mathbf{C}$ *with apex* $\mathsf{X} +_{\mathsf{B}} \mathsf{Y}$ *and* h *is given by first composing the images of* f *and* g *in* $|\mathsf{X}| +_{|\mathsf{B}|} |\mathsf{Y}|$ *and then applying the canonical functor* $|\mathsf{X}| +_{|\mathsf{B}|} |\mathsf{Y}| \to |\mathsf{X} +_{\mathsf{B}} \mathsf{Y}|$. *The identity proarrow at* (A,a) *is* $(\mathrm{id}_{\mathsf{A}}, 1_a)$.

Proof. Apply the double Grothendieck construction to the composite lax double functor

$$\mathbb{C}\mathsf{sp}(\mathbf{C}) \xrightarrow{\mathbb{C}\mathsf{sp}(|-|)} \mathbb{C}\mathsf{sp}(\mathbf{Cat}) \xrightarrow{\text{Comma}} \mathbb{S}\mathsf{pan}(\mathbf{Cat}).$$

Here the lax double functor $\mathbb{C}\mathsf{sp}(|-|)$ is a particular case of Example A.5. □

4 Conclusion

We have revisited structured and decorated cospans from the perspective of double category theory, showing that double categories of structured cospans form cocartesian equipments and that their maps are cocartesian double functors. We have also reconstructed and generalized double categories of decorated cospans using the double Grothendieck construction.

Looking to future developments, we have presented a reasonably complete and self-contained treatment of the theory of structured cospans, but less so for the theory of decorated cospans. We have not shown how to construct maps between double categories of decorated cospans, along the lines of Baez et al's [6, Theorem 2.5]. Just as the classical Grothendieck construction for categories is 2-functorial [21, §6], so should be the Grothendieck construction for double categories, which should in turn directly produce maps between decorated cospan double categories and natural transformations between those. Equally importantly, we have not recovered the symmetric monoidal product of decorated cospans, an absence clearly felt in our example of the double category of process theories. Monoidal products should be obtained as a corollary of a hypothetical Grothendieck construction for monoidal double categories, combining the monoidal and double Grothendieck constructions [10, 19]. In these and other ways, we expect the further development of the theory of double categories to immediately impact the study of open systems, simplifying known constructions, suggesting new ones, and enabling practitioners to focus on applications rather than general theoretical issues.

Acknowledgments This project was partially supported by the Air Force Office of Scientific Research (AFOSR) Young Investigator Program (YIP) through Award FA9550-23-1-0133. I thank Nathanael Arkor, John Baez, and Brandon Shapiro for helpful conversations. I am also grateful to Brandon Shapiro for comments on early versions of this work.

References

[1] Rebekah Aduddell, James Fairbanks, Amit Kumar, Pablo S. Ocal, Evan Patterson & Brandon T. Shapiro (2023): *A compositional account of motifs, mechanisms, and dynamics in biochemical regulatory networks*, arXiv: 2301.01445.

[2] Evangelia Aleiferi (2018): *Cartesian double categories with an emphasis on characterizing spans*, PhD thesis. Dalhousie University, arXiv: 1809.06940.

[3] John Baez, Xiaoyan Li, Sophie Libkind, Nathaniel Osgood & Evan Patterson (2022): *Compositional modeling with stock and flow diagrams*. In *Proceedings of the Applied Category Theory Conference 2022*, arXiv: 2205.08373. In press.

[4] John Baez & Mike Stay (2010): *Physics, topology, logic and computation: a Rosetta Stone*. In *New structures for physics*, pages 95–172. doi: 10.1007/978-3-642-12821-9_2, arXiv: 0903.0340.

[5] John C. Baez & Kenny Courser (2020): *Structured cospans*. *Theory and Applications of Category Theory*. 35(48), pages 1771–1822, arXiv: 1911.04630. Available at http://www.tac.mta.ca/tac/volumes/35/48/35-48abs.html.

[6] John C. Baez, Kenny Courser & Christina Vasilakopoulou (2022): *Structured versus decorated cospans*. *Compositionality*. 4(3), doi: 10.32408/compositionality-4-3, arXiv: 2101.09363.

[7] John C. Baez & Blake S. Pollard (2017): *A compositional framework for reaction networks*. *Reviews in Mathematical Physics*. 29(09), page 1750028, doi: 10.1142/S0129055X17500283, arXiv: 1704.02051.

[8] Aurelio Carboni, G. Max Kelly & Richard J. Wood (1991): *A 2-categorical approach to change of base and geometric morphisms I*. *Cahiers de topologie et géométrie différentielle catégoriques*. 32(1), pages 47–95.

[9] Kenny Courser (2020): *Open systems: A double categorical perspective*, PhD thesis. University of California, Riverside, arXiv: 2008.02394.

[10] Geoffrey Cruttwell, Michael Lambert, Dorette Pronk & Martin Szyld (2022): *Double Fibrations*. *Theory and Applications of Categories*. 38(35), pages 1326–1394, arXiv: 2205.15240. Available at http://www.tac.mta.ca/tac/volumes/38/35/38-35abs.html.

[11] José Luiz Fiadeiro & Vincent Schmitt (2007): *Structured co-spans: an algebra of interaction protocols*. In *International Conference on Algebra and Coalgebra in Computer Science (CALCO 2007)*, pages 194–208, doi: 10.1007/978-3-540-73859-6_14.

[12] Brendan Fong (2015): *Decorated cospans*. *Theory and Applications of Categories*. 30(33), pages 1096–1120, arXiv: 1502.00872. Available at http://www.tac.mta.ca/tac/volumes/30/33/30-33abs.html.

[13] Tobias Fritz (2020): *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. *Advances in Mathematics*. 370, page 107239, doi: 10.1016/j.aim.2020.107239, arXiv: 1908.07021.

[14] Marco Grandis (2019): *Higher dimensional categories: From double to multiple categories*. World Scientific. doi: 10.1142/11406.

[15] Marco Grandis & Robert Paré (1999): *Limits in double categories*. *Cahiers de topologie et géométrie différentielle catégoriques*. 40(3), pages 162–220. Available at http://www.numdam.org/item/CTGDC_1999__40_3_162_0/.

[16] Linde Wester Hansen & Michael Shulman (2019): *Constructing symmetric monoidal bicategories functorially*, arXiv: 1910.09240.

[17] Piergiulio Katis, Nicoletta Sabadini & Robert F. C. Walters (1997): *Span(Graph): A categorical algebra of transition systems*. In *Algebraic Methodology and Software Technology. AMAST 1997*, Springer, pages 307–321, doi: 10.1007/BFb0000479.

[18] Sophie Libkind, Andrew Baas, Micah Halter, Evan Patterson & James P. Fairbanks (2022): *An algebraic framework for structured epidemic modelling*. *Philosophical Transactions of the Royal Society A*. 380(2233), page 20210309, doi: 10.1098/rsta.2021.0309, arXiv: `2203.16345`.

[19] Joe Moeller & Christina Vasilakopoulou (2020): *Monoidal Grothendieck construction*. *Theory and Applications of Categories*. 35(31), pages 1159–1207, arXiv: `1809.00727`. Available at `http://www.tac.mta.ca/tac/volumes/35/31/35-31abs.html`.

[20] Evan Patterson (2020): *The algebra and machine representation of statistical models*, PhD thesis. Stanford University, arXiv: `2006.08945`.

[21] George Peschke & Walter Tholen (2020): *Diagrams, fibrations, and the decomposition of colimits*, arXiv: `2006.10890`.

[22] Robert Rosebrugh, Nicoletta Sabadini & Robert F. C. Walters (2005): *Generic commutative separable algebras and cospans of graphs*. *Theory and Applications of Categories*. 15(6), pages 164–177. Available at `http://www.tac.mta.ca/tac/volumes/15/6/15-06abs.html`.

[23] Michael Shulman (2008): *Framed bicategories and monoidal fibrations*. *Theory and Applications of Categories*. 20(18), pages 650–738, arXiv: `0706.1286`. Available at `http://www.tac.mta.ca/tac/volumes/20/18/20-18abs.html`.

[24] Michael Shulman (2010): *Constructing symmetric monoidal bicategories*, arXiv: `1004.0993`.

A Cocartesian Equipments

In this appendix, we review cocartesian double categories and equipments, and the maps between them. This material is known but may not be straightforward to access in the literature. It is included for the reader's convenience.

Just as a cocartesian category is (on one standard definition) a category with finite coproducts, a cocartesian double category is a double category with finite double-categorical coproducts. A highly conceptual way to make this precise is to define a cocartesian double category to be a cocartesian object in the 2-category **Dbl** of double categories, double functors, and natural transformations. Thus, a double category $\mathbb{D}$ is **cocartesian** if the diagonal and terminal double functors, $\Delta_{\mathbb{D}} : \mathbb{D} \to \mathbb{D} \times \mathbb{D}$ and $!_{\mathbb{D}} : \mathbb{D} \to \mathbb{1}$, have left adjoints in **Dbl**. This is (dual to) the approach taken by Aleiferi in her PhD thesis on cartesian double categories [2]. It will be convenient for us to have a more concrete description.[5]

Definition A.1. A double category $\mathbb{D}$ is **cocartesian** if its underlying categories $\mathbb{D}_0$ and $\mathbb{D}_1$ have finite coproducts; the source and target functors $\mathrm{src}, \mathrm{tgt} : \mathbb{D}_1 \rightrightarrows \mathbb{D}_0$ preserve finite coproducts; and the external composition $\odot : \mathbb{D}_1 \times_{\mathbb{D}_0} \mathbb{D}_1 \to \mathbb{D}_1$ and unit $\mathrm{id} : \mathbb{D}_0 \to \mathbb{D}_1$ also preserve finite coproducts, meaning that for all proarrows $x \overset{m}{\nrightarrow} y \overset{n}{\nrightarrow} z$ and $x' \overset{m'}{\nrightarrow} y' \overset{n'}{\nrightarrow} z'$ and objects x and x' in $\mathbb{D}$, the canonical comparison cells

$$\begin{tikzcd} x+x' \arrow[r, "(m\odot n)+(m'\odot n')", "\shortmid"{marking}] \arrow[d, equal] \arrow[dr, phantom, "{[\iota_m \odot \iota_n, \iota_{m'} \odot \iota_{n'}]}"] & z+z' \arrow[d, equal] \\ x+x' \arrow[r, "(m+m')\odot(n+n')"', "\shortmid"{marking}] & z+z' \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} x+x' \arrow[r, "\mathrm{id}_x+\mathrm{id}_{x'}", "\shortmid"{marking}] \arrow[d, equal] \arrow[dr, phantom, "{[\mathrm{id}_{\iota_x}, \mathrm{id}_{\iota_{x'}}]}"] & x+x' \arrow[d, equal] \\ x+x' \arrow[r, "\mathrm{id}_{x+x'}"', "\shortmid"{marking}] & x+x' \end{tikzcd}$$

[5]The equivalence of the two definitions follows from a general result about double adjunctions [14, Corollary 4.3.7].

given by the universal property of binary coproducts, as well as the comparison cells $0_{\mathbb{D}_1} \xrightarrow{!} 0_{\mathbb{D}_1} \odot 0_{\mathbb{D}_1}$ and $0_{\mathbb{D}_1} \xrightarrow{!} \mathrm{id}_{0_{\mathbb{D}_0}}$ given by the universal property of initial objects, are all isomorphisms in $\mathbb{D}_1$.

An equipment, also known as a fibrant double category or a framed bicategory, is a double category in which proarrows can be restricted or extended along pairs of arrows in a universal way. Equipments can be defined in at least three equivalent ways [23, Theorem 4.1], including as follows.

Definition A.2. An **equipment** is a double category $\mathbb{D}$ such that the pairing of the source and target functors, $\langle s,t\rangle : \mathbb{D}_1 \to \mathbb{D}_0 \times \mathbb{D}_0$, is a fibration.

Elaborating the definition, a double category $\mathbb{D}$ is an equipment if every niche in $\mathbb{D}$ of the form on the left can be completed to a cell as on the right

$$\begin{tikzcd} x \arrow[d, "f"'] & y \arrow[d, "g"] \\ w \arrow[r, "h"', "\shortmid" marking] & z \end{tikzcd} \quad \rightsquigarrow \quad \begin{tikzcd} x \arrow[r, "\mathrm{res}_g^f(n)", "\shortmid" marking] \arrow[d, "f"'] \arrow[dr, phantom, "\mathrm{res}"] & y \arrow[d, "g"] \\ w \arrow[r, "h"', "\shortmid" marking] & z \end{tikzcd}$$

called a **restriction** cell, with the universal property that for every pair of arrows $h : x' \to x$ and $k : y' \to y$, each cell α of the form on the left factors uniquely through the restriction cell as on the right:

$$\begin{tikzcd} x' \arrow[r, "m'", "\shortmid" marking] \arrow[d, "h"'] & y' \arrow[d, "k"] \\ x \arrow[r, phantom, "\alpha"] \arrow[d, "f"'] & y \arrow[d, "g"] \\ w \arrow[r, "h"', "\shortmid" marking] & z \end{tikzcd} \quad = \quad \begin{tikzcd} x' \arrow[r, "m'", "\shortmid" marking] \arrow[d, "h"'] \arrow[dr, phantom, "\exists!"] & y' \arrow[d, "k"] \\ x \arrow[r, "\mathrm{res}_g^f(n)", "\shortmid" marking] \arrow[d, "f"'] \arrow[dr, phantom, "\mathrm{res}"] & y \arrow[d, "g"] \\ w \arrow[r, "h"', "\shortmid" marking] & z \end{tikzcd}$$

Finally, a **cocartesian equipment** is a double category that is both cocartesian and an equipment. We emphasize again that being a cocartesian equipment is a property of, not a structure on, a double category.

Example A.3 (Cospan double categories)**.** The prototypical example of a cocartesian equipment is none other than $\mathbb{C}\mathsf{sp}(\mathsf{S})$, the double category of cospans in a category S with finite colimits. Let us sketch the proof behind this statement. For a more detailed proof, one can dualize the proof in Aleiferi's thesis that $\mathbb{S}\mathsf{pan}(\mathsf{S})$, for a category S with finite limits, is a cartesian equipment [2].

Finite coproducts in the category $\mathbb{C}\mathsf{sp}(\mathsf{S})_0 = \mathsf{S}$ exist by assumption, and finite coproducts in the functor category $\mathbb{C}\mathsf{sp}(\mathsf{S})_1 = \mathsf{S}^{\{\bullet\to\bullet\leftarrow\bullet\}}$ are computed pointwise in S. So the source and target functors $\mathrm{ft}_L, \mathrm{ft}_R : \mathbb{C}\mathsf{sp}(\mathsf{S})_1 \to \mathsf{S}$, extracting the left and right feet, preserve coproducts. The comparison cells are isomorphisms because colimits commute with colimits (specifically, pushouts commute with coproducts) up to canonical isomorphism. Thus, the double category of cospans is cocartesian.

It is also an equipment. To restrict a cospan $c \xrightarrow{\ell} y \xleftarrow{r} d$ along a pair of morphisms $f : a \to c$ and $g : b \to d$, simply compose the morphisms with the legs of the cospan. The restriction cell is trivial:

$$\begin{tikzcd} a \arrow[r, "\ell\circ f"] \arrow[d, "f"'] & y \arrow[d, equal] & b \arrow[l, "r\circ g"'] \arrow[d, "g"] \\ c \arrow[r, "\ell"'] & y & d \arrow[l, "r"] \end{tikzcd} .$$

We turn now to maps between cocartesian double categories and equipments. Since a cocartesian category is a cocartesian object in **Dbl**, a map between cocartesian double categories can be defined abstractly as a cocartesian morphism between cocartesian objects [8, §5.2]. As before, this definition reduces to a more concrete one:

Definition A.4. A double functor $F : \mathbb{D} \to \mathbb{E}$ between cocartesian double categories is **cocartesian** if both underlying functors $F_0 : \mathbb{D}_0 \to \mathbb{E}_0$ and $F_1 : \mathbb{D}_1 \to \mathbb{E}_1$ preserve finite coproducts.

Note that we will apply this definition to lax as well as pseudo double functors.

Perhaps surprisingly, no extra conditions on double functors between equipments are required. Any (op)lax double functor between equipments automatically preserves restriction (respectively, extension) cells, as proved by Shulman [23, Proposition 6.4]. In particular, a pseudo double functor between equipments preserves all the operations afforded by an equipment.

Example A.5 (Maps between cospan double categories)**.** The construction of the double category of cospans $\mathbb{C}\mathsf{sp}(\mathsf{S})$ extends to a 2-functor $\mathbb{C}\mathsf{sp} : \mathbf{Cat}_{\mathrm{po}} \to \mathbf{Dbl}_{\mathrm{lax}}$, where $\mathbf{Cat}_{\mathrm{po}}$ is the 2-category of categories with chosen pushouts, arbitrary functors, and natural transformations and $\mathbf{Dbl}_{\mathrm{lax}}$ is the 2-category of double categories, lax double functors, and natural transformations [14, §C3.11].

Let us describe the lax double functor $\mathbb{C}\mathsf{sp}(F) : \mathbb{C}\mathsf{sp}(\mathsf{S}) \to \mathbb{C}\mathsf{sp}(\mathsf{S}')$ induced by a functor $F : \mathsf{S} \to \mathsf{S}'$ between categories with pushouts. We have $\mathbb{C}\mathsf{sp}(F)_0 = F$ on objects and arrows, while $\mathbb{C}\mathsf{sp}(F)_1$ post-composes with F the diagrams defining cospans and maps of cospans in S. Since functors preserve identities, $\mathbb{C}\mathsf{sp}(F)$ is a *normal* lax double functor, meaning that it preserves identity proarrows strictly. Given cospans $m = (a \to x \leftarrow b)$ and $n = (b \to y \leftarrow c)$ in S, the laxator

$$\mathbb{C}\mathsf{sp}(F)_{m,n} : \mathbb{C}\mathsf{sp}(F)(m) \odot \mathbb{C}\mathsf{sp}(F)(n) \to \mathbb{C}\mathsf{sp}(F)(m \odot n)$$

has apex map given by the universal property of the pushout in S':

$$\begin{tikzcd}
 & Fb \arrow[dl] \arrow[dr] & \\
Fx \arrow[r, "\iota_{Fx}"] \arrow[dr, bend right, "F(\iota_x)"'] & Fx +_{Fb} Fy \arrow[d, dashed] & Fy \arrow[l, "\iota_{Fy}"'] \arrow[dl, bend left, "F(\iota_y)"] \\
 & F(x +_b y) &
\end{tikzcd} .$$

Clearly, $\mathbb{C}\mathsf{sp}(F)$ is pseudo if and only if F preserves pushouts.

Suppose that S and S' have all finite colimits, so that their double categories of cospans are cocartesian. Since coproducts of cospans are computed pointwise, $\mathbb{C}\mathsf{sp}(F)$ is a cocartesian lax double functor exactly when F preserves finite coproducts. Altogether, $\mathbb{C}\mathsf{sp}(F)$ is a cocartesian (pseudo) double functor if and only if F preserves all finite colimits.

Obstructions to Compositionality

Caterina Puca
Quantinuum*
caterina.puca@quantinuum.com

Amar Hadzihasanovic
[1] Quantinuum*
[2] Tallinn University of Technology
amar.hadzihasanovic@quantinuum.com

Fabrizio Genovese
20squares
noreply@20squares.xyz

Bob Coecke
Quantinuum*
bob.coecke@quantinuum.com

Compositionality is at the heart of computer science and several other areas of applied category theory such as computational linguistics, categorical quantum mechanics, interpretable AI, dynamical systems, compositional game theory, and Petri nets. However, the meaning of the term seems to vary across the many different applications. This work contributes to understanding, and in particular qualifying, different kinds of compositionality.

Formally, we introduce invariants of categories that we call zeroth and first homotopy posets, generalising in a precise sense the π_0 and π_1 of a groupoid. These posets can be used to obtain a qualitative description of how far an object is from being terminal and a morphism is from being iso. In the context of applied category theory, this formal machinery gives us a way to qualitatively describe the "failures of compositionality", seen as failures of certain (op)lax functors to be strong, by classifying obstructions to the (op)laxators being isomorphisms.

Failure of compositionality, for example for the interpretation of a categorical syntax in a semantic universe, can both be a bad thing and a good thing, which we illustrate by respective examples in graph theory and quantum theory.

Acknowledgements A.H. 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. We thank Sean Tull and Robin Lorenz for helpful comments on an earlier draft.

Introduction

Compositionality is probably the most relevant principle in applied category theory (ACT) research. While there is no unified definition [11, 9, 3], it refers, broadly speaking, to certain forms of relation between properties, behaviours, or observations of a composite system on one hand, and those of its components on the other. A common concern, in this context, is whether it is possible to derive properties of the whole from properties of its parts, and vice versa. In some cases, both directions are viable and inverse to each other, in which case a property is "fully compositional". More frequently, only one direction is viable.

The need to formally quantify and/or qualify compositionality has been widely discussed in the ACT community at least since 2018 [8], as researchers became increasingly aware of various "failures of compositionality", and wished to classify them beyond a simple yes-or-no statement.

Let us be more precise. Much research in ACT has been devoted to the study of *open systems*, that is, entities with open interfaces that can be composed with other entities of the same kind. This approach

*17 Beaumont Street, Oxford OX1 2NA, United Kingdom

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 226–245, doi:10.4204/EPTCS.397.14

has been pervasive, and has been applied in the study of *categorical quantum mechanics* [1], *natural language* [4], *dynamical systems* [7], *Petri nets* [2], *game theory* [9] and many other subjects. When studying open systems, it is not rare to define functors mapping a "theory of boxes" — in the form of a monoidal category or bicategory — where the composition rules of the systems are defined, to a certain "semantic universe" of properties or behaviours of the systems. The properties of these functors reflect how well the information that they capture adheres to the composition rules: a *lax* functor P, with structural *laxator* morphisms in the direction $\mathsf{P}f \mathbin{⨟} \mathsf{P}g \to \mathsf{P}(f \mathbin{⨟} g)$, means that one can derive information on the whole system from information on its components; an *oplax* functor, with structural morphisms in the direction $\mathsf{P}(f \mathbin{⨟} g) \to \mathsf{P}f \mathbin{⨟} \mathsf{P}g$, means that one can derive information on the components from information on the whole; while a *strong* functor means that the information on components and the information on the whole completely determine each other.

For example, the functor sending *open graphs* to their *reachability relation* (see Section 3.1) is lax, which tells us that the reachability relation of a composition of open graphs can be strictly bigger than the composition of the reachability relations defined on its parts. This is considered undesirable from a computational viewpoint, as it means that one cannot reconstruct the reachability of a graph by separately computing the reachability of its components.

On the other hand, in "Schrödinger compositionality" (covered in Section 3.2), quantum-mechanical behaviour arises from the laxity of the functor mapping each object to its set of states. This laxity implies that not all quantum states are separable, which is desirable, as it unlocks the use of *entanglement* as a resource unavailable in classical mechanics.

In both cases, laxity represents a "failure of compositionality" which has both practical and foundational importance: the "gap" between a lax and a strong functor represents the gap between what we can compute compositionally with a "divide-and-conquer" strategy and what we cannot, or the gap between a classical and non-classical theory of processes. In this light, the question: *how can we qualify (failures of) compositionality?* becomes the question: *how far is a lax functor from being strong?*[1] In this paper, we attempt to give a structured answer to the question. Our chain of reasoning is the following.

Definition 1. A lax functor is strong when all the components of its laxators are isomorphisms.

Thus, we can think of reducing our question to the more general one: *how far is a morphism from being an isomorphism?*[2] Let us use the following, well-known characterisation of isomorphisms.

Proposition 1. *A morphism $f\colon X \to Y$ in a category C is an isomorphism if and only if it is terminal as an object of the slice category $C/_Y$.*

This allows us to reduce further to the question: *how far is an object from being terminal?* Terminality can be split into the following pair of properties.

Definition 2. An object $\mathbb{1}$ in a category C is

- *weak terminal* if, for all objects X of C, there exists a morphism $X \to \mathbb{1}$;
- *subterminal* if, for all parallel pairs of morphisms $f,g\colon X \to \mathbb{1}$, we have $f = g$.

Hence, to describe how far $\mathbb{1}$ is from being terminal, we can separately describe how far $\mathbb{1}$ is from being weak terminal and subterminal, respectively.

Following this chain of reasoning, we focus on classifying *obstructions to weak terminality and subterminality* for objects in arbitrary categories. Surprisingly, it turns out that there exists a natural way of

[1] We will focus on lax functors in our discussion, but everything can be dualised to oplax functors.

[2] This approach, and the fact that it could be investigated with homotopical methods, was first suggested to us by Jules Hedges.

associating certain *pointed posets* to a pointed category (category with a chosen object), which we call the *zeroth* and *first homotopy poset*, because in a precise sense they generalise the π_0 and π_1 of a pointed groupoid seen as a homotopy 1-type. This opens up the possibility of an *invariant-based* approach to the formal study of compositionality: the homotopy posets contain no information that is not already in the functors and categories, but put it in a form which may be more tractable and intelligible.

In Section 1, we give the definitions of homotopy posets and state their basic properties, demonstrating in which sense they answer our question about terminal objects. In Section 2, going backwards in our chain of reasoning, we apply them to the study of obstructions to morphisms being iso. Finally, in Section 3, we sketch through a couple of simple examples how our framework can be applied to the study of failures of compositionality, seen as failures of certain (op)lax functors to be strong. Some particularly involved proofs are collected in the Appendix; we refer to the extended version [13] for other proofs and further details.

1 Homotopy posets

To begin, we focus on obstructions to weak terminality. Having fixed a category C, we interpret objects of a category C as points, and morphisms between them as paths. From this point of view, a weak terminal object is an object that is always reachable from any generic object x in C.

Intuitively, we can fix a "weak terminal object candidate"[3] $\mathbb{1}$ and consider any object x such that there is *no* morphism $x \to \mathbb{1}$ as an *obstruction to weak terminality*. Moreover:

- If x, y are obstructions for $\mathbb{1}$, and there are morphisms $x \to y$ and $y \to x$, we regard them as equivalent: if there were a morphism $x \to \mathbb{1}$ there would be a morphism $y \to \mathbb{1}$, and vice versa.
- If x, y are obstructions for $\mathbb{1}$ and there is a morphism $x \to y$, then we regard x as a "more fundamental obstruction than y". This is because, if there were a morphism $y \to \mathbb{1}$, we would automatically obtain a morphism $x \to \mathbb{1}$ by composition (one can "go from x to y and then to $\mathbb{1}$"), while the opposite is not true.

We will devote this section to making this intuition formal.

Definition 3 (Poset reflection)**.** Let **Pos** be the large[4] category of posets and order-preserving maps. There is a full and faithful functor $\iota\colon \mathbf{Pos} \hookrightarrow \mathbf{Cat}$, whose image consists of the categories that are

- *thin* (each hom-set contains at most one morphism), and
- *skeletal* (every isomorphism is an automorphism).

The *poset reflection* $\|C\|$ of a category C is its image under the left adjoint $\|-\|\colon \mathbf{Cat} \to \mathbf{Pos}$ to ι:

- the elements of $\|C\|$ are equivalence classes $\|x\|$ of objects x of C, where $\|x\| = \|y\|$ if and only if there exist morphisms $x \to y$ and $y \to x$ in C, and
- $\|x\| \leq \|y\|$ if and only if there exists a morphism $x \to y$ in C.

Proposition 2. *Let C be a category and $\mathbb{1}$ an object in C. The following are equivalent:*

(a) $\mathbb{1}$ is a weak terminal (respectively, initial) object in C;

[3] In this paper, we will use $\mathbb{1}$ to denote "terminal object candidates", that is, objects for which we want to investigate how far they are from being terminal. For an object that we know or presume to be terminal, we will instead use the notation $\mathbf{1}$.

[4] We will denote categories in *italics* and large categories in **bold**. Note that in our constructions, what matters is only the *relative* size: a construction which associates a poset to a category can be applied to a large category, producing a large poset.

(b) $\|\mathbb{1}\|$ *is the greatest (respectively, least) element of* $\|C\|$.

Definition 4 (Arrow category)**.** Let $\vec{I}$ be the "walking arrow" category, that is, the free category on the graph

$$0 \xrightarrow{a} 1 \ .$$

The *arrow category* of a category C is the functor category $C^{\vec{I}}$. Explicitly, the objects of $C^{\vec{I}}$ are morphisms of C, while morphisms of $C^{\vec{I}}$ are commutative squares in C. There are functors dom, cod: $C^{\vec{I}} \to C$ which, given a morphism (h_0, h_1), return h_0, respectively, h_1.

Definition 5 (Category of pointed objects)**.** Let C be a category with a chosen terminal object $\mathbf{1}$. A *pointed object* (x, v) of C is an object x of C together with a morphism $v\colon \mathbf{1} \to x$, called its *basepoint*. The *category of pointed objects* of C — denoted by $C_\bullet$ — is the coslice category $\mathbf{1}/C$.

Proposition 3 (Functoriality of arrow and pointed objects categories)**.** *Let* $\mathsf{F}\colon C \to D$ *be a functor. Then* F *lifts to a functor* $\mathsf{F}^{\vec{I}}\colon C^{\vec{I}} \to D^{\vec{I}}$ *using the pointwise action of* F *on* C.

If moreover C *and* D *have a chosen terminal object, and if* F *preserves it, then it also lifts to a functor* $\mathsf{F}_\bullet\colon C_\bullet \to D_\bullet$ *sending a pointed object* (x, v) *of* C *to* $(\mathsf{F}x, \mathsf{F}v)$*, a pointed object of* D.

Definition 6 (Quotient of an object by a morphism)**.** Let C be a category with chosen pushouts and a terminal object $\mathbf{1}$. Given a morphism $f\colon x \to y$, the *quotient of* y *by* f is the pushout

$$\begin{tikzcd} x \arrow[r, "!"] \arrow[d, "f"'] & \mathbf{1} \arrow[d, "{[x]}"] \\ y \arrow[r] & y \,/\!\!/\, f \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

where $!\colon x \to \mathbf{1}$ is the unique morphism from x to the terminal object.

Proposition 4 (Functoriality of the quotient)**.** *If* C *has chosen pushouts and a terminal object* $\mathbf{1}$*, then for each morphism* $f\colon x \to y$ *in* C Definition 6 *determines a pointed object* $\mathsf{Q}(f) := (y \,/\!\!/\, f, [x])$ *of* C*. This extends to a functor* $\mathsf{Q}\colon C^{\vec{I}} \to C_\bullet$*. If both* C *and* D *have chosen pushouts and a chosen terminal object* $\mathbf{1}$*, and if* F *preserves them, then* F *induces a commutative square of functors*

$$\begin{tikzcd} C^{\vec{I}} \arrow[r, "\mathsf{Q}"] \arrow[d, "\mathsf{F}^{\vec{I}}"'] & C_\bullet \arrow[d, "\mathsf{F}_\bullet"] \\ D^{\vec{I}} \arrow[r, "\mathsf{Q}"'] & D_\bullet . \end{tikzcd}$$

The categories **Cat** and **Pos** have all limits and colimits, so in particular they have pushouts and a terminal object. The poset reflection functor $\|-\|\colon \mathbf{Cat} \to \mathbf{Pos}$ sends the terminal category to the terminal poset, and preserves pushouts, since it is a left adjoint. The preservation can be made strict with respect to a choice on both sides. We are in the conditions of Proposition 4: there is a commutative square

$$\begin{tikzcd} \mathbf{Cat}^{\vec{I}} \arrow[r, "\mathsf{Q}"] \arrow[d, "\|-\|^{\vec{I}}"'] & \mathbf{Cat}_\bullet \arrow[d, "\|-\|_\bullet"] \\ \mathbf{Pos}^{\vec{I}} \arrow[r, "\mathsf{Q}"] & \mathbf{Pos}_\bullet . \end{tikzcd} \tag{1}$$

We are now ready to define the object of interest of this section.

Definition 7 (Zeroth homotopy poset)**.** Let C be a category and x an object in C. The *zeroth homotopy poset of C over x* is the pointed poset

$$(\pi_0(C/_x),\ [x])$$

obtained by applying the functor $\mathbf{Cat}^{\vec{I}} \to \mathbf{Pos}_\bullet$ from Equation 1 to the slice projection functor

$$\mathrm{dom}\colon\ C/_x \to C.$$

Let us unravel the definition of $\pi_0(C/_x)$ to a more explicit form. We start from the projection functor $\mathrm{dom}\colon C/_x \to C$. To this we may either apply Q or $\|-\|^{\vec{I}}$. Since quotients in **Pos** are simpler to compute than quotients in **Cat**, we apply poset reflection first, which gives us an order-preserving map

$$\|\mathrm{dom}\|\colon\ \|C/_x\| \to \|C\|.$$

Unravelling the explicit definition of poset reflection for $C/_x$, we see that:

- an element of $\|C/_x\|$ is an equivalence class $\|f\colon y \to x\|$ of morphisms of C with codomain x, where $\|f\| = \|g\|$ if and only if f factors through g and g factors through f, and
- $\|f\| \leq \|g\|$ if and only if f factors through g.

The map $\|\mathrm{dom}\|$ sends $\|f\|$ to $\|\mathrm{dom}\, f\|$. The image of $\|\mathrm{dom}\|$ is then the set

$$\{\|y\| \mid \text{there exists a morphism } f\colon y \to x \text{ in } C\},$$

which is, equivalently, the lower set of $\|x\|$ in $\|C\|$.

Applying $\mathsf{Q}\colon \mathbf{Pos}^{\vec{I}} \to \mathbf{Pos}_\bullet$ to this map produces the quotient of $\|C\|$ with all elements of this set identified, pointed with the element resulting from their identification, which we denote by $[x]$. Hence, an element of $\pi_0(C/_x)$ is either $[x]$, or it is $\|y\|$ for some object y such that there exists no morphism $f\colon y \to x$ in C. The order relation is defined as follows, by case distinction:

- $[x] \leq [x]$ trivially;
- $[x] \leq \|y\|$ if and only if there exists a span $(x \xleftarrow{f} z \xrightarrow{g} y)$ in C;
- it is never the case that $\|y\| \leq [x]$;
- $\|y\| \leq \|z\|$ if and only if there exists a morphism $f\colon y \to z$ in C.

Notice that $[x]$ is always minimal in $\pi_0(C/_x)$.

The partial order on $\pi_0(C/_x)$ ranks obstructions to weak terminality by "size": if we removed an obstruction $\|y\|$, adding a morphism $y \to x$, we would also have to remove all the "smaller" obstructions $\|z\| \leq \|y\|$. The minimal element $[x]$ represents the "non-obstructions":

Proposition 5. *Let C be a category and x an object in C. The following are equivalent:*

(a) $\pi_0(C/_x) = \{[x]\}$*;*

(b) x is a weak terminal object in C.

The notation and terminology is suggestive of the π_0 of a pointed topological space or groupoid, that is, its set of connected components, pointed with the connected component of the basepoint. The following result shows that, indeed, the notions coincide when C happens to be a groupoid.

Proposition 6 ($\pi_0(G/_x)$ for a groupoid)**.** *Let G be a groupoid and x an object in G. Then*

1. *$\pi_0(G/_x)$ is a "set", that is, a discrete poset, and*
2. *as a pointed set, it is isomorphic to the set $\pi_0(G)$ of connected components of G, pointed with the connected component of x.*

Now, we investigate obstructions to *subterminality*. Our main strategy will be to recast subterminality in a way that allows us to leverage Definition 7. We know that an object $\mathbb{1}$ fails to be subterminal when, for an object x, the arrow $x \to \mathbb{1}$ is not unique. As such, we will describe obstructions to subterminality as pairs of parallel, unequal arrows.

Definition 8 (Category of parallel arrows over an object)**.** Let C be a category and x an object in C. The *category of parallel arrows in C over x* is the category $\mathrm{Par}(C/_x)$ where:

- Objects are pairs of morphisms $(f_0, f_1 \colon y \to x)$ with codomain x.
- A morphism from $(f_0, f_1 \colon y \to x)$ to $(g_0, g_1 \colon z \to x)$ is a morphism $h \colon y \to z$ such that $f_0 = h \mathbin{⨾} g_0$ and $f_1 = h \mathbin{⨾} g_1$.

This comes with a projection functor $\mathrm{dom} \colon \mathrm{Par}(C/_x) \to C$ sending a parallel pair to its domain.

Proposition 7. *Let C be a category and $\mathbb{1}$ an object in C. The following are equivalent:*

(a) *$\mathbb{1}$ is subterminal in C;*

(b) *$(\mathrm{id}_{\mathbb{1}}, \mathrm{id}_{\mathbb{1}})$ is a terminal object in $\mathrm{Par}(C/_{\mathbb{1}})$;*

(c) *$(\mathrm{id}_{\mathbb{1}}, \mathrm{id}_{\mathbb{1}})$ is a weak terminal object in $\mathrm{Par}(C/_{\mathbb{1}})$.*

Proposition 7 allows us to reduce the study of obstructions to subterminality of an object $\mathbb{1}$ in C to the study of obstructions to weak terminality of $(\mathrm{id}_{\mathbb{1}}, \mathrm{id}_{\mathbb{1}})$ in $\mathrm{Par}(C/_{\mathbb{1}})$.

Definition 9 (First homotopy poset)**.** Let C be a category and x an object in C. The *first homotopy poset of C over x* is the pointed poset

$$(\pi_1(C/_x), [x]) := \Big(\pi_0(\mathrm{Par}(C/_x)/_{(\mathrm{id}_x, \mathrm{id}_x)}), [(\mathrm{id}_x, \mathrm{id}_x)]\Big).$$

Putting together the description of the 0th homotopy poset, the definition of $\mathrm{Par}(C/_x)$ in Definition 8, and Proposition 7, we see that an element of $\pi_1(C/_x)$ is either $[x]$, or $\|(f,g)\|$ for some parallel pair of morphisms $f, g \colon y \to x$ in C with $f \neq g$. The order relation is defined as follows:

- $[x] \leq [x]$ trivially;
- $[x] \leq \|(f, g \colon y \to x)\|$ if and only if there exists a morphism $h \colon z \to y$ in C equalising (f, g), that is, satisfying $h \mathbin{⨾} f = h \mathbin{⨾} g$;
- it is never the case that $\|(f,g)\| \leq [x]$;
- $\|(f, g \colon y \to x)\| \leq \|(f', g' \colon y' \to x)\|$ if and only if there exists a morphism $h \colon y \to y'$ such that $f = h \mathbin{⨾} f'$ and $g = h \mathbin{⨾} g'$ in C.

Proposition 8. *Let C be a category and x an object in C. The following are equivalent:*

(a) *$\pi_1(C/_x) = \{[x]\}$;*

(b) *x is subterminal in C.*

Corollary 1. *Let C be a category and x an object in C. The following are equivalent:*

(a) $\pi_0(C/_x) = \{[x]\}$ *and* $\pi_1(C/_x) = \{[x]\}$,

(b) *x is a terminal object in C.*

Remark 1. Recall that the (underlying set of the) fundamental group of a pointed topological space (X,x) is defined by

$$\pi_1(X,x) := \pi_0(\Omega(X,x), c_x)$$

where $\Omega(X,x)$ is the space of loops in X based at x, and c_x is the constant path at x. For a pointed groupoid, which may be seen as the fundamental groupoid of a pointed space, this reduces to the set of automorphisms of the object x, pointed with the identity automorphism.

The definition of $\pi_1(C/_x)$ is made in analogy with this, letting the category of parallel arrows over x replace the space of loops based at x, and a pair of identity morphisms replace the constant path. The following result proves that, just like the zeroth homotopy poset, the first homotopy poset is a generalisation of its groupoidal analogue.

Proposition 9 ($\pi_1(G/_x)$ for a groupoid)**.** *Let G be a groupoid and x an object in G. Then:*

1. *$\pi_1(G/_x)$ is a "set", that is, a discrete poset, and*
2. *as a pointed set, it is isomorphic to the underlying pointed set of the group $\pi_1(G,x) = \mathrm{Hom}_G(x,x)$.*

Remark 2. We mention here that the field of *directed algebraic topology* [10, 5] has also produced "non-invertible" versions of π_1, namely, the fundamental *category* and *monoids*, that apply to directed spaces. If applied to a category, these pick out "tautologically" the category itself and its monoids of endomorphisms. To our knowledge, there is no strong relation to our line of research.

To conclude this section, we show in what way the homotopy posets are functorial in the pair (C,x) of a category and an object.

Proposition 10 (Functoriality of the homotopy posets)**.** *Let C be a category, $i \in \{0,1\}$. Then:*

1. *the assignment $x \mapsto \pi_i(C/_x)$ extends to a functor $\pi_i(C/\_)\colon C \to \mathbf{Pos}_\bullet$;*
2. *a functor $\mathsf{F}\colon C \to D$ induces a natural transformation $\pi_i(\mathsf{F})\colon \pi_i(C/\_) \Rightarrow \pi_i(D/_{\mathsf{F}\_})$.*

Given another functor $\mathsf{G}\colon D \to E$, this assignment satisfies

$$\pi_i(\mathsf{F} \mathbin{⨟} \mathsf{G}) = \pi_i(\mathsf{F}) \mathbin{⨟} \pi_i(\mathsf{G}), \qquad \pi_i(\mathrm{id}_C) = \mathrm{id}_{\pi_i(C/\_)}.$$

A concise way of packaging this information is to say that π_i defines a functor from **Cat** to the *lax slice* $\mathscr{C}at \upharpoonright \mathbf{Pos}_\bullet$, where $\mathscr{C}at$ is the "huge" category of possibly large categories. The objects of the lax slice are pairs of a possibly large category $\mathbf{C}$ and a functor $\mathbf{C} \to \mathbf{Pos}_\bullet$, and the morphisms are triangles of functors commuting up to a natural transformation. Indeed, given $\mathsf{F}\colon C \to D$, we have a triangle

$$\begin{tikzcd} C \arrow[r, "\pi_i(C/\_)"] \arrow[d, "\mathsf{F}"'] \arrow[dr, phantom, "\swarrow \pi_i(\mathsf{F})"] & \mathbf{Pos}_\bullet \\ D \arrow[ur, "\pi_i(D/\_)"'] & \end{tikzcd}$$

commuting up to the natural transformation $\pi_i(\mathsf{F})$.

Remark 3 (Dual invariants)**.** As usual, all the constructions can be dualised to C^{op}. This will replace the slice over an object and its domain opfibration with the slice under an object and its codomain fibration, producing invariants classifying obstructions to *initiality* of the object.

2 Obstructions to a morphism being iso

As remarked in the Introduction, one of our main motivations for introducing homotopy posets was measuring how far a generic morphism is from being iso. Just as we could separate obstructions to terminality into obstructions to weak terminality and subterminality, we can separate obstructions to a morphism being iso into obstructions to a morphism being split epi and mono, respectively.

Proposition 11. *Let $f\colon X \to Y$ be a morphism in a category C. Then:*

- *f is split epi in C if and only if f is weak terminal in $C/_Y$,*
- *f is mono in C if and only if f is subterminal in $C/_Y$.*

Corollary 2. *Let $f : X \to Y$ be a morphism in a category C. Then:*

- *f is split epi if and only if $\pi_0((C/_Y)/_f)$ is trivial;*
- *f is mono if and only if $\pi_1((C/_Y)/_f)$ is trivial, and:*
- *f is iso if and only if both $\pi_0((C/_Y)/_f)$ and $\pi_1((C/_Y)/_f)$ are trivial.*

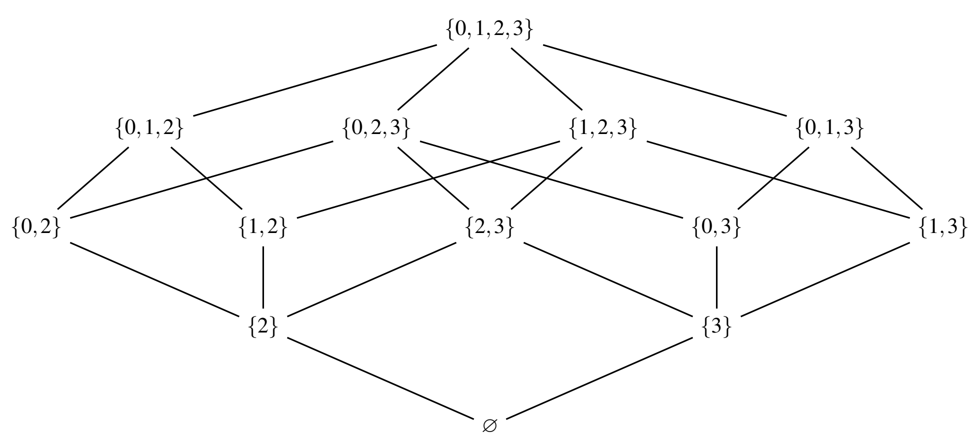
Furthermore, when the homotopy posets associated to a morphism f are not trivial, they give us precise information about why f fails to be split epi and mono.

To make this more concrete, let us spell out precisely how to compute the invariants associated to a function between sets, where split epi (assuming choice) means *surjective* and mono means *injective*. This amounts to calculating $\pi_0((\mathbf{Set}/_Y)/_f)$ and $\pi_1((\mathbf{Set}/_Y)/_f)$ for some function $f\colon X \to Y$.

Proposition 12. *Let $f\colon X \to Y$ be a function between sets. $\|\mathbf{Set}/_Y\|$ is isomorphic, as a poset, to the power set $\mathscr{P}Y$, via the assignment $(S \subseteq Y) \mapsto \|\imath_S\|$, where $\imath_S$ is the injective function including S into Y. Through this bijection, $\|f\|$ corresponds to the image $f(X)$ of f.*

Using this correspondence and quotienting by the lower set of $f(X)$, which contains in particular $\varnothing$, we may identify $\pi_0((\mathbf{Set}/_Y)/_f)$ with the subposet of $\mathscr{P}Y$ whose elements are either $\varnothing$ or subsets of Y that contain at least one element $y \notin f(X)$. The "minimal obstructions", that is, the minimal elements in the complement of the basepoint, are the singletons $\{y\}$ with $y \in Y \setminus f(X)$. This poset is trivial if and only if $f(X) = Y$, that is, iff f is surjective.

Example 1. Let $f\colon \{0,1\} \to \{0,1,2,3\}$ be the function mapping $0 \mapsto 0$ and $1 \mapsto 1$. The homotopy poset $\pi_0((\mathbf{Set}/_{\{0,1,2,3\}})/_f)$ has the following structure:



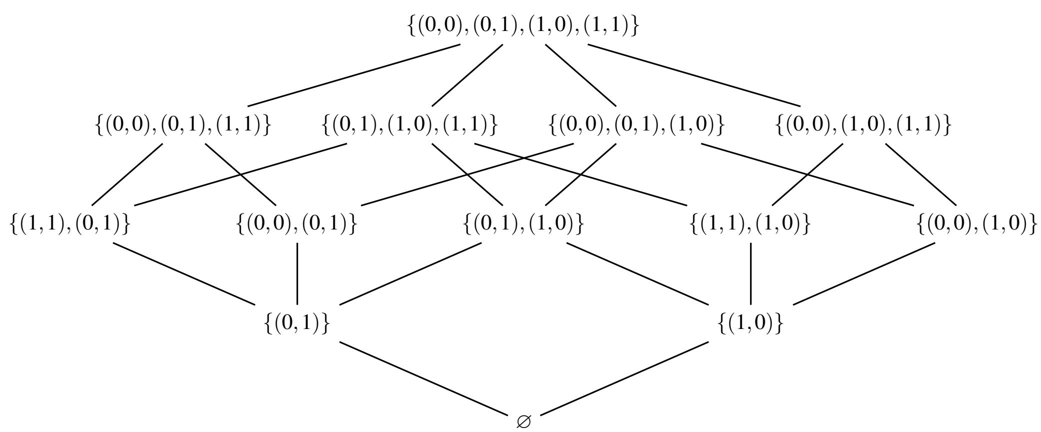
The minimal obstructions $\{2\}$ and $\{3\}$ are in bijection with the elements not in the image of f.

Proposition 13. *Let $X \times_f X$ be the pullback of f along itself — that is, the set $\{(x_0,x_1) \mid f(x_0) = f(x_1)\}$ — and let $p_f \colon X \times_f X \to Y$ be the function $(x_0,x_1) \mapsto f(x_0) = f(x_1)$. Then:*

1. *$\|\mathrm{Par}((\mathbf{Set}/_Y)/_f)\|$ is isomorphic to $\mathscr{P}(X \times_f X)$ via the assignment $(S \subseteq X \times_f X) \mapsto \|(p_0|_S, p_1|_S)\|$, where $p_i|_S$ are the projections $X \times_f X \to Y$, restricted to S, seen as morphisms $p_f|_S \to f$ in $\|\mathrm{Par}((\mathbf{Set}/_Y)/_f)\|$;*
2. *through this bijection, $\|(\mathrm{id}_f, \mathrm{id}_f)\|$ is identified with the diagonal ΔX.*

Using this correspondence, we may identify $\pi_1(\mathbf{Set}/_X)$ with the subposet of $\mathscr{P}(X \times_f X)$ whose elements are either $\varnothing$, or contain at least one pair (x_0,x_1) such that $x_0 \neq x_1$. This poset is trivial if and only if f is injective. Notice that the minimal obstructions to injectiveness of f are in bijection with pairs (x_0,x_1) where $x_0 \neq x_1$ but $f(x_0) = f(x_1)$.

Example 2. Let $f : \{0,1\} \to \{*\}$ be the function mapping $0 \mapsto *$, $1 \mapsto *$. Then $\{0,1\} \times_f \{0,1\}$ is the set {(0,0),(0,1),(1,0),(1,1)}, and $\pi_1((\mathbf{Set}/_{\{*\}})/_f)$ has the following structure:



Notice that, via the isomorphism $\mathbf{Set} \simeq \mathbf{Set}/_{\{*\}}$, this is isomorphic to $\pi_1(\mathbf{Set}/_{\{0,1\}})$.

To conclude, suppose that two morphisms are both components of the same natural transformation. Is there a relation between the associated invariants? The following result answers this question in the affirmative.

Proposition 14 (Covariance over the domain of a natural transformation)**.** *Let $\mathsf{F}, \mathsf{G} \colon C \to D$ be two functors and let $\alpha \colon \mathsf{F} \Rightarrow \mathsf{G}$ be a natural transformation. For all $i \in \{0,1\}$, the assignment*

$$x \mapsto \pi_i((D/_{\mathsf{G}x})/_{\alpha_x})$$

extends to a functor $C \to \mathbf{Pos}_\bullet$.

Notice that this is *not* simply a consequence of Proposition 10, that is, it does not arise from the general functoriality result by pre-composition with another functor.[5] It implies that we can naturally map obstructions for α_x to obstructions for α_y along a morphism $f \colon x \to y$ in C; we can think of morphisms in C as inducing a "flow" of obstructions to the components of α, under which a non-trivial obstruction may be trivialised, but it can never be the case that a non-obstruction is "un-trivialised".

[5]There is a unifying perspective on the two functoriality results, involving the theory of fibrations and cofibrations of categories; this will be discussed in an extended technical paper.

3 Qualifying compositionality

Now let $\mathsf{P}\colon C \to D$ be a *lax* functor of *bicategories*. This means that, for all triples of objects X,Y,Z in C, we have two functors

$$(\mathsf{P}-)\,\mathbin{;}\,(\mathsf{P}-),\ \mathsf{P}(-\mathbin{;}-)\colon \mathrm{Hom}_C(X,Y)\times\mathrm{Hom}_C(Y,Z)\to\mathrm{Hom}_D(\mathsf{P}X,\mathsf{P}Z)$$

connected by a natural transformation, the *laxator* $\varphi\colon (\mathsf{P}-)\mathbin{;}(\mathsf{P}-)\Rightarrow \mathsf{P}(-\mathbin{;}-)$.[6] As a special case, when C and D are monoidal categories seen as one-object bicategories, P is a lax monoidal functor, and the laxator is a natural transformation $(\mathsf{P}-)\otimes(\mathsf{P}-)\Rightarrow\mathsf{P}(-\otimes-)$.

By Proposition 14, we obtain functors $\mathrm{Hom}_C(X,Y)\times\mathrm{Hom}_C(Y,Z)\to\mathbf{Pos}_\bullet$ sending a pair of morphisms $(f\colon X\to Y, g\colon Y\to Z)$ to the homotopy posets

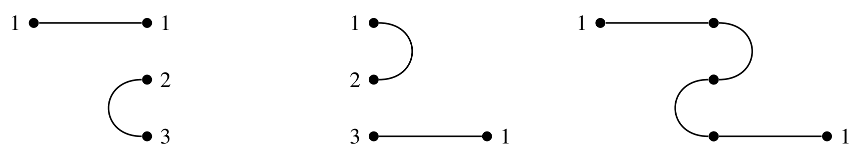
$$\pi_i((\mathrm{Hom}_D(\mathsf{P}X,\mathsf{P}Z)/\mathsf{P}(f\mathbin{;}g))/\varphi_{f,g})$$

associated to the component $\varphi_{f,g}$ of the laxator.

In the scenario sketched in the Introduction, the failure of $\varphi_{f,g}$ to be iso is a failure of the "semantic" functor P to be "fully compositional" with respect to the composition $f\mathbin{;}g$. Thus the elements of these homotopy posets may be seen as local *obstructions to compositionality* of P. Most interestingly, these obstructions are covariant with respect to the 2-morphisms of C; thus we can think of "modifying f and g" by acting on them with a 2-morphism, and see how that affects the obstructions.

3.1 Open Graphs

We apply our framework to a couple of tangible examples. Open graphs, defined in [6], can be thought of as *graphs with interfaces*. Formally, open graphs are (isomorphism classes of) decorated cospans with decorations in the category **Graph** of graphs and homomorphisms. Intuitively, they are depicted as in the examples below, with *input* vertices on the left and *output* vertices on the right:



Indeed, there is a bicategory **OpenGraph** that has sets as objects, open graphs as morphisms, and interface-preserving graph homomorphisms as 2-morphisms. For instance, the first and second open graphs above correspond to morphisms $G\colon\{1\}\to\{1,2,3\}$ and $H\colon\{1,2,3\}\to\{1\}$. These morphisms can be composed, resulting in the morphism $G\mathbin{;}H\colon\{1\}\to\{1\}$ corresponding to the third open graph in the picture above.

Every graph can be mapped to its *reachability relation*[7]: this is a relation on the vertexes of the graph, where two vertexes are considered related iff there is a path between them. Reachability can be recast as a lax functor **OpenGraph** $\to$ **Rel** to the bicategory of sets, relations, and inclusions of relations, which maps an open graph $G\colon X\to Y$ to the relation $\mathsf{R}G\colon X\to Y$ defined by

$\mathsf{R}G(x,y)$ if and only if there is a path between the input vertex x and the output vertex y.

[6]Technically, the laxators are a family of natural transformations indexed by X,Y,Z, but we will leave the indexing implicit.
[7]Cfr. [12], for the similar example of open causal models and causal influence.

Because **Rel** is locally posetal, to define R on 2-morphisms it suffices to verify that, if $f\colon G \to G'$ is a graph homomorphism, then $\mathsf{R}G \subseteq \mathsf{R}G'$. The laxators are also uniquely defined.

We can see that this functor is not strong. In the example above we have that $\mathsf{R}G \subseteq \{1\} \times \{1,2,3\}$ only contains the pair $(1,1)$, since there are no paths from 1 to 2 and from 1 to 3 in G. Similarly, $\mathsf{R}H \subseteq \{1,2,3\} \times \{1\}$ only contains the pair $(3,1)$. It follows that $\mathsf{R}G \mathbin{⨾} \mathsf{R}H\colon \{1\} \to \{1\}$ is the empty relation, but $\mathsf{R}(G \mathbin{⨾} H)\colon \{1\} \to \{1\}$ is total, so $\mathsf{R}G \mathbin{⨾} \mathsf{R}H \subsetneq \mathsf{R}(G \mathbin{⨾} H)$.

The result is that, if we want to compute the reachability relation of $G \mathbin{⨾} H$ by looking at the reachability relations of G and H separately, we are going to miss something. This "compositionality gap" is tracked by the π_0 associated to the laxator components $\varphi_{G,H}\colon \mathsf{R}G \mathbin{⨾} \mathsf{R}H \subseteq \mathsf{R}(G \mathbin{⨾} H)$ (because these are all injective, the π_1 will always be trivial).

In our example, $\pi_0((\mathrm{Hom}_{\mathbf{Rel}}(\{1\},\{1\})/\mathsf{R}(G \mathbin{⨾} H))/\varphi_{G,H})$ is isomorphic to the poset $(\varnothing < \{(1,1)\})$ pointed with $\varnothing$, so there is exactly one non-trivial obstruction. Using covariance, we can think of "removing the obstruction" by modifying one or both of the parts G or H with a 2-morphism, that is, with a graph homomorphism. For example, we can act on G with the homomorphism which identifies the output vertices 1 and 3. The resulting graph G' has $\mathsf{R}G' = \{(1,1),(1,3)\}$, so $\mathsf{R}G' \mathbin{⨾} \mathsf{R}H = \mathsf{R}(G' \mathbin{⨾} H) = \{(1,1)\}$; correspondingly, we obtain a map of pointed posets from the π_0 associated to $\varphi_{G,H}$ to the π_0 associated to $\varphi_{G',H}$, which "trivialises all obstructions".

3.2 Schrödinger Compositionality

The name *Schrödinger compositionality* was introduced in [3] to refer to the form of compositionality that exists in quantum mechanics, where *non-separable states* are present, to disambiguate it from others. [8] In the following, we will focus on the special case of a state that can be "more than its parts". This is arguably what makes composition interesting in quantum mechanics: it makes entanglement possible, which Schrödinger described as "the characteristic trait of quantum mechanics" [14]. In contrast with the example of open graphs, where the "compositionality gap" represents an obstacle to a computation strategy, here it can be seen as a positive feature. Our approach can be used in both contexts; we will focus on the case study of non-separable states, recasting it as the failure of a lax functor to be strong.

In the context of monoidal categories, a *state* is a morphism $I \to A$, where I is the monoidal unit. We say that a state $\psi\colon I \to A \otimes B$ is *separable* if there exist states $\psi_A\colon I \to A$ and $\psi_B\colon I \to B$ such that $\psi = \psi_A \otimes \psi_B$.

Definition 10. Let $(C,\otimes,I)$ be a monoidal category. The *state functor* of C is the representable functor $\mathrm{Hom}_C(I,-)\colon C \to \mathbf{Set}$.

Proposition 15 (Laxity of the state functor)**.** *The state functor lifts to a lax monoidal functor from* $(C,\otimes,I)$ *to* $(\mathbf{Set},\times,\{*\})$*, with laxator components*

$$\begin{aligned} \varphi_{A,B}\colon \mathrm{Hom}_C(I,A) \times \mathrm{Hom}_C(I,B) &\to \mathrm{Hom}_C(I,A \otimes B) \\ (\psi_A,\psi_B) &\mapsto \psi_A \otimes \psi_B. \end{aligned}$$

Recall that a monoidal category is *semicartesian* if its monoidal unit is terminal. The following result is a consequence of the general fact that a functor from a semicartesian to a cartesian monoidal category has a canonical oplax monoidal structure.

[8] For the purposes of this work, we are leaving out of the present analysis the aspects of Schrödinger compositionality regarding the "ontological interpretation", originally presented in [3].

Proposition 16 (Oplaxity of the state functor)**.** *Let* $(C, \otimes, \mathbf{1})$ *be a semicartesian category. Then the state functor lifts to an oplax monoidal functor from* $(C, \otimes, \mathbf{1})$ *to* $(\mathbf{Set}, \times, \{*\})$.

Clearly, there are cases where the state functor is not just lax or oplax, but strong. The following result captures the well-known fact that in a cartesian monoidal category every state is separable.

Proposition 17 (Strongness of the state functor)**.** *If* $(C, \times, \mathbf{1})$ *is cartesian, then the state functor is strong monoidal.*

Having turned Schrödinger compositionality into a question about (op)laxity of a functor, we can put our framework to good work. By Proposition 14, we have functors $C \times C \to \mathbf{Pos}_\bullet$ sending pairs of objects (A, B) of C to the homotopy posets

$$\pi_i((\mathbf{Set}/\mathrm{Hom}_C(I, A \otimes B))/\varphi_{A,B}), \quad i \in \{0, 1\}. \tag{2}$$

Using the description of homotopy posets for slices of **Set** from Section 2, we see that

- minimal obstructions in π_0 are in bijection with non-separable states of $A \otimes B$,
- minimal obstructions in π_1 are in bijection with pairs of pairs of states $((\psi_A, \psi_B), (\chi_A, \chi_B))$ such that $\psi_A \otimes \psi_B = \chi_A \otimes \chi_B$.

For example, in $(\mathbf{Vect}_\mathbb{C}, \otimes, \mathbb{C})$, the monoidal category of complex vector spaces with their tensor product, whenever A and B are at least 2-dimensional, we have instances of both:

- the state $1 \mapsto \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}$ of $\mathbb{C}^2 \otimes \mathbb{C}^2$ is non-separable,
- given any pair of states (ψ_A, ψ_B) and any non-zero $\lambda \in \mathbb{C}$, the pair $(\chi_A, \chi_B) := (\lambda \psi_A, \lambda^{-1} \psi_B)$ satisfies $\psi_A \otimes \psi_B = \chi_A \otimes \chi_B$.

We can derive a few simple, immediate consequences from the covariance of (2) in the pair (A, B).

1. Given morphisms $f \colon A \to A'$, $g \colon B \to B'$, the induced maps of posets preserve the basepoint, that is, map "non-obstructions" to "non-obstructions". In this case, this implies that *it is not possible to entangle a separable state by local actions*, that is, by applying morphisms on A and B separately.
2. On the other hand, it is, in principle, possible for the induced maps to send non-trivial obstructions to the basepoint. For example, in complex vector spaces, acting on A or B with a rank-1 linear map always has a separating effect.

Conclusion

We have introduced our new invariants of categories and stated their fundamental properties, before sketching, through a couple of simple examples, how they may be used to obtain a more fine-grained analysis of "failures of compositionality" than a simple yes-or-no judgement. In an extended technical paper, we will study their formal aspects more in depth, including criteria for the existence of joins and meets, induced monoidal structures, and finer aspects of functoriality.

Most importantly, we hope to have opened a new avenue in "formal compositionality theory". The greatest challenge will be to graduate from proof-of-concept examples to ones that reveal more interesting structure, perhaps in non-**Set**-like categories where a split epi or mono is not simply a surjective or injective map. We have been looking at case studies of this sort, which nevertheless have manageable combinatorics permitting an exhaustive study of their homotopy posets, and we hope to discuss them in future work.

References

[1] Samson Abramsky & Bob Coecke (2009): *Categorical quantum mechanics*. *Handbook of quantum logic and quantum structures* 2, pp. 261–325, doi:10.1016/B978-0-444-52869-8.50010-4.

[2] John C Baez, Fabrizio Genovese, Jade Master & Michael Shulman (2021): *Categories of nets*. In: *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, pp. 1–13, doi:10.1109/LICS52264.2021.9470566.

[3] Bob Coecke (2021): *Compositionality as we see it, everywhere around us*. *arXiv preprint arXiv:2110.05327*. arXiv:2110.05327.

[4] Bob Coecke (2021): *The mathematics of text structure*. *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*, pp. 181–217.

[5] Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, Samuel Mimram & Martin Raussen (2016): *Directed algebraic topology and concurrency*. 138, Springer, doi:10.1007/978-3-319-15398-8.

[6] Brendan Fong (2015): *Decorated Cospans*. *arXiv preprint arXiv:1502.00872* 30(33), pp. 1096–1120. arXiv:1502.00872.

[7] Brendan Fong & David I. Spivak (2019): *An invitation to applied category theory: seven sketches in compositionality*. Cambridge University Press, Cambridge; New York, NY, doi:10.1017/9781108668804.

[8] Fabrizio Romano Genovese (2018): *Modularity vs compositionality: a history of misunderstandings*. Online article, https://blog.statebox.org/modularity-vs-compositionality-a-history-of-misunderstandings-be0150033568. Accessed 17 July 2023.

[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] Marco Grandis (2009): *Directed algebraic topology: models of non-reversible worlds*. 13, Cambridge University Press, doi:10.1017/CBO9780511657474.

[11] Wolfram Hinzen, Edouard Machery & Markus Werning, editors (2012): *The Oxford Handbook of Compositionality*. Oxford Handbooks in Linguistics, Oxford University Press, Oxford; New York, NY.

[12] Robin Lorenz & Sean Tull (2023): *Causal models in string diagrams*. *arXiv preprint arXiv:2304.07638*. arXiv:2304.07638.

[13] Caterina Puca, Amar Hadzihasanovic, Fabrizio Genovese & Bob Coecke (2023): *Obstructions to Compositionality*. *arXiv preprint arXiv:2307.14461v1*. arXiv:2307.14461v1.

[14] Erwin Schrödinger (1935): *Discussion of Probability Relations between Separated Systems*. *Mathematical Proceedings of the Cambridge Philosophical Society* 31(4), p. 555–563, doi:10.1017/S0305004100013554.

Appendix

Proof of Proposition 10

Proving Proposition 10 requires to build a hefty amount of theory, which is why we reserve the Appendix for this.

Definition 11 (Past extension)**.** Let A be a category. A *past extension of* A is a functor $\iota\colon A \hookrightarrow B$ with the following property: there exists a functor $\chi_A\colon B \to \vec{I}$ such that

$$\begin{tikzcd} A \arrow[r, "!"] \arrow[d, hook, "\iota"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathbf{1} \arrow[d, hook, "1"] \\ B \arrow[r, "\chi_A"] & \vec{I} \end{tikzcd} \tag{3}$$

is a pullback in **Cat**.

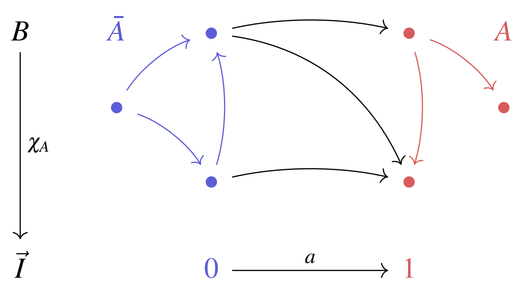
Remark 4. The following is an equivalent characterisation of past extensions: there exist a category $\bar{A}$ and a profunctor $\mathsf{H}\colon \bar{A}^{\mathrm{op}} \times A \to \mathbf{Set}$ such that

1. B is isomorphic to the *collage*, also known as *cograph*, of H, and
2. ι is, up to isomorphism, the inclusion of A into the collage.

A technical name for a functor satisfying the condition on ι is *codiscrete coopfibration*; it is one leg of a two-sided codiscrete cofibration of categories.

The idea is that ι embeds A into a larger category, whose objects outside of the image of A only have morphisms pointing *towards* A, hence are "in the past" of A if we interpret the direction of morphisms as a time direction. Notice that the fact that (3) is a pullback implies that ι is injective on objects and morphisms, using their representation as functors from $\mathbf{1}$ and $\vec{I}$, respectively.

The following picture illustrates the bipartition of B induced by χ_A, with the fibre $\bar{A}$ of 0 "in the past" of the fibre A of 1:



Definition 12 (Category of past extensions)**.** Let A be a category. The *category of past extensions of* A is the large category $\mathbf{Past}(A)$ whose

- objects are past extensions $\iota\colon A \hookrightarrow B$, and
- a morphism from $(\iota\colon A \hookrightarrow B)$ to $(j\colon A \hookrightarrow B')$ is a factorisation of j through ι, that is, a functor $\mathsf{K}\colon B \to B'$ such that $j = \iota \mathbin{⨟} \mathsf{K}$.

Proposition 18 (The indexed category of past extensions of functors)**.** *Let A and C be categories. Then there exists a functor*

$$\mathsf{Ext}_C^A\colon \mathbf{Past}(A)^{\mathrm{op}} \times C^A \to \mathbf{Cat}$$

whose object part is defined as follows: given a past extension $\iota\colon A \hookrightarrow B$ *and a functor* $\mathsf{F}\colon A \to C$*, the category* $\mathsf{Ext}_C^A(\iota, \mathsf{F})$ *is the subcategory of* C^B *whose*

- *objects are (strict) extensions of* F *along* ι*, that is, functors* $\tilde{\mathsf{F}}\colon B \to C$ *such that*

$$\begin{tikzcd} A \arrow[r, "\mathsf{F}"] \arrow[d, hook, "\iota"] & C \\ B \arrow[ur, "\tilde{\mathsf{F}}"'] & \end{tikzcd}$$

  strictly commutes, and

- *morphisms from* $\tilde{\mathsf{F}}_1$ *to* $\tilde{\mathsf{F}}_2$ *are natural transformations* $\tau\colon \tilde{\mathsf{F}}_1 \Rightarrow \tilde{\mathsf{F}}_2$ *that restrict along* ι *to the identity natural transformation on* F*.*

Proof. Given a morphism $\mathsf{K}\colon (\iota\colon A \hookrightarrow B) \to (j\colon A \hookrightarrow B')$ in $\mathbf{Past}(A)$,

$$\mathsf{K}^* := \mathsf{Ext}_C^A(\mathsf{K}, \mathsf{F})\colon \mathsf{Ext}_C^A(j, \mathsf{F}) \to \mathsf{Ext}_C^A(\iota, \mathsf{F})$$

is the functor that acts by precomposition, sending

- $\tilde{\mathsf{F}}\colon B' \to C$ to $\mathsf{K} \mathbin{⨟} \tilde{\mathsf{F}}\colon B \to C$, and
- $\tau\colon \tilde{\mathsf{F}}_1 \Rightarrow \tilde{\mathsf{F}}_2$ to $\mathsf{K} \mathbin{⨟} \tau\colon \mathsf{K} \mathbin{⨟} \tilde{\mathsf{F}}_1 \Rightarrow \mathsf{K} \mathbin{⨟} \tilde{\mathsf{F}}_2$.

This is well-defined as

$$\iota \mathbin{⨟} \mathsf{K} \mathbin{⨟} \tilde{\mathsf{F}} = j \mathbin{⨟} \tilde{\mathsf{F}} = \mathsf{F}, \qquad \iota \mathbin{⨟} \mathsf{K} \mathbin{⨟} \tau = j \mathbin{⨟} \tau = \mathrm{id}_{\mathsf{F}}.$$

Moreover, it is straightforward to check that

$$(\mathrm{id}_\iota)^* = \mathrm{id}_{\mathsf{Ext}_C^A(\iota,\mathsf{F})}, \qquad (\mathsf{K} \mathbin{⨟} \mathsf{L})^* = \mathsf{L}^* \mathbin{⨟} \mathsf{K}^*$$

for any composable pair K, L of morphisms in $\mathbf{Past}(A)$.

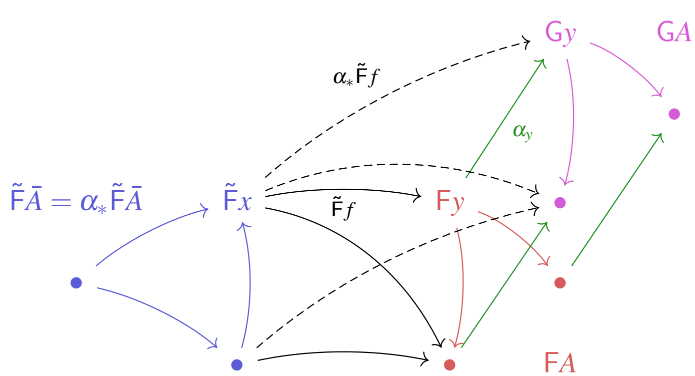
Given a natural transformation $\alpha\colon \mathsf{F} \Rightarrow \mathsf{G}$ between functors $\mathsf{F}, \mathsf{G}\colon A \to C$, the functor

$$\alpha_* := \mathsf{Ext}_C^A(\iota, \alpha)\colon \mathsf{Ext}_C^A(\iota, \mathsf{F}) \to \mathsf{Ext}_C^A(\iota, \mathsf{G})$$

is defined as follows. Given an object $\tilde{\mathsf{F}}\colon B \to C$ of $\mathsf{Ext}_C^A(\iota, \mathsf{F})$, the functor $\alpha_*\tilde{\mathsf{F}}\colon B \to C$ is defined, on each morphism $f\colon x \to y$ in B, by

$$\alpha_*\tilde{\mathsf{F}}(f) := \begin{cases} \mathsf{G}(f') & \text{if } \chi_A(f) = 1 \text{ and } f = \iota(f'), \\ \tilde{\mathsf{F}}(f) \mathbin{⨟} \alpha_{y'} & \text{if } \chi_A(f) = a \text{ and } y = \iota(y'), \\ \tilde{\mathsf{F}}(f) & \text{if } \chi_A(f) = 0. \end{cases}$$

By construction $\iota \mathbin{⨟} \alpha_*\tilde{\mathsf{F}} = \mathsf{G}$. The following picture illustrates the definition.



Let us show that $\alpha_*\tilde{\mathsf{F}}$ is well-defined as a functor.

1. Given an identity id_x in B, necessarily $\chi_A(\mathrm{id}_x) = 0$, in which case
$$\alpha_*\tilde{\mathsf{F}}(\mathrm{id}_x) = \tilde{\mathsf{F}}(\mathrm{id}_x) = \mathrm{id}_{\tilde{\mathsf{F}}(x)},$$
or $\chi_A(\mathrm{id}_x) = 1$, in which case
$$\alpha_*\tilde{\mathsf{F}}(\mathrm{id}_x) = \mathsf{G}(\mathrm{id}_{x'}) = \mathrm{id}_{\mathsf{G}(x')},$$
where x' is the unique lift of x to A. Thus $\alpha_*\tilde{\mathsf{F}}$ preserves identities.
2. Given a composable pair $f\colon x \to y$, $g\colon y \to z$, we have the following cases.
   - If $\chi_A(f) = \chi_A(g) = 1$, then $\chi_A(f \mathbin{;} g) = 1$, and
   $$\alpha_*\tilde{\mathsf{F}}(f) \mathbin{;} \alpha_*\tilde{\mathsf{F}}(g) = \mathsf{G}(f') \mathbin{;} \mathsf{G}(g') = \mathsf{G}(f' \mathbin{;} g') = \alpha_*\tilde{\mathsf{F}}(f \mathbin{;} g),$$
   where f', g' are the unique lifts of f, g to A.
   - If $\chi_A(f) = \chi_A(g) = 0$, then $\chi_A(f \mathbin{;} g) = 0$, and
   $$\alpha_*\tilde{\mathsf{F}}(f) \mathbin{;} \alpha_*\tilde{\mathsf{F}}(g) = \tilde{\mathsf{F}}(f) \mathbin{;} \tilde{\mathsf{F}}(g) = \tilde{\mathsf{F}}(f \mathbin{;} g) = \alpha_*\tilde{\mathsf{F}}(f \mathbin{;} g).$$
   - If $\chi_A(f) = 0$ and $\chi_A(g) = a$, then $\chi_A(f \mathbin{;} g) = a$, and
   $$\alpha_*\tilde{\mathsf{F}}(f) \mathbin{;} \alpha_*\tilde{\mathsf{F}}(g) = \tilde{\mathsf{F}}(f) \mathbin{;} \tilde{\mathsf{F}}(g) \mathbin{;} \alpha_{z'} = \tilde{\mathsf{F}}(f \mathbin{;} g) \mathbin{;} \alpha_{z'} = \alpha_*\tilde{\mathsf{F}}(f \mathbin{;} g),$$
   where z' is the unique lift of z to A.
   - If $\chi_A(f) = a$ and $\chi_A(g) = 1$, then $\chi_A(f \mathbin{;} g) = a$, and
   $$\alpha_*\tilde{\mathsf{F}}(f) \mathbin{;} \alpha_*\tilde{\mathsf{F}}(g) = \tilde{\mathsf{F}}(f) \mathbin{;} \alpha_{y'} \mathbin{;} \mathsf{G}(g') = \tilde{\mathsf{F}}(f) \mathbin{;} \mathsf{F}(g') \mathbin{;} \alpha_{z'},$$
   where $g'\colon y' \to z'$ is the unique lift of g to A, and we used naturality of α.
   Since $\mathsf{F}(g') = \tilde{\mathsf{F}}(\imath(g')) = \tilde{\mathsf{F}}(g)$, this is equal to
   $$\tilde{\mathsf{F}}(f) \mathbin{;} \tilde{\mathsf{F}}(g) \mathbin{;} \alpha_{z'} = \alpha_*\tilde{\mathsf{F}}(f \mathbin{;} g).$$

   No other cases are possible.

This proves that $\alpha_*\tilde{\mathsf{F}}$ is well-defined.

Given a morphism $\tau\colon \tilde{\mathsf{F}}_1 \Rightarrow \tilde{\mathsf{F}}_2$ of $\mathsf{Ext}^A_C(\imath, \mathsf{F})$, the natural transformation $\alpha_*\tau\colon \alpha_*\tilde{\mathsf{F}}_1 \Rightarrow \alpha_*\tilde{\mathsf{F}}_2$ is defined, on each object x in B, by

$$(\alpha_*\tau)_x := \begin{cases} \mathrm{id}_{\mathsf{G}(x')} & \text{if } \chi_A(x) = 1 \text{ and } x = \imath(x'), \\ \tau_x & \text{if } \chi_A(x) = 0. \end{cases}$$

To show that this is well-defined as a natural transformation, consider a morphism $f\colon x \to y$ in B.

- If $\chi_A(f) = 1$ and $f'\colon x' \to y'$ is the unique lift of f to A, then
$$\alpha_*\tilde{\mathsf{F}}_1(f) \mathbin{;} (\alpha_*\tau)_y = \mathsf{G}(f') \mathbin{;} \mathrm{id}_{\mathsf{G}(y')} = \mathrm{id}_{\mathsf{G}(x')} \mathbin{;} \mathsf{G}(f') = (\alpha_*\tau)_x \mathbin{;} \alpha_*\tilde{\mathsf{F}}_2(f).$$

- If $\chi_A(f) = a$ and y' is the unique lift of y to A, then

$$\alpha_* \tilde{\mathsf{F}}_1(f) \mathbin{⨾} (\alpha_* \tau)_y = \tilde{\mathsf{F}}_1(f) \mathbin{⨾} \alpha_{y'} \mathbin{⨾} \mathrm{id}_{\mathsf{G}(y')} = \tilde{\mathsf{F}}_1(f) \mathbin{⨾} \tau_y \mathbin{⨾} \alpha_{y'}$$

  since $\tau_y = \tau_{\iota(y')} = \mathrm{id}_{\mathsf{F}(y')}$. By naturality of τ, this is equal to

$$\tau_x \mathbin{⨾} \tilde{\mathsf{F}}_2(f) \mathbin{⨾} \alpha_{y'} = (\alpha_* \tau)_x \mathbin{⨾} \alpha_* \tilde{\mathsf{F}}_2(f).$$

- If $\chi_A(f) = 0$, then

$$\alpha_* \tilde{\mathsf{F}}_1(f) \mathbin{⨾} (\alpha_* \tau)_y = \tilde{\mathsf{F}}_1(f) \mathbin{⨾} \tau_y = \tau_x \mathbin{⨾} \tilde{\mathsf{F}}_2(f) = (\alpha_* \tau)_x \mathbin{⨾} \alpha_* \tilde{\mathsf{F}}_2(f).$$

This concludes the definition of α_*. It is straightforward to check that

$$(\mathrm{id}_{\mathsf{F}})_* = \mathrm{id}_{\mathsf{Ext}^A_C(\iota,\mathsf{F})}, \qquad (\alpha \mathbin{⨾} \beta)_* = \alpha_* \mathbin{⨾} \beta_*$$

for all pairs of natural transformations α, β composable as morphisms in C^A. Finally, one can verify that, for all morphisms $\mathsf{K}\colon \iota \to j$ in $\mathbf{Past}(A)$ and $\alpha\colon \mathsf{F} \to \mathsf{G}$ in C^A, the diagram of functors

$$\begin{tikzcd}
\mathsf{Ext}^A_C(j,\mathsf{F}) \arrow[r, "\mathsf{K}^*"] \arrow[d, "\alpha_*"] & \mathsf{Ext}^A_C(\iota,\mathsf{F}) \arrow[d, "\alpha_*"] \\
\mathsf{Ext}^A_C(j,\mathsf{G}) \arrow[r, "\mathsf{K}^*"] & \mathsf{Ext}^A_C(\iota,\mathsf{G})
\end{tikzcd}$$

commutes in **Cat**. Thus we can define $\mathsf{Ext}^A_C(\mathsf{K},\alpha)$ as either path in the commutative diagram, and conclude that Ext^A_C is well-defined as a functor. ■

Proposition 19 (Covariance of the Ext^A_C)**.** *The assignment $C \mapsto \mathsf{Ext}^A_C$ extends to a functor*

$$\mathsf{Ext}^A\colon \mathbf{Cat} \to \mathscr{C}at \nearrow \mathbf{Cat}.$$

Proof. Given a functor $\mathsf{P}\colon C \to D$, post-composition with P defines a functor $\mathsf{P}_*\colon C^A \to D^A$. Then there is a natural transformation

$$\begin{tikzcd}
\mathbf{Past}(A)^{\mathrm{op}} \times C^A \arrow[r, "\mathsf{Ext}^A_C"] \arrow[d, "\mathrm{id}\times\mathsf{P}_*"'] & \mathbf{Cat} \\
\mathbf{Past}(A)^{\mathrm{op}} \times D^A \arrow[ur, "\mathsf{Ext}^A_D"'] \arrow[u, phantom, shift left=6, "\Leftarrow \mathsf{Ext}^A_{\mathsf{P}}"] &
\end{tikzcd} \tag{4}$$

defined as follows: given a past extension $\iota\colon A \hookrightarrow B$ and a functor $\mathsf{F}\colon A \to C$, the functor

$$\mathsf{Ext}^A_{\mathsf{P}}(\iota,\mathsf{F})\colon \mathsf{Ext}^A_C(\iota,\mathsf{F}) \to \mathsf{Ext}^A_D(\iota,\mathsf{F} \mathbin{⨾} \mathsf{P})$$

acts both on objects and on morphisms by post-composition with P. It is straightforward to check that the assignment $\mathsf{P} \mapsto \mathsf{Ext}^A_{\mathsf{P}}$ respects identities and composition in **Cat**. ■

Remark 5 (General functoriality pattern)**.** A fixed morphism K in $\mathbf{Past}(A)$ is classified by a functor $\vec{I} \to \mathbf{Past}(A)$. Evaluating Ext_C^A at K thus determines a functor

$$\mathsf{Ext}_C^A(\mathsf{K},-)\colon \vec{I} \times C^A \to \mathbf{Cat},$$

which we can curry to obtain a functor

$$\Lambda.\mathsf{Ext}_C^A(\mathsf{K},-)\colon C^A \to \mathbf{Cat}^{\vec{I}}. \tag{5}$$

Given a functor $\mathsf{P}\colon C \to D$, we can also “curry the natural transformation” in (4) to obtain a diagram

$$\begin{tikzcd} C^A \arrow[rr, "\Lambda.\mathsf{Ext}_C^A(\mathsf{K},-)"] \arrow[d, "\mathsf{P}_*"'] \arrow[dr, phantom, "\overset{\Lambda.\mathsf{Ext}_\mathsf{P}^A(\mathsf{K},-)}{\Swarrow}"] & & \mathbf{Cat}^{\vec{I}} \\ D^A \arrow[urr, "\Lambda.\mathsf{Ext}_D^A(\mathsf{K},-)"'] & & \end{tikzcd} \tag{6}$$

which is part of a functor $\mathbf{Cat} \to \mathscr{C}at \upharpoonright \mathbf{Cat}^{\vec{I}}$.

Post-composing with the functor $\mathbf{Cat}^{\vec{I}} \to \mathbf{Pos}_\bullet$ from (1) we obtain a covariant family of functors $C^A \to \mathbf{Pos}_\bullet$.

We will show that, for suitable choices of A and K, the image of these functors is included in the subcategory of $\mathbf{Pos}_\bullet$ on the zeroth and first homotopy posets of C or categories associated with C, exhibiting various kinds of functorial dependence of homotopy posets.

Proposition 10 (Functoriality of the homotopy posets)**.** *Let C be a category, $i \in \{0,1\}$. Then:*

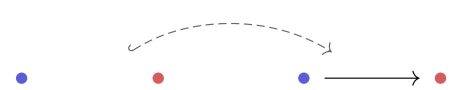
1. *the assignment $x \mapsto \pi_i(C/_x)$ extends to a functor $\pi_i(C/_-)\colon C \to \mathbf{Pos}_\bullet$;*
2. *a functor $\mathsf{F}\colon C \to D$ induces a natural transformation $\pi_i(\mathsf{F})\colon \pi_i(C/_-) \Rightarrow \pi_i(D/_{\mathsf{F}-})$.*

Given another functor $\mathsf{G}\colon D \to E$, this assignment satisfies

$$\pi_i(\mathsf{F} \mathbin{⨾} \mathsf{G}) = \pi_i(\mathsf{F}) \mathbin{⨾} \pi_i(\mathsf{G}), \qquad \pi_i(\mathrm{id}_C) = \mathrm{id}_{\pi_i(C/_-)}.$$

Proof. We will derive the results for both $i \in \{0,1\}$ from the general functoriality pattern of Remark 5.

First we consider the case $i = 0$. Let $\mathbf{1}$ be the terminal category. The inclusion K_0 of the endpoints of the walking arrow induces a morphism in $\mathbf{Past}(\mathbf{1})$, depicted as follows:



$$\begin{tikzcd} & \mathbf{1} \arrow[dl, hook, "\iota_1"'] \arrow[dr, hook, "1"] & \\ \mathbf{1}+\mathbf{1} \arrow[rr, hook, "\mathsf{K}_0"'] & & \vec{I} \end{tikzcd}$$

We claim that, up to isomorphism of categories,

$$\Lambda.\mathsf{Ext}_C^{\mathbf{1}}(\mathsf{K}_0,-)\colon C^1 \to \mathbf{Cat}^{\vec{I}}$$

sends an object x of $C^{\mathbf{1}}$ — which is, equivalently, an object of C — to the slice projection functor

$$\mathrm{dom}\colon C/_x \to C.$$

The domain of $\Lambda.\mathsf{Ext}_C^{\mathbf{1}}(\mathsf{K}_0,x)$ is the category $\mathsf{Ext}_C^{\mathbf{1}}(1,x)$ whose

- objects are functors $f\colon \vec{I} \to C$ such that

$$\begin{tikzcd} \mathbf{1} \arrow[r, "x"] \arrow[d, hook, "1"] & C \\ \vec{I} \arrow[ur, "f"'] & \end{tikzcd}$$

  commutes, which are in bijection with morphisms f of C whose codomain is x, and

- morphisms from f to g are natural transformations $h\colon f \Rightarrow g$ — which are in bijection with commutative squares

$$\begin{tikzcd} y \arrow[r, "h_0"] \arrow[d, "f"'] & z \arrow[d, "g"] \\ x \arrow[r, "h_1"'] & x \end{tikzcd}$$

  in C — that restrict to the identity along $1\colon \mathbf{1} \hookrightarrow \vec{I}$, that is, are such that $h_1 = \mathrm{id}_x$. These are in bijection with factorisations of f through g.

This establishes an isomorphism between $\mathsf{Ext}^{\mathbf{1}}_C(1,x)$ and $C/_x$. The codomain of $\Lambda.\mathsf{Ext}^{\mathbf{1}}_C(\mathsf{K}_0,x)$ is the category $\mathsf{Ext}^{\mathbf{1}}_C(\iota_1,x)$ whose

- objects are functors $(y_0,y_1)\colon \mathbf{1}+\mathbf{1} \to C$ such that

$$\begin{tikzcd} \mathbf{1} \arrow[r, "x"] \arrow[d, hook, "\iota_1"] & C \\ \mathbf{1}+\mathbf{1} \arrow[ur, "(y_0,y_1)"'] & \end{tikzcd}$$

  commutes, which are in bijection with pairs of objects (y_0,y_1) of C such that $y_1 = x$, which are in bijection with objects of C, and

- morphisms from (y,x) to (z,x) are in bijection with pairs of morphisms

$$\begin{tikzcd} y \arrow[r, "h_0"] & z \\ x \arrow[r, "h_1"'] & x \end{tikzcd}$$

  in C that restrict to the identity along ι_1, that is, are such that $h_1 = \mathrm{id}_x$. These are in bijection with morphisms $y \to z$.

This establishes an isomorphism between $\mathsf{Ext}^{\mathbf{1}}_C(\iota_1,x)$ and C. The functor $\mathsf{Ext}^{\mathbf{1}}_C(\mathsf{K}_0,x)$ acts by restriction of $f\colon \vec{I} \to C$ along $\mathsf{K}_0\colon \mathbf{1}+\mathbf{1} \hookrightarrow \vec{I}$; through the isomorphisms, this acts by mapping $f\colon y \to x$ to its domain y. This is, by inspection, the same as the action of dom.

We define

$$\pi_0(C/_{-})\colon C \to \mathbf{Pos}_{\bullet}$$

to be the post-composition of $\Lambda.\mathsf{Ext}^{\mathbf{1}}_C(\mathsf{K}_0,-)$ with the functor of Equation 1. It follows from our argument that, up to isomorphism, this sends x to the homotopy poset $\pi_0(C/_x)$. The covariance in C then follows as an instance of Equation 6: given a functor $\mathsf{F}\colon C \to D$, we whisker the natural transformation $\Lambda.\mathsf{Ext}^{\mathbf{1}}_{\mathsf{F}}(\mathsf{K}_0, -)$ with the functor of (1) to obtain $\pi_0(\mathsf{F})\colon \pi_0(C/\_) \Rightarrow \pi_0(D/\mathsf{F}\_)$.

Now, let us focus on the first homotopy poset. The functor K_1 identifying two parallel arrows also induces a morphism in $\mathbf{Past}(\mathbf{1})$, depicted as follows:

$$\begin{tikzcd} & \mathbf{1} \arrow[dl, hook, "c"'] \arrow[dr, hook, "1"] & \\ \mathrm{Par} \arrow[rr, hook, "\mathsf{K}_1"'] & & \vec{I} \end{tikzcd}$$

Here, Par denotes the “walking parallel pair of arrows”. We claim that, up to isomorphism of categories,

$$\Lambda.\mathsf{Ext}^{\mathbf{1}}_{C}(\mathsf{K}_1, -)\colon C \to \mathbf{Cat}^{\vec{I}}$$

sends an object x of C to the slice projection functor

$$\mathrm{dom}\colon \mathrm{Par}(C/x)/(\mathrm{id}_x, \mathrm{id}_x) \to \mathrm{Par}(C/x).$$

We have already established that the domain of $\Lambda.\mathsf{Ext}^{\mathbf{1}}_{C}(\mathsf{K}_1, x)$, which is the category $\mathsf{Ext}^{\mathbf{1}}_{C}(1, x)$, is isomorphic to C/x, which can be shown to be isomorphic to $\mathrm{Par}(C/x)/(\mathrm{id}_x, \mathrm{id}_x)$ using Proposition 7.

The codomain of $\Lambda.\mathsf{Ext}^{\mathbf{1}}_{C}(\mathsf{K}_1, x)$ is the category $\mathsf{Ext}^{\mathbf{1}}_{C}(c, x)$ whose

- objects are functors $(f_0, f_1)\colon \mathrm{Par} \to C$ such that

$$\begin{tikzcd} \mathbf{1} \arrow[r, "x"] \arrow[d, hook, "c"] & C \\ \mathrm{Par} \arrow[ur, "(f_0, f_1)"'] & \end{tikzcd}$$

  commutes, which are in bijection with pairs of morphisms (f_0, f_1) of C whose codomain is x, and

- morphisms from the pair (f_0, f_1) to (g_0, g_1) are natural transformations $h\colon (f_0, f_1) \Rightarrow (g_0, g_1)$ that restrict to the identity along c, which are in bijection with morphisms h such that $f_0 = h; g_0$ and $f_1 = h; g_1$.

This establishes an isomorphism between $\mathsf{Ext}^{\mathbf{1}}_{C}(c, x)$ and $\mathrm{Par}(C/x)$.

The functor $\mathsf{Ext}^{\mathbf{1}}_{C}(\mathsf{K}_1, x)$ acts by precomposing $f\colon \vec{I} \to C$ with $\mathsf{K}_1\colon \mathrm{Par} \to \vec{I}$, which through the isomorphisms sends a pair (f, f) with its unique morphism to $(\mathrm{id}_x, \mathrm{id}_x)$ to the pair (f, f) on its own. This is, by inspection, the same as the action of dom.

We define

$$\pi_1(C/\_)\colon C \to \mathbf{Pos}_{\bullet}$$

to be the post-composition of $\Lambda.\mathsf{Ext}^{\mathbf{1}}_{C}(\mathsf{K}_1, -)$ with the functor of Equation 1. It follows from our argument that, up to isomorphism, this sends x to the homotopy poset $\pi_1(C/x)$. Again, we obtain covariance in C by whiskering instances of Equation 6. This completes the proof. ∎

Posetal Diagrams for Logically-Structured Semistrict Higher Categories

Chiara Sarti
Computer Laboratory
University of Cambridge
cs2197@cam.ac.uk

Jamie Vicary
Computer Laboratory
University of Cambridge
jamie.vicary@cl.cam.ac.uk

We now have a wide range of proof assistants available for compositional reasoning in monoidal or higher categories which are free on some generating signature. However, none of these allow us to represent categorical operations such as products, equalizers, and similar logical techniques. Here we show how the foundational mathematical formalism of one such proof assistant can be generalized, replacing the conventional notion of string diagram as a geometrical entity living inside an n-cube with a posetal variant that allows exotic branching structure. We show that these generalized diagrams have richer behaviour with respect to categorical limits, and give an algorithm for computing limits in this setting, with a view towards future application in proof assistants.

1 Introduction

The development of proof assistants for category theory and higher category theory has recently been a active area for the applied category theory community, in particular from a string diagrammatic perspective. Recent work has included the CARTOGRAPHER tool of Sobocinski, Wilson and Zanasi applying hypergraph rewriting for symmetric monoidal diagrams [15]; the DISCOPY python library from de Felice et al for string diagram manipulation [7]; the REWALT tool by Hadzihasanovic and Kessler for rewriting with diagrammatic sets [8]; the WIGGLE.PY tool due to Burton for 3d string diagram rendering [5]; the QUANTOMATIC system developed by Dixon, Duncan, Kissinger and others for applying the ZX calculus in quantum information [6]; and the GLOBULAR [2, 3] and HOMOTOPY.IO [10, 13] web-based systems for finitely-generated semistrict n-categories rendered as higher string diagrams. While these tools represent a wide range of perspectives and use-cases, they share a common goal of allowing the user to manipulate terms in a monoidal or higher categorical structure which is freely generated under composition from some signature, perhaps with additional algebraic elements (for example, such as Frobenius algebras, in the case of QUANTOMATIC.)

The geometrical essence of these proof assistants allows the user to avoid some of the bureaucracy associated with some algebraic approaches to higher categories. However, much of the power of category theory arises from methods that go beyond direct composition, such as products, equalizers, colimits, and other standard categorical structures. These cannot be represented with any of the current family of string diagrammatic proof assistants.

Here we explore an alternative foundation for such proof assistants which may suggest a path towards new classes of tools, with the potential to combine the clarity and usability of string diagrammatic techniques, with the power of algebraic categorical methods. We illustrate our approach with the *zigzag construction*, a simple combinatorial model of higher string diagram that forms the basis of the proof assistant HOMOTOPY.IO. This construction depends on a contravariant equivalence between the augmented simplex category Δ_+, of finite total orders and monotone functions; and the category $\Delta_=$ of non-empty

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 246–259, doi:10.4204/EPTCS.397.15

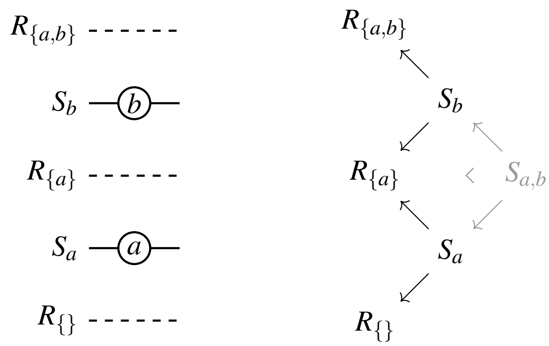


Figure 1: The linear zigzag over the total order $\{a \to b\}$.

finite total orders, with monotone functions preserving min and max elements. Elegantly described by Wraith [17], the equivalence is representable, acting as $\Delta_+(-, \{\perp \to \top\}) : \Delta_+^{\mathrm{op}} \to \Delta_=$.

The idea of the zigzag construction is sketched in Figure 1, for the 2-element total order $\{a \to b\} \in \Delta_+$. On the left, the two elements of $\{a \to b\}$ appear as *singular heights* S_a, S_b. These are interlaced vertically with the three elements of its dual $\Delta_+(\{a \to b\}, \{\perp \to \top\})$ written as *regular heights* $R_{\{\}}$, $R_{\{a\}}$, $R_{\{a,b\}}$, with the subscript indicating the preimage of $\top$. On the right, we equip the singular heights with maps into the adjacent regular heights.[1] We see an alternating pattern of regular and singular heights, drawn as dashed and solid lines; reading from bottom to top, we can consider these as "time slices" of the linear geometry of the diagram, which tell us that a happens first, and then b. For every category C we have a *zigzag category* $\mathsf{Z}(\mathsf{C})$, and an n-dimensional string diagram is simply an object of $\mathsf{Z}^n(\mathsf{C})$ over some suitable base category. For every zigzag category, a functor $R : \mathsf{Z}(\mathsf{C}) \to \Delta_=$ projects down to the total order of regular heights. Concerning the existence of limits in $\mathsf{Z}(\mathsf{C})$, a decision procedure has recently been presented [13], which supplies the following necessary condition: for a diagram in $\mathsf{Z}(\mathsf{C})$ to have a limit, its projection in $\Delta_=$ must have a limit. This rules out wide classes of limits in $\mathsf{Z}(\mathsf{C})$, since the limit structure of $\Delta_=$ is meagre. In particular, products do not exist in $\Delta_=$ for sets of cardinality greater than 1, so there can be no notion of product for nontrivial higher string diagrams.

However, the duality $\Delta_+^{\mathrm{op}} \simeq \Delta_=$ extends further. In particular, Wraith's original paper further exhibited a duality $\mathrm{FPos}^{\mathrm{op}} \simeq \mathrm{FDLat}$, where FPos is the category of finite posets, and FDLat is the category of finite distributive lattices and meet- and join- preserving functors; this can be considered a finitised version of Stone Duality, and is sometimes known as the Birkhoff Representation Theorem [4, 17]. This equivalence is again representable as $\mathrm{FPos}(-, \{\perp \to \top\})$, and extends the duality considered above, given the obvious full and faithful embeddings $\Delta_+ \hookrightarrow \mathrm{FPos}$ and $\Delta_= \hookrightarrow \mathrm{FDLat}$.

Here we exploit this duality to put forward a *posetal zigzag construction*, directly generalizing the existing linear zigzag construction. While the linear zigzag construction yields a notion of higher category which is purely compositional, we conjecture that the posetal zigzag construction will yield a notion of higher category with richer categorical structure, but retaining a geometrical essence which could in principle be implemented in a similar tool to HOMOTOPY.IO. Establishing this conjecture will require considerable future work.

In this paper we take the first steps, giving the first detailed investigation of posetal zigzags. We begin

[1] These maps are going in the opposite direction than in some previous literature, a technical choice we made following [9] which makes our development here more straightforward.

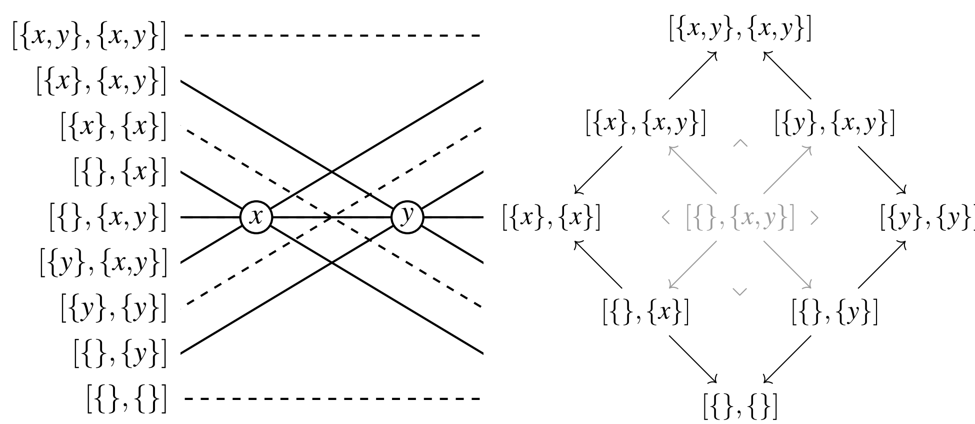


Figure 2: The posetal zigzag over the poset $\{x \mid y\}$.

here with an informal illustration of posetal zigzags, generalizing the linear example of Figure 1 above. In Figure 2 we illustrate the posetal zigzag construction for the poset $\{x \mid y\}$, with two disconnected objects. While the linear zigzags in Figure 1 had a linear sequence of time slices (since a must precede b), here a richer structure appears, with x and y now interpreted as "events" that can occur in either order. Now we have a dual distributive lattice $D = \mathrm{FPos}(\{x\ \ y\},\{\bot \to \top\})$ with elements $\{\}$, $\{x\}$, $\{y\}$, $\{x,y\}$ under the subset order, once again denoting in braces the preimage of $\top$. Our regular and singular levels are indexed by pairs of related elements $[A,B]$ in D, which we call *intervals*; by construction, A,B will be subsets of the original poset, with $A \subseteq B$. For this example there are 9 such intervals, and we list them on the left of Figure 2, once again considering them as distinct "time slices". For example, $[\{x\},\{x,y\}]$ represents the time slice in which x has already occurred, and y is in the moment of occurring. More generally, one can interpret any interval $[A,B]$ as the time slice in which the events A has already occurred, and the events $B \setminus A$ are occurring at that exact moment.

On the right of Figure 2, we present these 9 intervals in a different way, as nodes on a diamond grid, interpreted as a diagram in an underlying process category C. Morphisms arise as interval refinements, and we observe the presence of 4 squares, which are required to commute. As we move from the bottom to the top node by various paths, we observe different sequences of time slices: moving clockwise we observe the sequence $[\{\},\{\}]$, $[\{\},\{x\}]$, $[\{x\},\{x\}]$, $[\{x\},\{x,y\}]$, $[\{x,y\},\{x,y\}]$, interpreted as x occurring before y; moving anticlockwise we observe the sequence $[\{\},\{\}]$, $[\{\},\{y\}]$, $[\{y\},\{y\}]$, $[\{y\},\{x,y\}]$, $[\{x,y\},\{x,y\}]$, interpreted as y occurring before x; and moving vertically one observes the sequence $[\{\},\{\}]$, $[\{\},\{x,y\}]$, $[\{x,y\},\{x,y\}]$, in which x and y occur simultaneously. The posetal zigzag diagram provides semantic data in the base category C for all of these possibilities. The fact that the 4 squares commute means that whatever sequence one chooses, the semantic effect in C is the same. The fact that the squares are pullbacks means that much of this data is in fact redundant; the central part, drawn in gray, can be considered as "filler data" that need not be explicitly stored. Figure 1 contained similar redundant filler data, with the singular level $S_{a,b}$ required to arise as a pullback of S_a and S_b.

This account is of course informal, and intended to give intuition ahead of the precise mathematical development that follows. In Section 2 we introduce the notion of interval for a poset, and give a notion of labelled diagram structure, with label data assigned to all intervals, but with no pullback conditions imposed on filler data. In Section 3 we introduce the more refined posetal zigzag construction $\mathsf{P}(\mathsf{C})$, where the filler data is now obtained canonically via pullback, yielding a well-behaved conservative extension of the traditional linear zigzag formalism. In Section 4 we consider the construction of limits in posetal zigzag categories, giving an explicit construction procedure, and establishing our main result, Corollary 4.12, which states that if C has all finite limits, then so does $\mathsf{P}(\mathsf{C})$.

In this way, we establish the posetal zigzag construction as a potential foundation for a new class of geometrical proof assistant, with additional expressive power. In this setting, higher string diagrams would retain their semistrict geometrical essence, yet also exhibit new posetal features such as branches, sinks and forks; n-dimensional diagrams are no longer inscribed neatly within the n-cube, as with traditional higher string diagrams. Depending on the poset structure, the sequencing of morphisms within the string diagram would be dynamic, with re-sequencing steps appearing as higher cells. While the future applications of these ideas of course remain speculative, we hope our work may lead to further development of geometrical proof assistants with exciting new capabilities.

Acknowledgements.

We thank Lukas Heidemann, Alex Rice and Calin Tataru for many insightful discussions. We also thank the anonymous reviewers for their invaluable feedback. The first author acknowledges funding from King's College Cambridge. The second author acknowledges funding from the Royal Society.

2 The Interval Construction

2.1 From Posets to Labelled Intervals

Our formal development begins with a reconstruction of the categorical origins of the zigzag construction, which we generalise to poset shapes. We will initially focus our analysis on the combinatorial aspects of our theory, setting solid foundations for the establishing the theoretical results of Section 4, and allowing for the geometric content of our theory to emerge organically as a by-product of our categorical analysis.

In our discussion, we study the maps of our diagrams as factored or decomposed in two parts: one which may change the posetal shape but not the labels, and another which may change the labels, but fixes the shape. By adding the possibility of extra "filler data" in our diagrams, this relabelling map can be specified as a natural transformation between two functors. This sort of decomposition can be expressed naturally in the language of Grothendieck fibrations, unlocking powerful theoretical tools. Without getting too ahead of our formal development, this Section will build up to a formal description of the construction we depicted in Figure 1 and Figure 2, starting from the following key idea:

Definition 2.1 (Interval)**.** An *interval* in a poset P is a pair $[a,a']$ of $a,a' \in P$ with $a \leq a'$. We denote the set of intervals in P by $[P]$ and order it by the precision relation $\supseteq$

$$[a,a'] \supseteq [b,b'] :\longleftrightarrow (a \leq b) \wedge (b' \leq a')$$

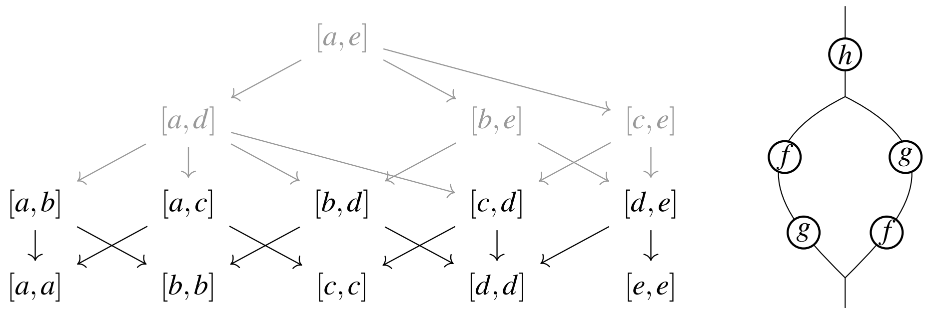
The strict counterpart of this relation is denoted $\supset$.

Proposition 2.2. *For every finite poset P, $[P]$ is also a finite poset.*

Proof. Finiteness is evident. We must show that $[P]$ inherits transitivity, reflexivity and anti-symmetry from P. For transitivity, if we have $[a,a'] \supseteq [b,b']$ and $[b,b'] \supseteq [c,c']$, then we must have $a \leq b \leq c$ and $c' \leq b' \leq a'$, so by transitivity of P, we have $a \leq c$ and $c' \leq a'$, and hence $[a,a'] \supseteq [c,c']$.

Reflexivity is evident, and for anti-symmetry, if we have $[a,a'] \supseteq [b,b']$ and $[b,b'] \supseteq [a,a']$, then we must have $a \leq b \leq a$ and $b' \leq a' \leq b'$, and hence $a = b$ and $a' = b'$ by anti-symmetry of P. $\square$

Example 2.3. The interval construction $[P]$ on the poset $P = \{a < (b \mid c) < d < e\}$ is depicted as the left-most diagram below. The data of the intervals below the gray arrows is easily rendered as the posetal string diagram on the right. This requires us to ignore the witnesses from the higher intervals, which have no clear interpretation on the diagrammatic side.



With anti-symmetry in mind, we specialise our terminology as follows:

Definition 2.4 (Degenerate Intervals)**.** We refer to an interval $[a,a']$ in P as a *degenerate*, if $a = a'$, or *non-degenerate*, if $a < a'$.

In the account of this construction offered in [13], degenerate intervals are called regular heights, and non-degenerate intervals are called singular heights. Though the alternative choice of terminology is supported by good motivation, we avoid it in our discussion to spare the reader from unnecessary confusion.

Definition 2.5 (Map of Intervals)**.** Let $f : P \to Q$ be a monotone function between posets. The associated *map of interval posets* $[f] : [P] \to [Q]$ is the function sending an interval $[a,a']$ in P to $[f(a), f(a')]$.

Proposition 2.6. *Let $f : P \to Q$ be a monotone function between posets. Then $[f]$ is a well-defined monotone map, and moreover, $[-]$ defines an endofunctor on* FPos*.*

Proof. Let $f : P \to Q$ be a monotone map. Then $a \le a'$ implies $f(a) \le f(a')$, hence if $[a,a']$ is an interval in P, $[f][a,a']$ is an interval in Q. Moreover, if we have $[a,a'] \supseteq [b,b']$, i.e. $a \le b$ and $b' \le a'$, then $f(a) \le f(b)$ and $f(b') \le f(a')$, and thus $[f][a,a'] \supseteq [f][b,b']$, which shows $[f]$ is monotone. Finally, the assignment $P \mapsto [P]$ and $f \mapsto [f]$ respects identities and composites, and thus makes $[-]$ into an endofunctor FPos. $\square$

Lemma 2.7. *The functor $[-]$ preserves products.*

Proof. Let P and Q be finite posets. An interval in $P \times Q$ is a pair of pairs $[(a,b),(a',b')]$ with $a \le a'$ and $b \le b'$, and thus defines two intervals $[a,a']$ and $[b,b']$. Moreover, the assignment and its inverse are monotone:

$$\begin{aligned}
[(a,b),(a',b')] \supseteq [(c,d),(c',d')] &\longleftrightarrow (a,b) \le (c,d) \wedge (c',d') \le (a',b') \\
&\longleftrightarrow a \le c \wedge b \le d \wedge c' \le a' \wedge d' \le b' \\
&\longleftrightarrow [a,a'] \supseteq [c,c'] \wedge [b,b'] \supseteq [d,d'].
\end{aligned}$$

$\square$

We now identify FPos with a corresponding full subcategory of Cat, by identifying each poset P with the category P with the elements of P for objects, and arrows $a \to a'$, whenever $a \leq a'$. Note that under this identification, the interval construction is isomorphic to the twisted arrow construction [11, §2]. The above data is combined as follows:

Definition 2.8 (Labelled Interval Category)**.** Let C be a category. The category $\mathsf{L}(\mathrm{C})$ of *intervals labelled in* C is defined as the Grothendieck construction of the functor

$$\begin{aligned} L_{\mathrm{C}} : \mathrm{FPos}^{\mathrm{op}} &\longrightarrow \mathrm{Cat} \\ P &\longmapsto \mathrm{Func}([P], \mathrm{C}), \\ f &\longmapsto - \circ [f]. \end{aligned}$$

Explicitly, the objects of $\mathsf{L}(\mathrm{C})$ are pairs (P, X), with $P \in$ FPos, a shape poset, and $X : [P] \to \mathrm{C}$, a labelling of $[P]$ in C. As for the morphisms, they are given by pairs (f, α), with $f : P \to Q$, a change of shape map, and $\alpha : X \to Y \circ [f]$ a relabelling natural transformation.

Example 2.9. Let P be the poset $\{a < (b \,|\, c) < d < e\}$ of Example 2.3 and C be the thin category generated by $\{x < (f \,|\, g \,|\, h) < (\alpha \,|\, \beta \,|\, \gamma) < \mu\}$. Then the pair (P, X) defines an object of $\mathsf{L}(\mathrm{C})$, where $X : [P] \to \mathrm{C}$ is the functor represented in the diagram below:

$$\begin{tikzcd}
 & g \arrow[dl] \arrow[r] & x & f \arrow[l] \arrow[dr] & \beta \arrow[l] \arrow[dr] & & \\
x & & \alpha \arrow[ul] \arrow[dl] \arrow[ur] \arrow[dr] & \mu \arrow[l, gray] \arrow[ur, gray] \arrow[dr, gray] & x & h \arrow[l] \arrow[r] & x \\
 & f \arrow[ul] \arrow[r] & x & g \arrow[l] \arrow[ur] & \gamma \arrow[l] \arrow[ur] & &
\end{tikzcd}$$

Example 2.10. As shapes for cells in a higher category, labelled intervals can admit extremely undesirable behaviour: they need not even be connected. For instance, if $P := 1 + 1$ is the discrete two-element poset, then $[P] \cong P$ and thus a labelled interval of shape P simply picks out two objects of C.

Example 2.11. A map of labelled intervals can be decomposed as a change of shape monotone function and a relabelling natural transformation. In the diagram below, the monotone map is between the posets $P := \{\bot \to \top\}$ and $Q := \{\mathrm{a} < (\mathrm{b} \mid \mathrm{c}) < \mathrm{d}\}$, and acts by $\bot \mapsto a, \top \mapsto d$. The relabelling has components id_x, $f \to g$ and id_y.

$$\begin{tikzcd}
 & g \arrow[dl] \arrow[r] & y & y \arrow[l] \arrow[dr] & \\
x & & & f \arrow[ull, gray] \arrow[u, gray] \arrow[dll, gray] \arrow[d, gray] & y \\
 & x \arrow[ul] \arrow[r] & x & g \arrow[l] \arrow[ur] & \\
 & x \arrow[uul] & f \arrow[l] \arrow[r] \arrow[uur] & y \arrow[uur] &
\end{tikzcd}$$

By virtue of its definition through a Grothendieck construction, we have an underlying shape functor $U : \mathsf{L}(\mathrm{C}) \to \mathrm{FPos}$, which is a Grothendieck fibration. If we adopt the convention of writing (x^0, x^1) for the components of an object $x \in \mathsf{L}(\mathrm{C})$, and similarly for morphisms, then the action of U is simply specified by taking first projections $x \mapsto x^0, f \mapsto f^0$.

Given a functor $F : \mathsf{C} \to \mathsf{D}$, we may define a post-composition with F mapping $\mathsf{L}(F) : \mathsf{L}(\mathsf{C}) \to \mathsf{L}(\mathsf{D})$, which acts by $(P,X) \mapsto (P, F \circ X)$ and $(f,\alpha) \mapsto (f, F\alpha)$. Then $\mathsf{L}(F)$ is a functor, and moreover it allows us to make an assignment $\mathsf{L}(-) : \mathsf{C} \mapsto \mathsf{L}(\mathsf{C}), F \mapsto \mathsf{L}(F)$, which is an endofunctor on Cat, with the usual caveats about size concerns we trust our readers to judge unproblematic. In particular, this allows us to iterate the construction, as $\mathsf{L}(\mathsf{C})$ can itself be taken as a category of labels, and we may thus define iterated versions of this construction by setting $\mathsf{L}^{n+1}(\mathsf{C}) := \mathsf{L}(\mathsf{L}^n(\mathsf{C}))$.

It seems useful at this point, before drawing this Section to an end, to briefly summarise our progress towards the overarching goal of posetal diagrams. The interval construction introduced in Definition 2.1 captures the combinatorial aspect of our intended construction remarkably well. Unfortunately, as we have seen in Example 2.3, the interval construction introduces labels which are foreign to the diagrammatic calculus. In keeping with our aim of being able to reconstruct the combinatorial object from the geometric representation, we need to ensure that those labels do not carry information that cannot be inferred from the explicit datum of a diagram. We will see in the next Section that this canonicity requirement can be succinctly stated in terms of limit constructions.

3 Posetal Diagrams

3.1 Posetal Diagrams as Local Functors

In the previous Section we have presented the construction of the category of labelled intervals, arguing that it correctly captures the desiderata combinatorics our theory. However, this approach requires too much filler data to be specified, preventing a clean diagrammatic presentation of our structures. In drawing intuition from Example 2.3, we wish to find technical conditions under which the missing filler data in the diagrams can be faithfully reconstructed.

A close inspection of Example 2.3 and Example 2.9 shows that all the intervals whose data we wish to suppress satisfy the universal property of being a pullback in $[P]$. This would suggest we consider labellings $X : [P] \to \mathsf{C}$ which preserve pullbacks. However, this condition is far too strong for our interest: $[P]$ is a thin category, and thus every arrow is monic. If X were to preserve all pullbacks, it could only involve monic arrows in C, which is exceedingly restrictive, especially in light of our wishes to iterate the construction. We will thus exercise some care in determining exactly which pullbacks in $[P]$ should be preserved:

Definition 3.1 (Atomic Cospan, Local Functor)**.** Let J, C be categories. A cospan $(a \xrightarrow{f} x \xleftarrow{g} b)$ in J is *atomic* if for any cospan $(a \xrightarrow{f'} y \xleftarrow{g'} b)$ and map $h : y \to x$ with $h \circ f = f'$ and $h \circ g = g'$, then h is an isomorphism. We say a functor $X : \mathsf{J} \to \mathsf{C}$ is *local* if it preserves all pullbacks of atomic cospans.

Example 3.2. If f is not an isomorphism, then $(a \xrightarrow{f} x \xleftarrow{f} a)$ is never atomic. Hence non-iso monic arrows need not be preserved by local functors.

Example 3.3. The cospan $[b,e] \supset [e,e] \subset [c,e]$ in our Example 2.3 is also not atomic.

Example 3.4. The labelling of the diamond $\{a < (b\,|\,c) < d\}$ in Set depicted below is a local functor:

$$\begin{tikzcd}
 & & & 1+1 \arrow[dlll] \arrow[dll] \arrow[drr] \arrow[drrr] & & & \\
1 \arrow[d] \arrow[dr] & 1+1 \arrow[dl] \arrow[drrrr] & & & & 1+1 \arrow[dllll] \arrow[dr] & 1 \arrow[d] \arrow[dl] \\
1 & 1 & & & & 1 & 1
\end{tikzcd}$$

Though this technical condition seems cumbersome to check explicitly, a remarkable fact about our formalism is that for a rather large class of poset shapes we almost never actually need to do this. We will prove in the remainder of the paper that so long as the diagrams are constructed by taking finite limits of other local diagrams of the right shape, we will remain within this fragment of our framework. For our interest in using this combinatorial framework as the basis of a future proof assistant, this shows we can maintain strong invariants on our data structures, so that any actual implementation of our procedures could drastically reduce the data it needs to explicitly keep track of.

Another, even more surprising property of our formalism is that this class of well-behaved poset shapes can be morally taken to be the whole of FPos, albeit viewed through a looking glass. This is because we may take our well-behaved posets to be finite distributive lattices, by which we mean posets P admitting finite meets and joins and that satisfying the condition that for all $a,b,c \in P$ we have $a \wedge (b \vee c) = (a \vee b) \wedge (a \vee c)$ and $a \vee (b \wedge c) = (a \wedge b) \vee (a \wedge c)$. Such posets assemble into a non-full subcategory FDLat of FPos by taking as maps monotone functions $f : P \to Q$ preserving finite meets and joins. Note that, in particular, our lattices are always bounded, and moreover if f is a lattice homomorphism then we must have $f(\bot) = \bot$ and $f(\top) = \top$. Although in our formal development some individual results would hold with weaker regularity conditions, the category FDLat enjoys the property of being equivalent to $\mathrm{FPos}^{\mathrm{op}}$ by the Birkhoff Representation Theorem [17, page 262]. The sum of these wonderful properties suggests us the following definition:

Definition 3.5 (Labelled Posetal Diagram)**.** The category $\mathsf{P}(\mathsf{C})$ of *posetal diagrams labelled in* C is defined to be the subcategory of labelled intervals $\mathsf{L}(\mathsf{C})$ with objects pairs (P,X) with P being a distributive lattice and X a local functor, and morphisms pairs (f,α), with f a lattice homomorphism.

3.2 Intervals in Lattices

Having formally introduced our key notion of posetal diagrams in Section 3.1, we will now embark a fine-grained analysis of the relation between logical properties of a poset P and the locality property of the labelled interval with shape P. Our key result for this Section will be an explicit characterisation of atomic cospans in $[P]$ for distributive lattices. This will allow us to extract useful consequences about preservation of locality under proposition by interval maps associated with lattice homomorphisms. To do this, however, we first require some intermediate lemmas.

Lemma 3.6. *Let P be a finite poset with binary meets and joins. Then, for any two intervals $[a,a']$ and $[b,b']$ in $[P]$, the meet $[a,a'] \wedge [b,b']$ exists and is given by $[a \wedge b, a' \vee b']$.*

Proof. We have $a \wedge b \leq a$ and $a' \leq a' \vee b'$, and similarly for $[b,b']$, hence $[a \wedge b, a' \vee b']$ is an interval and $[a \wedge b, a' \vee b'] \supseteq [a,a'],[b,b']$. Moreover, for any other interval $[c,c']$ in P with $[c,c'] \supseteq [a,a'],[b,b']$, we must have $c \leq a$ and $c \leq b$, and thus $c \leq a \wedge b$. The dual calculation shows $[c,c'] \supseteq [a \wedge b, a' \vee b']$. ☐

Lemma 3.7. *Let P be a finite poset with binary meets and joins. Then, for any two intervals $[a,a']$ and $[b,b']$ in $[P]$, the join $[a,a'] \vee [b,b']$ exists iff $a \vee b \leq a' \wedge b'$, in which case it is given by $[a \vee b, a' \wedge b']$.*

Proof. Assume $[a,a'] \vee [b,b']$ exists and equal to some interval $[c,c']$. Then in particular we have $a \leq c$ and $b \leq c$, and thus $a \vee b \leq c$, and dually $c' \leq a' \wedge b'$. Since $c \leq c'$, we must have $a \vee b \leq a' \wedge b'$.

We establish the other direction by verifying the universal property of the interval $[a \vee b, a' \wedge b']$, whenever it exists. We have $[a,a'] \supseteq [a \vee b, a' \wedge b']$, and similarly for $[b,b']$, and moreover for every interval $[c,c']$ satisfying the containments $[a,a'] \supseteq [c,c'] \subseteq [b,b']$, we must have $[a \vee b, a' \wedge b'] \supseteq [c,c']$ by the above verifications. ☐

Proposition 3.8. *Let P be a finite lattice. Then a cospan $[a,a'] \supseteq [b,b'] \subseteq [c,c']$ in $[P]$ is atomic iff $a \vee c \leq a' \wedge c'$ and $[b,b'] = [a \vee c, a' \wedge c']$.*

Proof. In the forward direction, we have that $a \leq b$ and $c \leq b$, hence $a \vee c \leq b$ and dually $b' \leq a' \wedge c'$. In particular, $[a \vee c, a' \wedge c'] \supseteq [b,b']$, and thus we have $[a \vee c, a' \wedge c'] = [b,b']$ by atomicity and anti-symmetry. The backward direction is given by the universal property of $a \vee c$ and $a' \wedge c'$. ☐

This yields an immediate but essential consequence for our forthcoming study of limits of posetal diagrams in Section 4.2:

Corollary 3.9. *If $f : P \to Q$ is a lattice homomorphism, $[f]$ preserves atomic cospans and their pullbacks.*

Proof. By Proposition 3.8, a cospan $[a,a'] \supseteq [b,b'] \subseteq [c,c']$ is atomic iff $[b,b'] = [a \vee c, a' \wedge c']$. But lattice homomorphisms preserve binary meets and joins, so $[f][b,b'] = [f(a) \vee f(c), f(a') \wedge f(c')]$, hence the cospan $[f][a,a'] \supseteq [f][b,b'] \subseteq [f][c,c']$ is atomic. Moreover, $[P]$ is a thin category, so the pullback of our atomic cospan coincides with the product of $[a,a']$ and $[c,c']$. By Lemma 3.6, this is given by $[a \wedge c, a' \vee c']$, and thus it is preserved by f. ☐

4 Limits of Posetal Diagrams

4.1 Limit Procedure

Having given the construction of the category $\mathsf{L}(\mathrm{C})$ in our preceding Section, we will now describe a procedure to compute limits in $\mathsf{L}(\mathrm{C})$ from limits in C and FPos. By taking the category of labels C to be a suitable n-fold iterate of the labelled interval construction $\mathsf{L}^n(\mathrm{D})$, this procedure provides a recursive strategy for computing limits entirely in terms of those in the base category D.

Before we describe the limit procedure, since we are interested in stating our results for categories which may be fail to admit many limits, we need to fix some terminology. Recall that for functors $F : \mathrm{J} \to \mathrm{C}$ and $G : \mathrm{C} \to \mathrm{D}$, we say that G preserves F-limits if whenever $(L, \eta : \Delta_L \to F)$ is a limit for F, $(GL, G\eta)$ is a limit for $G \circ F$. We also say that G reflects F-limits if whenever we have a cone (L, η) over F such that $(GL, G\eta)$ is a limit for $G \circ F$, then (L, η) is a limit for F [14, 3.3.1].

Definition 4.1 (Pointwise Limit)**.** Let (L, η) be a limit for a diagram $F : \mathrm{J} \to \mathrm{Func}(\mathrm{C}, \mathrm{D})$. We say (L, η) is *pointwise* if for each $c \in \mathrm{C}$, it is preserved by the evaluation at c functor $\mathrm{ev}_c : \mathrm{Func}(\mathrm{C}, \mathrm{D}) \to \mathrm{D}$.

If D has all J-limits, every J-limit in $\mathrm{Func}(\mathrm{C}, \mathrm{D})$ is pointwise, but this need not be the case otherwise, and our analysis will necessitate the distinction.

Construction 4.2 (Limit Procedure)**.** Given a category of labels C and a finite diagram $F : \mathrm{J} \to \mathsf{L}(\mathrm{C})$, we compute the limit of F or fail according to the following procedure:

1. We take the limit (L, ρ) of the diagram $U \circ F : \mathrm{J} \to \mathsf{L}(\mathrm{C}) \to \mathrm{FPos}$,
2. We use ρ to produce from F a diagram $G : \mathrm{J} \to \mathrm{Func}([L], \mathrm{C})$,
3. For each $j \in \mathrm{J}$, we take $Gj := (Fj)^1 \circ [\rho_j]$,
4. For each $h \in \mathrm{J}(j, j')$, we take $Gh : Gj \to Gj'$ to have components

$$(Gh)_{[a,b]} := (Fh)^1_{[\rho_j][a,b]} : (Fj)^1[\rho_j][a,b] \longrightarrow (Fj')^1[\rho_{j'}][a,b],$$

5. We define $\varepsilon : (\Delta_L, G) \to F$ to have components $\varepsilon_j := (\rho_j, \mathrm{id}_{(Gj)^1})$,

$$\begin{tikzcd}
(L,L) \arrow[d, "{(\mathrm{id}_L,\eta)}"'] & & \\
(L,Gj) \arrow[r, "{(\rho_j,\mathrm{id})}"] \arrow[d, "{(\mathrm{id}_L,Gh)}"'] & Fj \arrow[dr, "Fh"] & \\
(L,Gj') \arrow[rr, "{(\rho_{j'},\mathrm{id})}"'] & & Fj'
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{L}(\mathrm{C}) \arrow[d, "U"'] \\
\mathrm{FPos}
\end{tikzcd}$$

$$\begin{tikzcd}
L \arrow[r, "\rho_j"] \arrow[drr, "\rho_{j'}"'] & (Fj)^0 \arrow[dr, "(Ff)^0"] & \\
& & (Fj')^0
\end{tikzcd}$$

Figure 3: A diagrammatic presentation of Construction 4.2.

6. For each interval $[a,b]$ in L, we take the limit $(L_{[a,b]}, \eta_{[a,b]})$ of $\mathrm{ev}_{[a,b]} \circ G : \mathrm{J} \to \mathrm{C}$, if it exists, and fail if not,
7. For each pair $[a,b] \supseteq [c,d]$ of intervals in L, we use naturality of ev_{-} to get cones $(L_{[a,b]}, \mathrm{ev}_{[a,b]\supseteq[c,d]} \circ \eta_{[a,b]})$ over $\mathrm{ev}_{[c,d]} \circ G$,
8. We get cone maps $L_{[a,b]\supseteq[c,d]} : L_{[a,b]} \to L_{[c,d]}$ via the u.p. of $(L_{[c,d]}, \eta_{[c,d]})$,
9. We assemble the above into a limit cone (L,η) with $L : [a,b] \mapsto L_{[a,b]}$, $([a,b] \supseteq [c,d]) \mapsto L_{[a,b]\supseteq[c,d]}$ and $(\eta_j)_{[a,b]} := (\eta_{[a,b]})_j$.
10. We return the cone $((L,L), \eta \circ \varepsilon)$ as a limit for F.

The end-to-end procedure is depicted in Figure 3.

Lemma 4.3. *For every finite diagram $F : \mathrm{J} \to \mathsf{L}(\mathrm{C})$, Construction 4.2 is well-defined.*

Proof. The limit at Step (1) exists because FPos is finitely complete [1, 12.6.1]. Since (L,ρ) is a cone, for every $h \in \mathrm{J}(j,j')$, we have $(Fh)^0 \circ \rho_j = \rho_{j'}$, hence Gh is well-defined. Moreover, since F and $[-]$ are functors, G respects identities, and since for $h \in \mathrm{J}(j,j')$ and $h' \in \mathrm{J}(j',j'')$, we have $(F(h' \circ h)^1)_{[\rho_j][a,b]} = (F(h') \circ [\rho_{j'}]) \circ Fh)^1)_{[\rho_j][a,b]}$, G respects composites and thus is a functor. The naturality condition for ε in Step (5) follows by unwinding definitions. Functoriality of L in Step (9) follows by uniqueness of the cone maps, and finally, naturality of η holds along J due to $(\eta_{-})_{[a,b]} := \eta_{[a,b]}$ being a cone, and along $[P]$ due to $L(- \supseteq -)$ being a map of cones. ☐

Proposition 4.4. *Let $f : P \to Q$ be a monotone function and* C *a category. Then* $- \circ [f] : \mathrm{Func}([Q],\mathrm{C}) \to \mathrm{Func}([P],\mathrm{C})$ *preserves all pointwise limits which exist in* $\mathrm{Func}([Q],\mathrm{C})$.

Proof. Let (L,η) be a pointwise limit for a diagram $G : \mathrm{J} \to L(Q)$, and let (K,χ) be a cone over $G \circ [f]$. We define a natural transformation $\gamma : K \to L \circ [f]$ by taking for each interval $[a,b]$ in P, the component $\gamma_{[a,b]}$ to be the unique map of cones $(K[a,b], \mathrm{ev}_{[a,b]}\chi) \to (Lf[a,b], \mathrm{ev}_{[f][a,b]}\eta)$, which is given by the universal property of the pointwise limit. Naturality and uniqueness of γ then follow both by uniqueness of the components. ☐

Theorem 4.5. *Let* C *be a category. If $F : \mathrm{J} \to \mathsf{L}(\mathrm{C})$ is a finite diagram, and Construction 4.2 succeeds for F, its output $((L,L), \varepsilon \circ \eta)$ is a limit for F.*

Proof. By Lemma 4.3, Construction 4.2 is well-defined. Let $((K,K),\gamma)$ be a cone over F. Since L is a limit for $U \circ F$, we have a unique map $k : K \to L$. Since (L,η) is a pointwise limit, by Proposition 4.4, $-\circ[k]$ preserves it, so we have a unique map of cones $\chi : K \to L$, so we take (k,χ) as our map. ☐

Corollary 4.6. *Let* J *be a finite category. If a category* C *has* J*-limits, then so does* $\mathsf{L}(\mathrm{C})$.

Proof. This is a known result about fibred categories, see e.g. [16, Thm.1], which we can extract as a consequence of Theorem 4.5. Since C has J-limits, so will each functor category Func$([P],\mathrm{C})$, as they inherit the pointwise limits from C. Hence Construction 4.2 will succeed for all diagrams $F : \mathrm{J} \to \mathsf{L}(\mathrm{C})$, and thus by Theorem 4.5 every such diagram has a limit. ☐

For a converse statement, we can prove that if C has an initial object, then our Construction 4.2 is searching for a limit in the correct fibre:

Proposition 4.7. *Let* C *be a category with an initial object* $0 \in \mathrm{C}$. *The fibration* $U : \mathsf{L}(\mathrm{C}) \to \mathrm{FPos}$ *preserves all existing limits.*

Proof. The functor U has a left-adjoint $F : \mathrm{FPos} \to \mathsf{L}(\mathrm{C})$, which acts as $P \mapsto (P, \Delta_0 : [P] \to \mathrm{C})$ and $f \mapsto (f, !)$, where Δ_0 is the constant functor on $0 \in \mathrm{C}$. ☐

The following lemma allows us to safely invoke completion arguments:

Lemma 4.8. *Let* $F : \mathrm{C} \to \mathrm{D}$ *be a functor. If* F *preserves, resp. reflects, all* J*-limits which exist in* C*, resp.* D*, then* $\mathsf{L}(F) : \mathsf{L}(\mathrm{C}) \to \mathsf{L}(\mathrm{D})$ *preserves, resp. reflects, all* J*-limits produced by Construction 4.2.*

Proof. For preservation, let $((L,L),\eta)$ be a limit for a diagram $G : \mathrm{J} \to \mathsf{L}(\mathrm{C})$ obtained via Construction 4.2. Then $\mathsf{L}(F)$ sends this limit to $((L, F \circ L), \varepsilon)$, where $\varepsilon := (\eta^0_j, F\eta^1_j)$. Since F preserves J-limits in C, this cone matches the output of Construction 4.2 for $\mathsf{L}(F) \circ G$, and thus by Theorem 4.5 is limiting.

For reflection, let $((L,L),\eta)$ be a cone over a diagram $G : \mathrm{J} \to \mathsf{L}(\mathrm{C})$ which is mapped under $\mathsf{L}(F)$ to the limit cone $((K,K),\varepsilon)$ obtained via Construction 4.2 on $\mathsf{L}(F) \circ G$. Since $\mathsf{L}(F)$ acts only on labels, we must have $L = K$. Moreover, since the limit (L,K) is pointwise and F reflects J-limits in D, $((L,L),\eta)$ satisfies the specification of Construction 4.2, and thus is a limit for G. ☐

4.2 Limits of Posetal Diagrams

Having presented a procedure for computing limits in $\mathsf{L}(\mathrm{C})$, we will conclude our technical exposition with a study of the corresponding limit procedure for the subcategory $\mathsf{P}(\mathrm{C})$, and extract some consequence for local diagrams. A lot of the heavy lifting of our results in this section hinges on the following lemma, which allows a translation of Construction 4.2 from labelled intervals to posetal diagrams.

Lemma 4.9. *The subcategory inclusion* $\mathrm{FDLat} \to \mathrm{FPos}$ *preserves finite limits.*

Proof. Let (L,η) be a limit cone for a finite diagram $F : \mathrm{J} \to \mathrm{FDLat}$. By the Birkhoff Representation Theorem [17, page 262], the functors $\mathrm{FPos}(-,2) : \mathrm{FPos}^{\mathrm{op}} \to \mathrm{FDLat}$ and $\mathrm{FDLat}(-,2) : \mathrm{FDLat}^{\mathrm{op}} \to \mathrm{FPos}$ form an adjoint equivalence, where 2 denotes the two-element distributive lattice $\{\bot \to \top\}$ and the hom-sets are equipped with their respective pointwise orders. Since $\mathrm{FDLat}(-,2)$ is a right adjoint, it preserves the limit (L,η), sending it to the colimit of $\mathrm{FDLat}(F-,2) : \mathrm{J}^{\mathrm{op}} \to \mathrm{FPos}$.

It suffices now to show that the composite of the dualising functor $\mathrm{FPos}(-,2)$ and subcategory inclusion $\mathrm{FDLat} \to \mathrm{FPos}$ preserves finite colimits. But this is just given by the internal contravariant hom $\mathrm{FPos}(-,2) : \mathrm{FPos}^{\mathrm{op}} \to \mathrm{FPos}$ for the Cartesian closed category FPos [1, 27.3.1]. By the enriched variant of the familiar continuity result [12, 3.29], this sends the colimit $(\mathrm{FDLat}(L,2), \mathrm{FDLat}(\eta,2))$ to the limit $(\mathrm{FPos}(\mathrm{FDLat}(L,2),2), \mathrm{FPos}(\mathrm{FDLat}(\eta,2),2))$ of the composite diagram $\mathsf{J} \to \mathrm{FPos}$, which, by the duality, is isomorphic to the image of the initial cone under the inclusion $\mathrm{FDLat} \to \mathrm{FPos}$. $\square$

Proposition 4.10. *If* C *has all equalisers, then* $\mathsf{P}(\mathsf{C})$ *is closed under equalisers taken in* $\mathsf{L}(\mathsf{C})$.

Proof. Let us identify $\mathsf{P}(\mathsf{C})$ with its image in $\mathsf{L}(\mathsf{C})$, and consider the parallel pair $(f,\alpha),(g,\beta) : (P,X) \to (Q,Y)$ of posetal maps. Let $(E, e : E \subseteq P)$ be the equaliser of f and g in FPos, where E is identified with its image in P under e, and (W,γ) be the equaliser of $\alpha_{[e]}$ and $\beta_{[e]}$ in $\mathrm{Func}([E],\mathsf{C})$. We wish to show W is local, so let $[a,a'] \supseteq [b,b'] \subseteq [c,c']$ be an atomic cospan of intervals in E with pullback $[d,d']$. By Lemma 4.9, e is a lattice homomorphism, so by Corollary 3.9, the image of the cospan under e is an atomic cospan in P with pullback $[d,d']$, and similarly for $[f][d,d']$ under $f \circ e = g \circ e$. Since X and Y are local, we are left to prove that the square $W([d,d'] \supseteq [a,a'], [c,c'] \supseteq [b,b'])$ of pointwise equalisers of two pullback squares is again a pullback. Our result hence follows either by general preservation of limits by limits, or by a diagram chase on the following:

$$\begin{tikzcd}
 & & Y[f][d,d'] \arrow[rrr] \arrow[ddd] \arrow[dr, phantom, "\lrcorner"] & & & Y[f][a,a'] \arrow[ddd] \\
 & X[d,d'] \arrow[ur, shift left] \arrow[ur, shift right] \arrow[rrr] \arrow[ddd] \arrow[dr, phantom, "\lrcorner"] & & & X[a,a'] \arrow[ur, shift left] \arrow[ur, shift right] \arrow[ddd] & \\
W[d,d'] \arrow[ur] \arrow[rrr] \arrow[ddd] & & & W[a,a'] \arrow[ur] \arrow[ddd] & & \\
 & & Y[f][c,c'] \arrow[rrr] & & & Y[f][b,b'] \\
 & X[c,c'] \arrow[ur, shift left] \arrow[ur, shift right] \arrow[rrr] & & & X[b,b'] \arrow[ur, shift left] \arrow[ur, shift right] & \\
W[c,c'] \arrow[ur] \arrow[rrr] & & & W[b,b'] \arrow[ur] & &
\end{tikzcd}$$

$\square$

Proposition 4.11. *Let* C *be a category with finite products. The subcategory* $\mathsf{P}(\mathsf{C})$ *is closed under finite products taken in* $\mathsf{L}(\mathsf{C})$.

Proof. Let $(P,X),(Q,Y)$ be posetal diagrams, and denote their product in $\mathsf{L}(\mathsf{C})$ by $(P \times Q, W)$. By Lemma 2.7, we have an isomorphism $[P \times Q] \cong [P] \times [Q]$, under which W acts by $([a,a'],[b,b']) \mapsto X[a,a'] \times Y[b,b']$. By Lemma 4.9, $P \times Q$ is a lattice, and thus by Corollary 3.9 the projections $[P \times Q] \to [P],[Q]$ preserve atomic cospans and their pullbacks. Since X,Y are local, so is W. Furthermore, the terminal object $(1,\Delta_1)$ is always local, so the result holds. $\square$

Corollary 4.12. *If* C *has all finite limits, then* $\mathsf{P}(\mathsf{C})$ *has all finite limits.*

Proof. By Proposition 4.10 and Proposition 4.11, $\mathsf{P}(\mathsf{C})$ is closed under arbitrary finite limits in $\mathsf{L}(\mathsf{C})$. But by Corollary 4.6, $\mathsf{L}(\mathsf{C})$ is finitely complete, hence so is $\mathsf{P}(\mathsf{C})$. $\square$

References

[1] Jiri Adamek, Horst Herrlich & George E. Strecker (1990): *Abstract and concrete categories: the joy of cats*. Pure and applied mathematics, Wiley, New York.

[2] Krzysztof Bar, Aleks Kissinger & Jamie Vicary (2018): *Globular: an online proof assistant for higher-dimensional rewriting*. Logical Methods in Computer Science 14(1), pp. 1–16, doi:10.23638/LMCS-14(1:8)2018. Publisher: Episciences. org.

[3] Krzysztof Bar & Jamie Vicary (2017): *Data structures for quasistrict higher categories*. Proceedings of the 32nd ACM/IEEE Symposium on Logic in Computer Science (LICS 2017), doi:10.1109/lics.2017.8005147. arXiv:1610.06908.

[4] Garrett Birkhoff (1937): *Rings of sets*. Duke Mathematical Journal 3(3), pp. 443–454, doi:10.1215/S0012-7094-37-00334-X. Publisher: Duke University Press.

[5] Simon Burton (2023): *String diagrams for higher mathematics with wiggle.py*. Presented at the 11th Symposium on Compositional Structures (SYCO), `https://www.youtube.com/watch?v=WLRXkFXaoAM`.

[6] Lukas Dixon & Ross Duncan (2008): *Graphical Reasoning in Compact Closed Categories for Quantum Computation*. Annals of Mathematics and Artificial Intelligence, doi:10.1007/s10472-009-9141-x. arXiv:0902.0514. Special Issue on Artificial Intelligence and Symbolic Computation.

[7] Giovanni de Felice, Alexis Toumi & Bob Coecke (2020): *DisCoPy: Monoidal Categories in Python*. Proceedings of the 3rd International Conference in Applied Category Theory (ACT 2020), doi:10.48550/arXiv.2005.02975. arXiv:2005.02975.

[8] Amar Hadzihasanovic & Diana Kessler (2023): *Higher-dimensional subdiagram matching*. Proceedings of the 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2023), doi:10.1109/LICS56636.2023.10175726. arXiv:2304.09216.

[9] Lukas Heidemann (2023): *Framed Combinatorial Topology with Labels in ∞-Categories*, doi:10.48550/arXiv.2305.06288. In preparation.

[10] Lukas Heidemann, David Reutter & Jamie Vicary (2022): *Zigzag normalization for associative n-categories*. Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2022), doi:10.1145/3531130.3533352. arXiv:2205.08952.

[11] Peter Johnstone & J Giraud (1999): *A Note on Discrete Conduché Fibrations*. Theory and Applications of Categories 5(1), pp. 1–11.

[12] Gregory Maxwell Kelly (1982): *Basic concepts of enriched category theory*. London Mathematical Society Lecture Notes 64, Cambridge University Press, Cambridge. Basic concepts of enriched category theory.

[13] David Reutter & Jamie Vicary (2019): *High-level methods for homotopy construction in associative n-categories*. Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2019), doi:10.1109/LICS.2019.8785895. arXiv:1902.03831.

[14] Emily Riehl (2016): *Category theory in context*. Aurora: Dover Modern Math Originals, Dover Publications, Mineola, New York.

[15] Pawel Sobocinski, Paul W. Wilson & Fabio Zanasi (2019): *CARTOGRAPHER: A Tool for String Diagrammatic Reasoning (Tool Paper)* 139, pp. 20:1–20:7. doi:10.4230/LIPIcs.CALCO.2019.20. Available at `http://drops.dagstuhl.de/opus/volltexte/2019/11448`.

[16] Andrzej Tarlecki, Rod M. Burstall & Joseph A. Goguen (1991): *Some fundamental algebraic tools for the semantics of computation: Part 3. indexed categories*. Theoretical Computer Science 91(2), pp. 239–264, doi:10.1016/0304-3975(91)90085-G. Available at `https://www.sciencedirect.com/science/article/pii/030439759190085G`.

[17] Gavin Wraith (1993): *Using the generic interval*. Cahiers de topologie et géométrie différentielle catégoriques 34(4), pp. 259–266.

String Diagrams with Factorized Densities

Eli Sennesh
Khoury College of Computer Science
Northeastern University
Boston, Massachusetts, United States of America
sennesh.e@northeastern.edu

Jan-Willem van de Meent
Amsterdam Machine Learning Lab (AMLab)
University of Amsterdam
Amsterdam, the Netherlands
j.w.vandemeent@uva.nl

A growing body of research on probabilistic programs and causal models has highlighted the need to reason compositionally about model classes that extend directed graphical models. Both probabilistic programs and causal models define a joint probability density over a set of random variables, and exhibit sparse structure that can be used to reason about causation and conditional independence. This work builds on recent work on Markov categories of probabilistic mappings to define a category whose morphisms combine a joint density, factorized over each sample space, with a deterministic mapping from samples to return values. This is a step towards closing the gap between recent category-theoretic descriptions of probability measures, and the operational definitions of factorized densities that are commonly employed in probabilistic programming and causal inference.

1 Introduction

Statisticians and machine learners analyze observed data by synthesizing models of those data. These models take a variety of forms, with several of the most widely used being directed graphical models, probabilistic programs, and structural causal models (SCMs). Applications of these frameworks have included cognitive modeling [7, 20], simulation-based inference [9], and model-based planning [12, 21]. Unfortunately, the richer the model class, the weaker the mathematical tools available to reason rigorously about it: SCMs built on linear equations with Gaussian noise admit easy inference, while graphical models have a clear meaning and a wide array of inference algorithms but encode a limited family of models. Probabilistic programs can encode any computably sampleable distribution, but the definition of their densities commonly relies on operational analogies with directed graphical models.

In recent years, category theorists have developed increasingly sophisticated ways to reason diagrammatically about a variety of complex systems. These include (co)parameterized categories of systems that may modify their parameters [5] and hierarchical string diagrams for rewriting higher-order computations [1]. Recent work on Markov categories of probabilistic mappings has provided denotational semantics to probabilistic programs [32, 18], abstract categorical descriptions of conditioning, disintegration, sufficient statistics, conditional independence [8, 13], and generalized causal models [14, 15].

This paper will take a step towards closing the gap between categorical probability and operational practice in probabilistic programming and applied Bayesian statistics. Denotational semantics for probabilistic programs define a measure over return values of a program given its inputs [32, 18]. To reason about inference methods, practitioners need to consider the joint distribution of internal random variables, as well as its density's factorization into conditionals. Section 2 will review basic definitions from probability and measure theory necessary to do so. Section 3 will then develop a category whose morphisms express joint (rather than marginal) distributions with factorized joint densities. Section 4 will show that generalized causal models can factorize these densities and admit interventional and counterfactual queries. Section 5 will work through a pair of examples and summarize the paper's developments.

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 260–278, doi:10.4204/EPTCS.397.16

Appendix A reviews the measure-theoretic concepts employed here; Appendix B reviews parametric and coparametric categories [5]; and Appendix C reviews free copy/delete and Markov categories.

Notation The notation $(\mathscr{C}, \otimes, I)$ will range over strict symmetric monoidal categories (SMC's for short). We denote composition as $g \circ f$ or equivalently as $f \mathbin{⨟} g$, write $(X^*, \odot, ())$ for the finite list monoid on X's, and overload $\otimes$ and $\oplus$ for direct products and sums. We draw string diagrams from the top (domain) to the bottom (codomain), showing products from left to right. Given a Markov category $\mathscr{C}$ we will draw deterministic maps in $\mathscr{C}_{det} \subset \mathscr{C}$ (which commute with **copy**) as rectangles and stochastic ones as ellipses/circles. We nest brackets with parentheses $([])$ equivalently.

2 Background: abstract and concrete categorical probability

This section will review the background on which the rest of the paper builds. Categorical probability begins from an abstract notion of nondeterminism: processes with a notion of "independent copies". Categorical probability then refines from a setting in which those nondeterministic processes "happen" whether observed or not, to a refined setting in which processes only "happen" when they affect an observed output. Categories of probability kernels, taking into account the details of measure theory (see Appendix A), will form a concrete instance of the abstract setting.

Definition 1 represents nondeterministic processes abstractly. A copy/delete category is an SMC whose morphisms generate information which can be copied or deleted freely.

Definition 1 (Copy/delete category). *A copy-delete or* CD-category *is an SMC $(\mathscr{C}, \otimes, I)$ in which every object $X \in Ob(\mathscr{C})$ has a commutative comonoid structure* $\mathbf{copy}_X : \mathscr{C}(X, X \otimes X)$ *and* $\mathbf{del}_X : \mathscr{C}(X, I)$ *which commutes with the monoidal product structure.*

Definition 2 then refines the abstract setting of CD categories to require that deleting the only result of a nondeterministic process is equivalent to deleting the process itself.

Definition 2 (Markov category). *A* Markov category *is a semicartesian CD-category $(\mathscr{C}, \otimes, I)$, so that the comonoidal counit is natural ($\forall f : \mathscr{C}(Z, X), f \mathbin{⨟} \mathbf{del}_X = I$) and makes $I \in Ob(\mathscr{C})$ a terminal object.*

Example 1 gives the canonical Markov category, consisting of measurable spaces and maps.

Example 1 (Measurable spaces and functions form a category [33]). *Measurable spaces and functions form a Cartesian category* **Meas** *with objects $(X, \Sigma_X) \in Ob(\mathbf{Meas})$ consisting of sets $X \in Ob(\mathbf{Set})$ and their σ-algebras*[1] *Σ_X and morphisms $\mathbf{Meas}((Z, \Sigma_Z), (X, \Sigma_X)) = \left\{ f \in X^Z \mid \forall \sigma_X \in \Sigma_X, f^{-1}(\sigma_X) \in \Sigma_Z \right\}$ consisting of measurable functions between measurable spaces.*

Meas acquires its Markov comonoid structure from its Cartesian structure. Definition 3 below provides the canonical Markov category for measure-theoretic probability.

Definition 3 (Category of measurable spaces and Markov kernels). *The category* $\mathbf{Stoch} = Kl(\mathbb{P})(\mathbf{Meas})$ *of measurable spaces and Markov kernels is the Kleisli category of the Giry monad [17] over* **Meas**, *having measurable spaces as objects and Markov kernels (Definition 19) between them as morphisms.*

Much of this paper will require a *strict* Markov category as in Definition 4 below.

Definition 4 (Strict Markov category). *A* strict *Markov category is one whose underlying SMC (with comonoid structure thrown away) is strict monoidal (its associator and unitors are identity).*

[1] Collections of "measurable subsets" closed under complements, countable unions, and countable intersections

Theorem 10.17 in Fritz [13] showed that every Markov category is comonoid equivalent to a strict one, licensing us to work with strictified Markov categories **Meas** and **Stoch** without further concern.

Unless otherwise mentioned, this paper will work with **Meas** and **Stoch** as strict, causal Markov categories[2]. When the ambient category and σ-algebra is clear from context, $f : Z \rightsquigarrow X$ will abbreviate $f : \mathbf{Stoch}((Z, \Sigma_Z), (X, \Sigma_X))$. In the concrete case of **Stoch**, measurable maps give the deterministic maps $\mathbf{Meas} \simeq \mathbf{Stoch}_{det} \subseteq \mathbf{Stoch}$. While Markov categories provide a compositional setting for nondeterministic processes, Markov kernels in these categories only provide probability measures for their outputs given their inputs. By design, they "forget" (i.e. marginalize over) all intermediate randomness in long chains of composition. Section 3 will build up a novel setting that "remembers" (i.e. does not marginalize over) joint distributions over all intermediate random variables through long chains of composition, and will show when there exist probability densities with respect to the joint distributions thus formed.

3 Joint distributions and densities for string diagrams

Statisticians cannot utilize input-output (parameter to distribution) mappings alone, except for maximum likelihood estimation. Instead, these typically appear as conditional probability distributions in a larger probability model. This larger model necessarily encodes a *joint distribution* over all relevant random variables, both those observed as data and the *latent* variables that give rise to observations. Practical probabilistic reasoning then consists of applying the laws of probability (product law for conjunctions, sum law for disjunctions, marginalization for unconditional events, Bayesian inversion) to numerical *densities* representing the joint distribution. This section will model the algebra of joint probability densities in a novel Markov category **Joint** constructed on the underlying Markov category **Stoch**.

Section 3.1 will first review an abstraction for categories in which morphisms act by "pushing forward" an internal "parameter space" and then instantiate that abstraction on a Markov category to yield a Markov category **Joint** of joint distributions. Section 3.2 will give the conditions for a concrete Markov kernel to admit a density. Section 3.3 will use those preliminaries to define a Markov category whose morphisms generate and push forward a joint probability density.

3.1 Accumulating random variables into joint distributions

Structural graphical models and probabilistic programs separate between the functions and variables they allow into deterministic and random ones [24]. Representing deterministic mechanisms categorically requires assuming that each nondeterministic process consists of a deterministic mechanism and a (potentially conditional) distribution over a random variable. This subsection will exploit "cybernetic" constructions (overviewed in Appendix B) for parameterization of deterministic mechanisms by random inputs and "writing out" of internal joint distributions as coparameters.

Proposition 1 will show the concrete category **Stoch** supports those constructions.

Proposition 1 (**Stoch** forms a symmetric monoidal $\mathscr{M}$-actegory[3])**.** *The concrete category* **Stoch** *forms a symmetric monoidal $\mathscr{M}$-actegory for $\mathscr{M} =$* **Stoch** *and $\mathscr{C} =$* **Stoch**.

Proof. Any SMC $\mathscr{C}$ forms a symmetric monoidal $\mathscr{M}$-actegory for $\mathscr{M} = \mathscr{C}$ with the product functor $\mathscr{M} \bullet \mathscr{C} = \mathscr{C} \times \mathscr{C}$ from the product category. Any Markov category is also an SMC, and so **Stoch** suffices. □

In this trivial case, Definition 25 simplifies so that Definition 5 will form an SMC.

[2]The latter property is shown in Example 11.35 of Fritz [13]

[3]Definition 24 in Appendix B

Definition 5 (Symmetric monoidal parametric categories)**.** *Given a strict SMC $(\mathscr{C}, \otimes, I)$, the symmetric monoidal* parametric (bi)category $\mathbf{Para}_{\otimes}(\mathscr{C})$ *has as objects those of $\mathscr{C}$ and as morphisms the pairs* $\mathbf{Para}_{\otimes}(\mathscr{C})(A,B) = \{(M,k) \in Ob(\mathscr{C}) \times \mathscr{C}(M \otimes A, B)\}$. *Composition for the two parameterized morphisms $(M,k) : \mathbf{Para}_{\otimes}(\mathscr{C})(A,B)$ and $(M',k') : \mathbf{Para}_{\otimes}(\mathscr{C})(B,C)$ consists of $(M' \otimes M, k' \circ (id_{M'} \otimes k))$; identities on objects A consist of (I, id_A); and $(\mathbf{Para}_{\otimes}(\mathscr{C}), \otimes, I)$ inherits its monoidal structure from $\mathscr{C}$*[4].

$\mathbf{Para}_{\otimes}(\mathbf{Stoch})$ will suffice for Definition 6 to model a Markov kernel over a joint distribution. The jointly random *residual* $(M, \Sigma_M) \in Ob(\mathbf{Stoch})$ will parameterize the deterministic map k.

Definition 6 (Joint Markov kernel)**.** *A* joint Markov kernel *is a pair of a Markov kernel with a deterministic mapping parameterized by that Markov kernel, up to permutation of residual components*

$$\mathbf{Joint}(Z,X) := \{(f,[M,k]) : \mathbf{Stoch}(Z,M) \times \mathbf{Para}_{\otimes}(\mathbf{Stoch}_{det})(Z,X)\}.$$

As implied by the hom-set notation above, joint Markov kernels will form a category of nondeterministic processes. Since the residuals of joint distributions only contribute to downstream processes through their local outputs, Theorem 1 will show this to be a copy/delete category.

Theorem 1 (Joint Markov kernels form a copy/delete category)**.** **Joint** *is a strict copy/delete category having $Ob(\mathbf{Joint}) = Ob(\mathbf{Stoch})$ and joint Markov kernels as morphisms.*

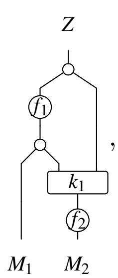
Proof. **Joint** must admit the typical requirements of a category as well as deterministic, copy-delete symmetric monoidal structure. We can demonstrate the necessary deterministic structure by exhibiting joint kernels $(I,k) : \mathbf{Para}_{\otimes}(\mathbf{Stoch}_{det})(Z,X) \implies ([I,k], \mathbf{del}_Z) : \mathbf{Joint}(Z,X)$ for any noiseless causal mechanism. Setting $k = \mathbf{copy}_X$ or $k = \mathbf{del}_Z$ yields the necessary copy and delete maps. Setting $k = \mathbf{swap}_{Z \otimes X}$ gives the necessary symmetry of the monoidal product. It remains to show that **Joint** has a monoidal product over morphisms and that its hom-sets are closed under composition.

Given two joint Markov kernels $(f_1,[M_1,k_1]) : \mathbf{Joint}(Z,X)$ and $(f_2,[M_2,k_2]) : \mathbf{Joint}(W,Y)$, their monoidal product is formed by pairing their causal mechanisms and noise distributions

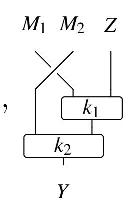
$$(f_1,[M_1,k_1]) \otimes (f_2,[M_2,k_2]) := (f_1 \otimes_{\mathbf{Stoch}} f_2, [M_1,k_1] \otimes_{\mathbf{Para}_{\otimes}(\mathbf{Stoch}_{det})} [M_2,k_2]) : \mathbf{Joint}(Z \otimes W, X \otimes Y).$$

Composing two joint Markov kernels $(f_1,[M_1,k_1]) : \mathbf{Joint}(Z,X)$ and $(f_2,[M_2,k_2]) : \mathbf{Joint}(X,Y)$ along their intermediate object involves composing their parametric maps and taking a conditional product of their stochastic kernels to form the composite joint distribution

$$(f_1,[M_1,k_1]) \fatsemi (f_2,[M_2,k_2]) := \Bigg($$



$$, \Bigg(M_1 \otimes M_2,$$



$$\Bigg)\Bigg) : \mathbf{Joint}(Z,Y). \qquad (1)$$

□

Anything called a *joint* Markov kernel ought to expose its internal joint distribution in a structured way. Definition 7 will link the composition of joint distributions to the cybernetics literature.

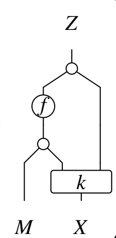
[4] see Proposition 6

Definition 7 (Symmetric monoidal coparametric categories[5]). *Given a strict SMC $(\mathscr{C}, \otimes, I)$, the symmetric monoidal* coparametric (bi)category $\mathbf{CoPara}_{\otimes}(\mathscr{C})$ *has as objects those of $\mathscr{C}$ and as morphisms the pairs* $\mathbf{CoPara}_{\otimes}(\mathscr{C})(A,B) = \{(M,k) \in Ob(\mathscr{C}) \times \mathscr{C}(A, M \otimes B)\}$ *of a residual object and a morphism from A to $M \otimes B$. Composition for the morphisms* $(M,k) : \mathbf{CoPara}_{\otimes}(\mathscr{C})(A,B)$ *and* $(M',k') : \mathbf{CoPara}_{\otimes}(\mathscr{C})(B,C)$ *consists of* $(M' \otimes M, (id_M \otimes k') \circ k))$*; identities on objects A consist of (I, id_A); and* $(\mathbf{CoPara}_{\otimes}(\mathscr{C}), \otimes, I)$ *inherits its monoidal structure from $\mathscr{C}$*[6].

Joint serves to work with joint distributions compositionally rather than marginalizing them out. Theorem 2 will show how mapping from $\mathbf{Joint} \to \mathbf{CoPara}_{\otimes}(\mathbf{Stoch})$ exposes the full joint distribution.

Theorem 2 (Joint Markov kernels coparameterize joint distributions). *There exists a full, identity-on-objects Markov functor* $[\![\cdot]\!] : \mathbf{Joint} \to \mathbf{CoPara}_{\otimes}(\mathbf{Stoch})$ *which maps the residual of a joint Markov kernel in* **Joint** *onto the residual of its image in* $\mathbf{CoPara}_{\otimes}(\mathbf{Stoch})$.

Proof. The required functor sends morphisms $[\![\cdot]\!] : \mathbf{Joint}(Z,X) \to \mathbf{CoPara}_{\otimes}(\mathbf{Stoch})(Z,X)$ to coparameterized Markov kernels whose codomain is the joint distribution over the residual and the output

$$[\![(f,[M,k])]\!] = \left(M, \right.$$


$$\left. \right).$$

This functor is trivially full, since any morphism $f : \mathbf{CoPara}_{\otimes}(\mathbf{Stoch})(Z,X)$ embeds trivially into **Joint** by setting the corresponding deterministic $k = id_{M \otimes X}$. It is not faithful: multiple "divisions of labor" between f and k can yield the same Markov kernel in $\mathbf{CoPara}_{\otimes}(\mathbf{Stoch})$. $\square$

Corollary 3 will give the trivial extension of marginalizing over the residual.

Corollary 3 (Marginalizing a joint Markov kernel's residual yields a Markov kernel). *There exists a full, identity-on-objects functor* $J : \mathbf{Joint} \to \mathbf{Stoch}$.

Proof. The required functor J just applies $[\![\cdot]\!]$ and then forgets the residual by composition with $\mathbf{del}_M$: its action on morphisms is $J((f,[M,k])) = [\![(f,[M,k])]\!] \mathbin{⨟} (\mathbf{del}_M \otimes id_X)$. $\square$

This subsection has considered arbitrary, unstructured joint distributions **Joint**. Section 3.2 will examine the special case in which the residual object is a standard Borel space and the conditional distribution into it meets the necessary conditions to admit a probability density.

3.2 Base measures and densities over standard Borel spaces

Applied probability typically works not with probability measures but with probability densities, functions over a finite-dimensional sample space giving the "derivative" of a probability measure at a point. However, probability densities only exist for measures that meet the conditions of the Radon-Nikodym Theorem, and only relative to a specified base measure over the sample space. This section will restrict

[5] See Definition 26 for the more general case

[6] See Proposition 6

the residual objects or internal noises of joint Markov kernels to standard Borel sample spaces admitting probability densities, and then show that this restriction still admits a broad class of joint Markov kernels.

Definition 8 provides a suitable ambient category for base measures.

Definition 8 (Category of measure spaces)**.** *The* category of measure spaces $\mathbb{M}$ *has as objects the measure spaces (X, Σ_X, μ) (Definition 22) and as morphisms the measure-preserving maps*

$$\mathbb{M}((Z,\Sigma_Z,\mu_Z),(X,\Sigma_X,\mu_X)) = \left\{ f : \mathbf{Meas}((Z,\Sigma_Z),(X,\Sigma_X)) \mid \forall \sigma_X \in \Sigma_X, \mu_Z(f^{-1}(\sigma_X)) = \mu_X(\sigma_X) \right\}.$$

Applications typically deal with probability densities over finite-dimensional Euclidean spaces and countable sets. In **Meas**, these can be characterized by the standard Borel spaces **Sbs** $\subset$ **Meas**, which are unique for each cardinality up to uncountability. Assigning these their canonical base measures will provide a suitable setting of measure spaces for characterizing densities.

However, the Radon-Nikodym Theorem requires that the sample space admit not only a measure but a σ-finite (Definition 20) base measure. Proposition 2 and Proposition 3 will therefore characterize the algebraic operations under which σ-finite measure spaces are closed. Proposition 2 below will characterize the base measures for joint probability densities.

Proposition 2 (σ-finite measure spaces have finite direct products)**.** *Let $I \in Ob(\mathbf{FinSet})$ be a set and let there be an I-indexed family of σ-finite measure spaces $(X_i, \Sigma_{X_i}, \mu_{X_i})_{i\in I} \in Ob(\mathbb{M})$. Then there exists a σ-finite* direct product *measure space $\bigotimes_{i\in I}(X_i, \Sigma_{X_i}, \mu_{X_i}) = (X, \Sigma_X, \mu_X)$.*

Proof. The product $\bigotimes_{i\in I}(X_i, \Sigma_i) \in Ob(\mathbf{Meas})$ exists thanks to **Meas** being Cartesian, so that the resulting set is that of Cartesian products and the σ-algebra is also that of Cartesian products. Letting π_i be the projection indexed by $i \in I$ of a Cartesian product, we write the σ-finite product measure (which exists and is unique when $(X_i, \Sigma_{X_i}, \mu_{X_i})$ are σ-finite [33][7]) as $\mu_X(\sigma_X) = \prod_{i\in I} \mu_{X_i}(\{\pi_i(x) : x \in \sigma_X\})$, yielding the direct product $(\bigotimes_{i\in I} X_i, \bigotimes_{i\in I} \Sigma_{X_i}, \mu_X) \in Ob(\mathbb{M})$. □

The reader can check that the direct product of measure spaces does not form a categorical product: the pairing required to witness the universal property will not be measure-preserving, with intervals of different lengths in the real line providing a counterexample.

Proposition 3 will then characterize the base measures for mixture probability densities.

Proposition 3 (σ-finite measure spaces have countable direct sums [11][8])**.** *Let $I \in Ob(\mathbf{Set})$ be a countable set and $(X_i, \Sigma_{X_i}, \mu_{X_i})_{i\in I} \in Ob(\mathbb{M})$ be a family of σ-finite measure spaces indexed by I. Then there exists a σ-finite* direct sum *measure space $\bigoplus_{i\in I}(X_i, \Sigma_{X_i}, \mu_{X_i}) \in Ob(\mathbb{M})$.*

Proof. The direct sum $\bigoplus_{i\in I}(X_i, \Sigma_{X_i}, \mu_{X_i}) = (X, \Sigma_X, \mu_X) \in Ob(\mathbb{M})$ of the indexed family consists of the set $X = \bigcup_{i\in I}(X_i \times \{i\})$, the σ-algebra $\Sigma_X = \{\sigma_X : \sigma_X \subseteq X, \forall i \in I, \{x : (x,i) \in \sigma_X\} \in \Sigma_{X_i}\}$, and the sum measure $\mu_X(\sigma_X) = \sum_{i\in I} \mu_{X_i}(\{x : (x,i) \in \sigma_X\})$. □

The reader can check that the direct sum of measure spaces does not form a categorical coproduct: the copairing required to witness the universal property will not be measure-preserving.

The above propositions characterized the algebra of σ-finite measure spaces, which thus now requires base cases. Restricting our attention to the standard Borel spaces, we can take the singleton set $(I, \mathscr{B}(I), \mu_{\#})$ equipped with the counting measure $\mu_{\#}$ and the real line $(\mathbb{R}, \mathscr{B}(\mathbb{R}), \lambda)$ with the Lebesgue measure as those base cases. An n-fold or countable direct sum of the singleton set gives finite and

[7] Definition 1.7.4, page 161

[8] 214L, page 38

countable discrete measure spaces, whose counting measure is σ-finite, while an n-fold product of the real line gives the Euclidean spaces, whose n-dimensional Lebesgue measures are σ-finite. Definition 9 will therefore formally give the class of measure spaces suitable for forming probability densities.

Definition 9 (σ-finite standard Borel measure space)**.** *The subcategory $\mathbb{M}_{\mathscr{B}} \subset \mathbb{M}$ restricts the category of measure spaces to the σ-finite* standard Borel *measure spaces freely generated by finite direct products $\otimes$ (Proposition 2) and countable direct sums $\oplus$ (Proposition 3) of the counting-measured singleton space $(\mathbb{1}, \mathscr{B}(\mathbb{1}), \mu_\#)$ and the Lebesgue-measured reals $(\mathbb{R}, \mathscr{B}(\mathbb{R}), \lambda)$.*

Definition 9 covers the most common sample spaces and their base measures, as instances of a more general construction assigning base measures to finite-dimensional manifolds as sample spaces for probability densities [27]. The above only allows finite products, since the product-of-Lebesgues measure on the Hilbert cube $\mathbb{R}^{\mathbb{N}}$ (via the Borel isomorphism $\mathbb{R} \simeq [0,1]$) fails to be σ-finite [2]. The rest of the paper will therefore work with measure spaces $\mathbb{M}_{\mathscr{B}}$, whose isomorphisms preserve base measures.

Having a class of measure spaces suitable for stating probability densities with respect to count, length, area, volume, etc., Definition 10 gives the class of Markov kernels which will admit densities.

Definition 10 (Density kernel)**.** *A standard Borel* density kernel *is a σ-finite (Definition 20) Markov kernel $f : Z \rightsquigarrow X$ whose codomain forms a σ-finite standard Borel measure space $(X, \Sigma_X, \mu_X) \in Ob(\mathbb{M}_{\mathscr{B}})$ and which is absolutely continuous $\forall z, f(z) \ll \mu_X$ with respect to the base measure μ_X*

$$\mathbf{Dens}((Z,\Sigma_Z),(X,\Sigma_X)) := \{(f,\mu_X) : (Z \rightsquigarrow X) \times \mathbb{M}(X) \mid (X,\Sigma_X,\mu_X) \in Ob(\mathbb{M}_{\mathscr{B}}), \forall z \in Z, f(z) \ll \mu_x\}.$$

Probability (and arbitrary measure) densities $p(x \mid z)$ also admit an alternative interpretation as measure kernels $Z \times X \times \Sigma_I \to [0,\infty]$ whose integration under the base measure yields the normalizing constant. Proposition 4 verifies that density kernels in fact admit probability densities.

Theorem 4 (Density kernels admit densities)**.** *Every density kernel $(f,\mu_X) : (Z \rightsquigarrow X) \times \mathbb{M}(X)$ (Definition 10) into a standard Borel measure space admits a density with respect to the base measure μ_X.*

Proof. σ-finiteness of the kernel f and the base measure μ_X, plus absolute continuity, give the necessary conditions for the classical Radon-Nikodym theorem: a Radon-Nikodym derivative therefore exists

$$\frac{df(z)}{d\mu_X} : \mathbf{Meas}(X, \mathbb{R}_{\geq 0}) \qquad\qquad f(z)(\sigma_X) = \int_{x \in \sigma_X} \frac{df(z)}{d\mu_X}(x)\ \mu_X(dx).$$

The Radon-Nikodym derivative is the measure-theoretic notion of a probability density function

$$\frac{df}{d\mu_X} : \mathbf{Meas}(Z \times X, \mathbb{R}_{\geq 0}), \qquad p_f(\cdot \mid \cdot) : \mathbf{Meas}(X \times Z, \mathbb{R}_{\geq 0}), \qquad p_f(x \mid z) := \frac{df(z)}{d\mu_X}(x).$$

The conditions on density kernels are therefore sufficient to yield probability densities. □

Despite the hom-set notation used for convenience, density kernels do not form a category: identity Markov kernels are Dirac delta measures that only admit densities in discrete spaces. They do, however, support all compositional structure under which the resulting base measure still indexes a standard Borel measure space. Definition 11 lays the foundation for this structure.

Definition 11 (Precomposition of a density kernel)**.** *Given a density kernel $(f,\mu_X) : (Z \rightsquigarrow X) \times \mathbb{M}(X)$ and a Markov kernel $h : W \rightsquigarrow Z$, their* precomposition *is $(f,\mu_X) \circ_{\mathbf{Dens}} h = (f \circ_{\mathbf{Stoch}} h, \mu_X)$.*

The above precomposition gives a definition for the composition of two density kernels: given (f,μ_X) and (g,μ_Y) their composite will just be $(g \circ f, \mu_Y)$. The existence of precomposition supports a product and coproduct algebra of density kernels, as expected based on the probability algebra itself.

Theorem 5 (Density kernels admit products and coproducts)**.** *Density kernels have products $(f,\mu_X)\otimes(g,\mu_Y)$ and coproducts $(f,\mu_X)\oplus(g,\mu_Y)$, witnessed by a pairing and copairing.*

Proof. Any two density kernels $(f,\mu_X):\mathbf{Dens}((Z,\Sigma_Z),(X,\Sigma_X))$ and $(g,\mu_Y):\mathbf{Dens}((Z,\Sigma_Z),(Y,\Sigma_Y))$ admit a pairing via precomposition with copying and the product measure space $(X,\Sigma_X,\mu_X)\otimes(Y,\Sigma_Y,\mu_Y)=(X\times Y,\Sigma_X\times\Sigma_Y,\mu_X\otimes\mu_Y)\in Ob(\mathbb{M}_{\mathscr{B}})$

$$(\mathbf{copy}_Z\,\mathbin{⨾}\,(f\otimes g),\mu_x\otimes\mu_Y):\mathbf{Dens}((Z,\Sigma_Z),(X,\Sigma_X)\otimes(Y,\Sigma_Y)).$$

Any two density kernels $(f,\mu_Y):\mathbf{Dens}((Z,\Sigma_Z),(Y,\Sigma_Y))$ and $(g,\mu_Y):\mathbf{Dens}((X,\Sigma_X),(Y,\Sigma_Y))$ also admit a copairing $\binom{(f,\mu_Y)}{(g,\mu_Y)}=\left(\binom{f}{g},\mu_Y\right)$ via the copairing of their Markov kernels in **Stoch**. □

The above theorems demonstrate that density kernels represent probability densities compositionally. However, density kernels do not admit post-composition with arbitrary Markov kernels. Section 3.3 will remedy this issue by applying density kernels to generate the residuals in joint Markov kernels.

3.3 Joint densities over joint distributions

Density kernels are not closed under pushforwards, and they do not form a category. **Joint** cannot apply directly to them. Definition 12 therefore gives an appropriate definition for joint density kernels.

Definition 12 (Joint density kernel)**.** *A* joint density kernel *between objects $Z,X\in Ob(\mathbf{Stoch})$ is a pair of a density kernel into $(M,\Sigma_M,\mu_M)\in Ob(\mathbb{M}_{\mathscr{B}})$ with a deterministic map parameterized by the residual*

$$\partial\mathbf{Joint}(Z,X):=\{((f,\mu_M),[M,k]):\mathbf{Dens}(Z,M)\times\mathbf{Para}_{\otimes}(\mathbf{Stoch}_{det})(Z,X)\mid(M,\Sigma_M,\mu_M)\in Ob(\mathbb{M}_{\mathscr{B}})\}.$$

Hom-set notation once again implies these kernels form a category, which in fact they will. First, Corollary 6 shows density kernels are closed under the joint distribution construction of Equation 1.

Corollary 6 (Density kernels admit joint distributions as conditional products)**.** *Given a density kernel $(f_1,\mu_{M_1}):\mathbf{Dens}(Z,M_1)$, a measurable map $k_1:\mathbf{Stoch}_{det}(Z\otimes M_1,X)$, and a density kernel $(f_2,\mu_{M_2}):\mathbf{Dens}(X,M_2)$, composing them according to the diagram in Equation 1 forms a joint density kernel*

$$\left(\mathbf{copy}_Z\,\mathbin{⨾}\,((\mathbf{copy}_{M_1}\circ f_1)\otimes id_Z)\,\mathbin{⨾}\,(id_{M_1}\otimes(f_2\circ k_1)),\mu_{M_1}\otimes\mu_{M_2}\right):\mathbf{Dens}(Z,M_1\otimes M_2).$$

Theorem 7 will show that joint density kernels form a category, and characterize them as joint Markov kernels with the extra data of a base measure on the residual.

Theorem 7 (Joint density kernels form a category)**.** *Joint density kernels ∂**Joint** form a wide subcategory of the restriction $\mathbf{Joint}_{\mathbf{BorelStoch}}$ of **Joint** to standard Borel Markov kernels in* **BorelStoch**.

Proof. First we show the joint density kernels form a subcategory, then show that subcategory is wide.

Corollary 6 shows that density kernels are closed under the composition of **Joint** (Equation 1), and so along with the obvious identity morphisms and associativity law they form a category. Theorem 5 shows that this category inherits the product and coproduct structure of **Joint**. The structure morphisms in **Joint** all have the unit I for their residual, which admits a trivial density as a finite standard Borel space; ∂**Joint** therefore inherits the copy/delete structure of **Joint**. This implies $\partial\mathbf{Joint}\subset\mathbf{Joint}_{\mathbf{BorelStoch}}$.

Objects and structure morphisms are inherited from **Joint**, so the subcategory is wide. □

The theorem above gives a copy/delete categorical structure for joint density kernels, whose base and probability measures will be σ-finite (Definition 20) as conditions for Radon-Nikodym. There is then a precise class of measures formed by pushing forward a σ-finite measure [34]: the s-finite measures (Definition 23). Proposition 4 shows that such s-finite measure kernels form a copy/delete category.

Proposition 4 (s-finite measure kernels form a CD-category [8][9]). *s-finite measure kernels (Definition 23) between measurable spaces form a CD-category* **sfKrn** *with* $Ob(\mathbf{sfKrn}) = Ob(\mathbf{Meas})$ *and hom-sets given by* $\mathbf{sfKrn}((Z,\Sigma_Z),(X,\Sigma_X)) = \{f : Z \times \Sigma_X \to [0,\infty] \mid \forall z, f(z) \textit{ is s-finite}\}$.

sfKrn only forms a copy/delete category, not a Markov category, since different measure kernels may have different normalizing constants, including an infinite normalizing constant. Corollary 8 shows that restricting to probability kernels forms a Markov category.

Corollary 8 (s-finite probability kernels form a Markov category). *The s-finite* probability *kernels* $f : \mathbf{sfKrn}((Z,\Sigma_Z),(X,\Sigma_X))$, *for which* $\forall z \in Z, f(z,X) = 1$, *form a Markov category* $\mathbf{sfStoch} \subset \mathbf{Stoch}$.

Proof. The restriction of all kernels to normalize to measure 1 renders every map $\mathbf{del}_Z$ unique, making I a terminal object and the resulting subcategory **sfStoch** a Markov category. □

Having a categorical setting capturing the Markov kernels used in computable applications, the remainder of this paper will interpret morphisms in $\partial\mathbf{Joint}$ into s-finite Markov kernels $\mathbf{sfStoch}(Z,X)$ with densities $\mathbf{sfKrn}(Z \otimes X, I)$. Theorem 9 shows that the joint Markov kernels of $\partial\mathbf{Joint}$ are s-finite and admit densities jointly measurable in the parameter and the residual.

Theorem 9 (Joint density kernels give s-finite probability kernels and densities). *Joint density kernels* $(f,[M,k]) : \partial\mathbf{Joint}((Z,\Sigma_Z),(X,\Sigma_X))$ *admit probability kernels* $p : \mathbf{sfStoch}((Z,\Sigma_Z),(X,\Sigma_X))$ *marginalizing out their randomness and probability densities* $p_f(\cdot \mid \cdot) : \mathbf{sfKrn}((Z,\Sigma_Z) \otimes (M,\Sigma_M), I)$.

Proof. Any density kernel $f : \mathbf{Dens}((Z,\Sigma_Z),(M,\Sigma_M))$ gives a σ-finite probability measure and any $(M,k) : \mathbf{Para}_{\otimes}(\mathbf{Stoch}_{det})(Z,X)$ pushes it forward. Every pushforward of a σ-finite Markov kernel is s-finite (Proposition 5), so $\partial\mathbf{Joint}$ consists entirely of s-finite joint Markov kernels. Being s-finite, joint density kernels admit the required probability kernels $p : \mathbf{sfStoch}((Z,\Sigma_Z),(X,\Sigma_X))$ with $p(z,\sigma_X) = f(z,k(z)^{-1}(\sigma_X))$ and densities $p_f(\cdot \mid z) : M \times \Sigma_I \to [0,\infty]$ measurable in z and m. Proposition 4 defines these as the Radon-Nikodym derivative $p_f(m \mid z)(\{*\}) = \frac{df(z)}{d\mu_M}(m)$. □

Theorem 7 and Theorem 9 finally gives a desirable categorical setting: one which supports composition, products, and coproducts as a copy/delete category should, while decomposing into a deterministic causal mechanism applied to a random variable with a joint density as a structural causal model should. Section 4 will put together the machinery in this section with existing work on factorizing string diagrams syntactically to interpret those factorizations as generalizing directed graphical models.

4 Diagrams as causal factorizations of joint distributions and densities

This section demonstrates that string diagrams with factorized densities support the full "ladder of causation" [25] as probabilistic models: factorized distributions, interventions, and counterfactual queries. Section 3 presented the $\partial\mathbf{Joint}$ construction for building up joint densities while still expressing arbitrary pushforward measures over them. Reasoning about directed graphical models or probabilistic programs compositionally requires providing a graphical syntax interpretable into $\partial\mathbf{Joint}$. Recent work [14, 15]

[9] Example 7.2

treated a combinatorial syntax of string diagrams as generalized causal models. This section first reviews the definitions of a generalized causal model and its factorization of a Markov kernel, then applies that syntax to this paper's novel constructions. Doing so will enable show that via generalized causal models, joint density kernels admit factorization of their densities (Theorem 10), interventional distributions (Theorem 11), and counterfactual distributions (Theorem 12).

Generalized causal models [14] provide several advantages over causal Bayesian networks as a representation of causal structure in probability models. They allow for global inputs to and outputs from a causal model, making explicit the interface necessary to reason compositionally about causal structures. It also makes explicit the grouping of "nodes" (in the underlying graph or hypergraph) into Markov kernels, clarifying how the joint distribution decomposes into random variables and causal mechanisms.

Definition 13 will now describe a generalized causal model.

Definition 13 (Generalized causal model [15])**.** *A generalized causal model φ over $\Sigma \in$ **FinHyp**[10] is a string diagram $p \to \mathrm{dom}(\tau) \leftarrow q : \mathbf{FreeMarkov}_{\Sigma}(n,m)$ for $n,m \in \mathbb{N}$ with a bijection q on wires.*

Any generalized causal model $p \to \mathrm{dom}(\tau) \leftarrow q$ is equivalent to a morphism [14]

$$\varphi : \mathbf{FreeMarkov}_{\Sigma}\left(\bigotimes_{i=1}^{n} \tau(p(i)), \bigotimes_{j=1}^{m} \tau(q(j))\right).$$

Definition 14 will capture factorization of a Markov kernel by a generalized causal model; Fritz and Klinger [14] called it causal compatibility in their Definition 11.

Definition 14 (Factorization of a Markov kernel by a causal model [14])**.** *A factorization (f,φ,F) in* **Stoch** *consists of a morphism with decomposed domain and codomain $f : \mathbf{Stoch}\left(\bigotimes_{i=1}^{n} D_i, \bigotimes_{j=1}^{m} C_j\right)$, a causal model $\varphi : \mathbf{FreeMarkov}_{\Sigma}(n,m)$, and a strict Markov functor $F : \mathbf{FreeMarkov}_{\Sigma} \to \mathbf{Stoch}$ such that $f = F(\varphi)$, $\forall i \in [1..n], D_i = F(\mathrm{dom}(\varphi)_i)$, and $\forall j \in [1..m], C_j = F(\mathrm{cod}(\varphi)_j)$.*

The joint density kernels $\partial\mathbf{Joint}(Z,X)$ have an important difference from the simple Markov kernels factorized by generalized causal models in Definition 14: the density to factorize is not over $x \in X$ but over the extra structure of the residual $m \in M$. This subsection will show how to add this extra structure to a factorization, then show how to access that structure to show that generalized causal models over joint density kernels support causal inference as such: interventions and counterfactual reasoning.

Definition 15 will require a factorization to label each box's residual to apply to joint Markov kernels.

Definition 15 (Joint factorization functor)**.** *A joint factorization functor for a signature $\Sigma \in$ **FinHyp** is a labeling of boxes with residual wires $r : B(\Sigma) \to W(\Sigma)^*$ and a strict Markov functor $F : \mathbf{FreeMarkov}_{\Sigma} \to$ **Joint** respecting $\forall b \in B(\Sigma), F(b) = ([\bigotimes_{w \in r(b)} F(w), k], f) : \mathbf{Joint}(F(\mathrm{dom}(b)), F(\mathrm{cod}(b)))$.*

Joint factorizations label residuals in the signature and also map to joint density kernels. Theorem 10 shows they factorize the implied joint density of a causal model.

Theorem 10 (Joint density kernels admit factorized densities)**.** *Given a signature $\Sigma \in$ **FinHyp**, a strict Markov functor $F : \mathbf{FreeMarkov}_{\Sigma} \to \partial\mathbf{Joint}$ gives a joint density $p_f(\cdot \mid \cdot \in F(\mathrm{dom}(\varphi)))$ for every causal model $\varphi : \mathbf{FreeMarkov}_{\Sigma}(n,m)$.*

Proof. Definition 15 requires for any sub-diagram $\varphi' \subseteq \varphi$ there will be some $F(\varphi') = (f,[M,k])$. Theorem 9 then gives a density over the residual, while the functoriality of F and Corollary 6 together imply that products of individual joint-densities yield the complete joint density. □

[10] see Appendix C

Theorem 11 then shows that by assigning boxes optional points in their codomains, joint factorizations also admit interventional distributions.

Theorem 11 (Joint factorizations admit interventional distributions)**.** *Consider a joint factorization* (f, φ, F) *over a signature* Σ*. Then any intervention* $\mathbf{do} : \prod_{b:B(\Sigma)} I \oplus \mathscr{C}_{det}(I, F(\mathrm{cod}(b))$ *induces a functor* $\mathrm{Int} : \mathbf{FreeMarkov}_{\Sigma} \to \mathbf{Joint}$ *and an interventional distribution* $\mathrm{Int}(\varphi)$*.*

Proof. Any single-box free string diagram has an image $F(\langle b \rangle)$. We define the required functor Int : $\mathbf{FreeMarkov}_{\Sigma} \to \mathbf{Joint}$ by extension of a hypergraph morphism $\alpha : \Sigma \to \mathrm{hyp}(\mathbf{Joint})$ following Fritz and Liang [15] (see their Remark 4.3). α will be identity on wires and intervene on boxes

$$\alpha(b) : B(\Sigma) \to B(\mathrm{hyp}(\mathbf{Joint}))$$
$$\alpha(b) = \begin{cases} \mathrm{hyp}(([I, \mathbf{del}_{\mathrm{dom}(b)}], \mathbf{del}_{\mathrm{dom}(b)} \fatsemi x)) & \mathbf{do}(b) = \mathbf{inr}(x) \\ \mathrm{hyp}(F(\langle b \rangle)) & \mathbf{do}(b) = \mathbf{inl}(I) \end{cases}. \qquad \square$$

Finally, Theorem 12 employs similar reasoning to model counterfactual queries over jointly factorized causal models, given fixed values for random variables and an intervention.

Theorem 12 (Joint factorizations give counterfactuals)**.** *Consider a signature* $\Sigma \in \mathbf{FinHyp}$ *and a joint factorization* (f, φ, F)*. Then any intervention* $\mathbf{do} : \prod_{b:B(\Sigma)} I \oplus \mathscr{C}_{det}(I, F(\mathrm{cod}(b))$ *and any assignment* $U : B(\Sigma) \to [0, 1]$ *of uniform random variates to boxes induces a functor* $\mathrm{If} : \mathbf{FreeMarkov}_{\Sigma} \to \mathbf{Joint}$ *and a counterfactual distribution* $\mathrm{If}(\varphi)$*.*

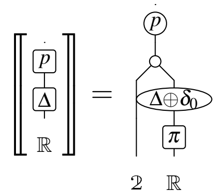
Proof. We work as above, but this time explicitly consider the structure of the image $F(\langle b \rangle) = (f, [M, k])$. f gives a standard Borel probability measure, so the Randomization Lemma [3] demonstrates equality of f with a pushforward $F(\langle b \rangle)_1 = f(\cdot) = g(\cdot, z)_*(U)(du)$ of the uniform distribution $U(du)$ by a deterministic map $g(\cdot, z)$. Our hypergraph morphism utilizes that fact

$$\alpha(b) = \begin{cases} \mathrm{hyp}(([I, \mathbf{del}_{\mathrm{dom}(b)}], \mathbf{del}_{\mathrm{dom}(b)} \fatsemi x)) & \mathbf{do}(b) = \mathbf{inr}(x) \\ \mathrm{hyp}(\delta_{U(b)}(g(b, \cdot))) & \mathbf{do}(b) = \mathbf{inl}(I) \end{cases}. \qquad \square$$

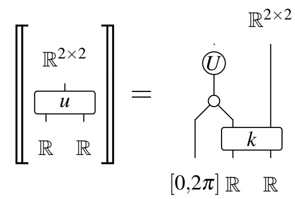
Together, Theorems 10, 11, and 12 demonstrate that joint density kernels, jointly factorized by a generalized causal model, support the properties that have made directed graphical models so widely useful. With these theorems as "sanity checks", Section 5 will summarize the paper's overall contributions, give some worked examples applying $\partial\mathbf{Joint}$, and discuss future work.

5 Discussion

This paper started from the existing work on copy/delete categories, Markov categories, and the factorization of morphisms in those categories by generalized causal models. From there, Section 3 constructed a novel Markov category **Joint** whose morphisms keep internal track of the joint distribution they denote, defined a subcategory $\partial\mathbf{Joint} \subset \mathbf{Joint}$ whose morphisms support only joint densities over standard Borel spaces as their internal distributions. Section 4 then demonstrated that **Joint** supports factorization by generalized causal models, that these factorize joint densities $\partial\mathbf{Joint}$, and that these support the interventional and counterfactual reasoning necessary for causal inference. This section will discuss some short worked examples of using $\partial\mathbf{Joint}$ for real probability models (Section 5.1), and then move on to speculate what future work could spring from the paper's developments (Section 5.2).



(a) The wiring diagram in ∂**Joint** of a mixture model between a delta and a Gaussian, and its image in **Stoch** with a coproduct projection



(b) A Markov kernel in ∂**Joint** projecting a sample from the uniform circle through a linear transformation, and its image in **Stoch**

Figure 1: Example joint density kernels (Definition 12): a mixture model between a constant and a Gaussian distribution depending on a coin flip (*left*) and a Markov kernel projecting a random angle onto a parametrically skewed ellipse (*right*). The $[\![\cdot]\!]$ functor (Corollary 3) maps into **Stoch**.

5.1 Worked examples

The previous sections have focused on formalism. Section 3 defined a Markov category ∂**Joint** of joint density kernels in **Stoch** (rather than the typical restriction to **FinStoch**) whose residuals (by construction) admit probability densities. Section 4 then established that the generalized causal models recently described in the categorical probability literature can indeed apply to ∂**Joint** morphisms, factorizing their joint densities and providing for causal reasoning. This subsection will apply the ∂**Joint** formalism to the models shown in Figure 1, taken from Wu et al [36] and Radul and Alexeev [27].

Figure 1a shows a generative model in which we detect fake coins by placing an even number of coins on a well-calibrated balance. The presence of a fake coin, whose weight deviates from the others, will tip the balance away from the neutral position. p determines whether the a fake coin is present, which in turn determines whether the balance position is distributed according to a Gaussian $\Delta \sim \mathcal{N}(1, 0.5)$ or according to a Dirac measure $\Delta \sim \delta_0$. The joint distribution shown on the right-hand side of the equation admits a density with respect to the standard Borel measure space $(\mathbb{2}, \mathscr{B}(\mathbb{2}), \mu_{\#}) \otimes ((\mathbb{R}, \mathscr{B}(\mathbb{R}), \lambda) \oplus (\mathbb{1}, \mathscr{B}(\mathbb{1}), \mu_{\#}))$, whereas the marginal on $\mathbb{R}$ lacks a density for the Lebesgue measure λ.

Figure 1b shows the example from Radul and Alexeev [27] in which a sample from $U(0, 2\pi)$ is projected onto a non-isotropic ellipse. Those authors calculate a probability density on the ellipse via the projection's Jacobian. Figure 1b shows the two components of a ∂**Joint** morphism: how the uniformly random angle U and a linear transformation $\mathbb{R}^{2\times 2}$ parameterize the the geometric projection k. The equation shows how $[\![\cdot]\!]$ maps the single box in ∂**Joint** (left) to the Markov kernel in **Stoch** (right).

The two examples in Figure 1 both show how the ∂**Joint** construction can compactly encode complex, parameterized joint probability densities linked by deterministic causal mechanisms. Section 5.2 will discuss potential future work extending this paper's construction and conclude.

5.2 Future work and conclusion

This paper's mathematical constructions could generalize or be strengthened in a number of ways. It would be desirable to obtain a category in which Markov kernels admit common-sense densities without having to separate into a density over a standard Borel space and a pushforward through a deterministic map; the Lebesgue decomposition of arbitrary measures into mutually singular absolutely-continuous, diffuse, and atomic portions suggests a possible route to that goal. Up to a normalization constant, every reference measure in $\mathbb{M}$ is a Hausdorff measure. This suggests densities could be obtained by considering manifolds, standardizing on the Hausdorff measure as Radul and Alexeev [27] suggest, and then defining density kernels on that foundation. Finally, Definition 9 forms an endofunctor in the category of measure spaces whose algebras and coalgebras may prove of interest. For example, recent work by Dash [10] explored defining probability measures on quasi-Borel spaces as pushforwards of a uniform distribution on the Hilbert cube, an element of the endofunctor's terminal coalgebra.

Future work can go in a number of directions to unify the formalisms of applied probabilistic reasoning. Instantiating this paper's constructions in a Markov category in which all randomness arises from an independent noise source would transform any causal factorization of a joint (density) kernel into a structural causal model [24], unifying causal Bayes nets with structural equation models. In the application area of probabilistic programming, this paper has only described "first-order" probabilistic programming languages lacking general stochastic recursion [22], corresponding to non-closed Markov categories. A combinatorial syntax for hierarchical string diagrams [1] would extend our reasoning in this paper to the closed Markov categories such as **QBS** [18] that provide denotations for higher-order probabilistic programming languages. We intend to extend this paper's formalism to categorify Sequential Monte Carlo methods [23] for generalized causal models of unnormalized distributions. We aim to apply the ∂**Joint** construction alongside recent work on unique name generation [28] to model heterogeneous tracing in probabilistic programming. Recent work on free string diagrams [35] has also suggested ways to map from free string diagrams to free diagrams of optics; equipping joint density kernels with optic structure would follow up on the work of Smithe [31] and Schauer [29].

Acknowledgements We would like to thank the anonymous reviewers for their feedback and advice in refining the paper for camera-ready. We would also like to thank the ACT 2023 program chairs for their careful shepherding of the review process. We thank Tobias Fritz, Luke Ong, Sam Staton, and Matthijs Vákár for laying the categorical foundations of s-finite Markov kernels. Finally, we would like to extensively thank Alex Lew for early discussions and cooperation on preliminary work to this paper. Eli Sennesh was supported by NSF award 2047253.

References

[1] Mario Alvarez-picallo, Dan Ghica, David Sprunger & Fabio Zanasi (2022): *Rewriting for Monoidal Closed Categories*. In: *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, 228, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany, pp. 29:1–29:0, doi:10.4230/LIPIcs.FSCD.2022.29.

[2] Richard Baker (1991): *"Lebesgue measure" on $\mathbb{R}^\infty$*. *Proceedings of the American Mathematical Society* 113(4), pp. 1023–1029, doi:10.2307/2048779.

[3] V. I. Bogachev (2007): *Measure theory*. Springer, Berlin; New York, doi:10.1007/978-3-540-34514-5.

[4] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński & Fabio Zanasi (2016): *Rewriting modulo symmetric monoidal structure*. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, ACM, New York NY USA, p. 710–719, doi:10.1145/2933575.2935316. Available at https://dl.acm.org/doi/10.1145/2933575.2935316.

[5] Matteo Capucci, Bruno Gavranović, Jules Hedges & Eigil Fjeldgren Rischel (2021): *Towards foundations of categorical cybernetics*. In: *Applied Category Theory Conference (ACT 2021)*, EPTCS, pp. 235–248. Available at http://arxiv.org/abs/2105.06332.

[6] Matteo Capucci & Bruno Gavranović (2022): *Actegories for the Working Amthematician*.

[7] Nick Chater, Joshua B Tenenbaum & Alan Yuille (2006): *Probabilistic models of cognition: Conceptual foundations*. Trends in cognitive sciences 10(7), pp. 287–291, doi:10.1016/j.tics.2006.05.008.

[8] Kenta Cho & Bart Jacobs (2019): *Disintegration and Bayesian inversion via string diagrams*. Mathematical Structures in Computer Science 29(7), pp. 938–971, doi:10.1017/S0960129518000488.

[9] Kyle Cranmer, Johann Brehmer & Gilles Louppe (2020): *The frontier of simulation-based inference*. Proceedings of the National Academy of Sciences 117(48), pp. 30055–30062, doi:10.1073/pnas.1912789117.

[10] Swaraj Dash, Younesse Kaddar, Hugo Paquet & Sam Staton (2023): *Affine monads and lazy structures for bayesian programming*. Proceedings of the ACM on Programming Languages 7(POPL), pp. 1338–1368, doi:10.1145/3571239.

[11] David H. Fremlin (2010): *Measure theory. 2: Broad foundations*, 2. ed edition. Torres Fremlin, Colchester.

[12] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck & Giovanni Pezzulo (2017): *Active inference: a process theory*. Neural computation 29(1), pp. 1–49, doi:10.1162/NECO_a_00912.

[13] 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.

[14] Tobias Fritz & Andreas Klingler (2023): *The d-Separation Criterion in Categorical Probability*. Journal of Machine Learning Research 24(46), pp. 1–49.

[15] Tobias Fritz & Wendong Liang (2023): *Free gs-Monoidal Categories and Free Markov Categories*. Applied Categorical Structures 31(2), p. 21, doi:10.1007/s10485-023-09717-0.

[16] Giorgio Gallo, Giustino Longo, Stefano Pallottino & Sang Nguyen (1993): *Directed hypergraphs and applications*. Discrete Applied Mathematics 42(2–3), p. 177–201, doi:10.1016/0166-218X(93)90045-P.

[17] Michèle Giry (1982): *A categorical approach to probability theory*. In B. Banaschewski, editor: Categorical Aspects of Topology and Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, p. 68–85, doi:10.1007/BFb0092872.

[18] Chris Heunen, Ohad Kammar, Sam Staton & Hongseok Yang (2017): *A convenient category for higher-order probability theory*. In: Proceedings - Symposium on Logic in Computer Science, pp. 1–12, doi:10.1109/LICS.2017.8005137. ArXiv: 1701.02547 Citation Key: Heunen2017 ISSN: 10436871.

[19] Kiyosi Itô et al. (1984): *An Introduction to Probability Theory*. Cambridge University Press, doi:10.1017/9781139171809.

[20] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum & Samuel J Gershman (2017): *Building machines that learn and think like people*. Behavioral and brain sciences 40, p. e253, doi:10.1017/S0140525X16001837.

[21] Sergey Levine (2018): *Reinforcement learning and control as probabilistic inference: Tutorial and review*. arXiv preprint arXiv:1805.00909.

[22] Jan-Willem van de Meent, Brooks Paige, Hongseok Yang & Frank Wood (2018): *An introduction to probabilistic programming*. arXiv preprint arXiv:1809.10756.

[23] Christian A Naesseth, Fredrik Lindsten & Thomas B Schon (2019): *Elements of Sequential Monte Carlo*. Foundations and Trends in Machine Learning 12(3), pp. 187–306, doi:10.1561/2200000074.

[24] Judea Pearl (2012): *The causal foundations of structural equation modeling*. Handbook of structural equation modeling, pp. 68–91.

[25] Judea Pearl & Dana Mackenzie (2018): *The book of why: the new science of cause and effect*. Basic books.

[26] Paolo Perrone (2019): *Notes on Category Theory with examples from basic mathematics*. arXiv preprint arXiv:1912.10642.

[27] Alexey Radul & Boris Alexeev (2021): *The Base Measure Problem and its Solution*. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, 130, Proceedings of Machine Learning Research, San Diego, California, p. 3583–3591.

[28] Marcin Sabok, Sam Staton, Dario Stein & Michael Wolman (2021): *Probabilistic programming semantics for name generation*. Proceedings of the ACM on Programming Languages 5(POPL), pp. 1–29, doi:10.1145/3434292.

[29] Moritz Schauer & Frank van der Meulen (2023): *Compositionality in algorithms for smoothing*. arXiv preprint arXiv:2303.13865.

[30] Adam Ścibior, Ohad Kammar, Matthijs Vákár, Sam Staton, Hongseok Yang, Yufei Cai, Klaus Ostermann, Sean K. Moss, Chris Heunen & Zoubin Ghahramani (2017): *Denotational Validation of Higher-Order Bayesian Inference*. Proc. ACM Program. Lang. 2(POPL), doi:10.1145/3158148.

[31] Toby St Clere Smithe (2020): *Bayesian updates compose optically*. arXiv preprint arXiv:2006.01631.

[32] Sam Staton (2017): *Commutative Semantics for Probabilistic Programming*, p. 855–879. Lecture Notes in Computer Science 10201, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-662-54434-1_32. Available at https://link.springer.com/10.1007/978-3-662-54434-1_32.

[33] Terence Tao (2011): *An introduction to measure theory*. Graduate studies in mathematics 126, American Mathematical Society, Providence, R.I, doi:10.1090/gsm/126/02.

[34] Matthijs Vákár & Luke Ong (2018): *On S-Finite Measures and Kernels*. Available at http://arxiv.org/abs/1810.01837. ArXiv:1810.01837 [math].

[35] Paul Wilson & Fabio Zanasi (2023): *Data-Parallel Algorithms for String Diagrams*. arXiv:2305.01041.

[36] Yi Wu, Siddharth Srivastava, Nicholas Hay, Simon Du & Stuart Russell (2018): *Discrete-Continuous Mixtures in Probabilistic Programming: Generalized Semantics and Inference Algorithms*. In: Proceedings of the 35th International Conference on Machine Learning, PMLR, p. 5343–5352. Available at https://proceedings.mlr.press/v80/wu18f.html.

A Measure theory background

Measure theory studies ways of assigning a “size” to a set (beyond its cardinality); these can include count, length, volume, and probability. Definition 16 begins with a nice class of measurable spaces.

Definition 16 (Standard Borel space)**.** *Let* $(X, T_X) \in Ob(\mathbf{Top})$ *be a separable complete metric space or homeomorphic to one. Equipping* X *with its Borel* σ*-algebra* $\mathscr{B}(X)$ *generated by complements, countable unions, and countable intersections of open subsets* $U \in T$ *yields a* standard Borel space $(X, \mathscr{B}(X)) \in Ob(\mathbf{Sbs})$*, which is also a measurable space since* $\mathbf{Sbs} \subset \mathbf{Meas}$.

The paper uses standard Borel spaces as a basis for its category of measure spaces (Definition 9). Example 2 is such a space.

Example 2 (The unit interval)**.** *The closed unit interval* $[0,1]$ *with its Borel* σ*-algebra of open sets* $\mathscr{B}(0,1)$ *forms a standard Borel space* $([0,1], \mathscr{B}(0,1))$.

Having a category of measurable spaces and some nice examples, Definition 17 formally defines what it means to assign a “size” to a measurable set.

Definition 17 (Measure)**.** *A* measure $\mu : \mathbb{M}(Z)$ *on a measurable space* $(Z, \Sigma_Z) \in Ob(\mathbf{Meas})$ *is a function* $\mu : \Sigma_Z \to [0, \infty]$ *that is null on the empty set (*$\mu(\emptyset) = 0$*) and countably additive over pairwise disjoint sets*

$$\frac{\{\sigma_k \in \Sigma_Z\}_{k \in \mathbb{N}} \qquad \forall k \in \mathbb{N}, n \in \mathbb{N}, n \neq k \implies \sigma_k \cap \sigma_n = \emptyset}{\mu\left(\bigcup_{k \in \mathbb{N}} \sigma_k\right) = \sum_{k \in \mathbb{N}} \mu(\sigma_k)}$$

Reasoning compositionally about measure requires a class of maps between a domain and a codomain that form measures. The Giry monad [17] sends a measurable space (X, Σ_X) to its space of measures $\mathbb{M}(X)$ and probability measures $\mathbb{P}(X) \subset \mathbb{M}(X)$. Definition 18 defines maps into those spaces, treating the domain as a parameter space for a measure over the codomain.

Definition 18 (Measure kernel)**.** *A* measure kernel *between two measurable spaces* $(Z, \Sigma_Z), (X, \Sigma_X) \in Ob(\mathbf{Meas})$ *is a function* $f : Z \times \Sigma_X \to [0, \infty]$ *such that* $\forall z \in Z, f(z, \cdot) : \mathbb{M}(X)$ *is a measure and* $\forall \sigma_X \in \Sigma_X, f(\cdot, \sigma_X) : \mathbf{Meas}((Z, \Sigma_Z), ([0, \infty], \Sigma_{[0,\infty]}))$ *is measurable.*

Measure kernels serve both to define Markov kernels below, and to form a broader class of copy/delete categories, which in Theorem 9 are seen to admit probability densities as morphisms. Definition 19 specializes to measure kernels yielding only normalized probability measures.

Definition 19 (Markov kernel)**.** *A* Markov kernel *is a measure kernel* $f : Z \times \Sigma_X \to [0, \infty]$ *whose measure is a probability measure so that* $\forall z \in Z, f(z, \cdot) : \mathbb{P}(X)$ *and* $\forall z \in Z, f(z, X) = 1$.

The Giry monad, restricted to probability spaces, yields Markov kernels as its Kleisli morphisms $\mathbf{Meas}((Z, \Sigma_Z), \mathbb{M}(X))$, forming the main category of Markov kernels in this paper (**Stoch**, Definition 3). Describing densities categorically then requires invoking the Radon-Nikodym Theorem, which determines when probability measures have densities. The next two definitions give the Theorem’s conditions, which must be satisfied for a density to exist.

Definition 20 will formalize the condition that both the base measure and a probability measure consist of sums over countable partitions of the sample space.

Definition 20 (σ-finite measure kernel)**.** *A* σ-finite measure kernel $f : Z \times \Sigma_X \to [0, \infty]$ *is a measure kernel which at every parameter* $z \in Z$ *splits its codomain into countably many measurable sets* $X = \bigcup_{n \in \mathbb{N}} X_n \in \Sigma_X$*, each of which has finite measure* $f(z)(X_n) < \infty$.

Definition 21 will now formalize the further requirement that for a probability measure to admit a density function, it must have only the same null-sets as the underlying base measure.

Definition 21 (Absolute continuity)**.** *One σ-finite measure kernel $f : Z \times \Sigma_X \to [0,\infty]$ is* absolutely continuous *($f \ll g$) with respect to another σ-finite measure kernel over the same codomain $g : Y \times \Sigma_X \to [0,\infty]$ when $\forall z \in Z, y \in Y, \sigma_X \in \Sigma_X, g(y)(\sigma_X) = 0 \implies f(z)(\sigma_X) = 0$.*

The conditions in Definition 20 and Definition 21 are necessary and sufficient for the existence of a probability density via the Radon-Nikodym Theorem, as used in density kernels in Definition 10. Density kernels use measure *spaces* as their codomains: these group together the desired topology, dimensionality, and base measure. Definition 22 below formally defines measure spaces, which the paper uses in the specific form of standard Borel measure spaces (Definition 9).

Definition 22 (Measure space)**.** *A* measure space *is a pair $((X,\Sigma_X),\mu)$ of a measurable space $(X,\Sigma_X) \in Ob(\mathbf{Meas})$ with a measure $\mu : \mathbb{M}(X)$ on that space.*

The measure spaces just defined form objects in a category which Definition 8 describes. Passing from the category of measurable spaces **Meas** to the category of measure spaces $\mathscr{M}$ requires the resulting morphisms to respect the chosen measure, so that measurable sets do not "grow" or "shrink".

Having given the conditions for densities to exist, the paper passes from density kernels to joint density kernels. Definition 23 will give a class of Markov kernels encompassing all those in this paper, particularly joint density kernels.

Definition 23 (s-finite measure kernel)**.** *An s-*finite measure kernel *$f : Z \times \Sigma_X \to [0,\infty]$ is a measure kernel (as in Definition 18 above) which decomposes into a sum of finite kernels $f = \sum_{n\in\mathbb{N}} f_n$ such that $\forall n \in \mathbb{N}, f_n : Z \times \Sigma_X \to [0,\infty]$ and $\forall n \in \mathbb{N}, \exists r_n \in \mathbb{R}_{\geq 0}, \forall z \in Z, f_n(z,X) \leq r_n$.*

Proposition 5 will demonstrate that the class of s-finite kernels (Definition 23) includes all pushforwards of σ-finite kernels, and therefore the pushforwards of all measure kernels admitting densities.

Proposition 5 (s-finite kernels are pushforwards of σ-finite kernels [34, 32])**.** *A measure kernel $f : Z \times \Sigma_X \to [0,\infty]$ is s-finite if and only if it is a pushforward $f = \mathbf{copy}_Z \mathbin{⨟} (p \otimes id_Z) \mathbin{⨟} k$ of a σ-finite measure kernel p through a deterministic k.*

The above proposition includes trivial pushforwards, so every σ-finite (Definition 20) measure kernel is s-finite (Definition 23) but not the other way around.

B Parametric and coparametric categories

This section will review the definitions of parametric and coparametric (bi)categories, first given in the categorical cybernetics literature [5]. For the sake of rigor, the reader can also see a recent review on actegories [6]. As a starting point, Definition 24 will describe how a symmetric monoidal category (SMC) can "act upon" another category functorially.

Definition 24 ($\mathscr{M}$-actegory)**.** *Consider a symmetric monoidal category $(\mathscr{M}, J, \odot)$ and a category $\mathscr{C}$. An $\mathscr{M}$-actegory is a pair of the two with a functor $\bullet : \mathscr{M} \times \mathscr{C} \to \mathscr{C}$ from the product category and natural transformations $\varepsilon : J \bullet X \simeq X$ and $\delta : (M \bullet N) \bullet X = M \bullet (N \bullet X)$.*

Definition 25 will then apply the actegory concept to define a bicategory whose morphisms accumulate parameters in the course of composition.

Definition 25 (Parametric categories [5])**.** *Given an $\mathscr{M}$-actegory $\mathscr{C}$, the* parametric (bi)category *$\mathbf{Para}_\bullet(\mathscr{C})$ has as objects those of $\mathscr{C}$ and as morphisms the pairs $\mathbf{Para}_\bullet(\mathscr{C})(A,B) = \{(M,k) \in Ob(\mathscr{M}) \times \mathscr{C}(M \bullet A, B)\}$. Composition for morphisms $(M,k) : \mathbf{Para}_\bullet(\mathscr{C})(A,B)$ and $(M',k') : \mathbf{Para}_\bullet(\mathscr{C})(B,C)$ consists of $(M' \odot M, k' \circ (id_{M'} \bullet k)))$ while identities on objects A consist of (I, id_A).*

Parametric (bi)categories of course have a dual, definable as $\mathbf{Para}_\bullet(\mathscr{C}^{op})^{op}$. Definition 26 will describe this category, whose morphisms admit "coparameters" accumulate extra elements of the codomain.

Definition 26 (Coparametric categories [5]). *Given an $\mathscr{M}$-actegory $\mathscr{C}$, the* coparametric category

$$\mathbf{CoPara}_\bullet(\mathscr{C}) \in Ob(\mathbf{Cat})$$

has as objects those of $\mathscr{C}$ and as morphisms $\mathbf{CoPara}_\bullet(\mathscr{C})(A,B)$ the pairs $(M,f) \in Ob(\mathscr{M}) \times \mathscr{C}(A, M \bullet B)$. Composition for $(M,f) : \mathbf{CoPara}_\bullet(\mathscr{C})(A,B)$ and $(M',g) : \mathbf{CoPara}_\bullet(\mathscr{C})(B,C)$ consists of $(M \odot M', (id_M \bullet g) \circ f))$ while identities on objects A consist of (I, id_A).

The coparametric category construction generalizes the idea of a writer monad to more than one object, and represents morphisms that "log" or "leave behind" a cumulative effect. Definition 27 will describe symmetric monoidality for the $\mathscr{M}$-actegory on $\mathscr{C}$ when $(\mathscr{C}, \otimes I)$ is symmetric monoidal.

Definition 27 (Symmetric monoidal $\mathscr{M}$-actegory). *A* symmetric monoidal $\mathscr{M}$-actegory *is an $\mathscr{M}$-actegory $\mathscr{C}$ equipped with a symmetric monoidal structure and a natural isomorphism $\kappa_{M,X,Y} : M \bullet (X \otimes Y) \simeq X \otimes (M \bullet Y)$, satisfying coherence laws similar to those of a costrong comonad.*

Finally, Proposition 6 will demonstrate that given a symmetric monoidal actegory as in Definition 27, the constructions above admit symmetric monoidal structure themselves.

Proposition 6 (Parametric and coparametric categories admit monoidal structure [6][11]). *Given a symmetric monoidal $\mathscr{M}$-actegory $(\mathscr{C}, \otimes, I)$, the parametric bicategory $\mathbf{Para}_\bullet(\mathscr{C})$ and coparametric bicategory $\mathbf{CoPara}_\bullet(\mathscr{C})$ form symmetric monoidal bicategories $(\mathbf{Para}_\bullet(\mathscr{C}), \otimes, I)$ and $(\mathbf{CoPara}_\bullet(\mathscr{C}), \otimes, I)$.*

C Free copy/delete and Markov categories

Generalized causal models [14] employ hypergraphs, which "flip" the status of nodes and edges relative to ordinary graphs: "hypernodes" are drawn as wires and "hyperedges" connecting them as boxes. These hypergraphs represent string diagrams combinatorially; restricting hypergraphs to conditions matching certain kinds of categories defines "free" categories of those kinds. This subsection will build up free copy/delete and Markov categories with generalized causal models as morphisms.

Definition 28 defines hypergraphs via sets [16]; Bonchi et al [4] provides categorical intuition.

Definition 28 (Hypergraph). *A* hypergraph *is a 4-tuple $(W, B, \mathrm{dom}, \mathrm{cod})$ consisting of a set of vertices, nodes, or "wires" W; a set of hyperedges or "boxes" B; a function* $\mathrm{dom} : B \to W^*$ *assigning a domain to each box; and a function* $\mathrm{cod} : B \to W^*$ *assigning a codomain to each box.*

We abuse notation and write individual boxes $b \in B : \mathrm{dom}(b) \to \mathrm{cod}(b)$.

Definition 29 specifies relabelings of one hypergraph's wires and boxes with those of another.

Definition 29 (Hypergraph morphism). *Given hypergraphs G, H, a* hypergraph morphism $\alpha : G \to H$ *is a pair of functions assigning wires to wires and boxes to boxes, the latter respecting the former*

$$\mathbf{Hyp}(G,H) := \left\{ (\alpha_W, \alpha_B) \in W(H)^{W(G)} \times B(H)^{B(G)} \mid \forall b \in B(G), \alpha_B(b) : \alpha_W(\mathrm{dom}(b)) \to \alpha_W(\mathrm{cod}(b)) \right\}.$$

As implied by the hom-set notation, hypergraphs and their morphisms form a category $\mathbf{Hyp}$ [4], and our application will employ the full subcategory $\mathbf{FinHyp}$ in which W and B both have finite cardinality. Finally, a hypergraph H is *discrete* when $B(H) = \emptyset$; $\underline{n}$ denotes a discrete hypergraph with $n \in \mathbb{N}$ wires.

[11] Example 5.1.8

Any monoidal category has a (potentially infinite) underlying hypergraph, which we denote $\mathrm{hyp}(\cdot) : \mathbf{MonCat} \to \mathbf{Hyp}$ following Fritz and Liang [15].

Often a finite hypergraph $\Sigma \in \mathbf{FinHyp}$ denotes the generating objects and morphisms of a free monoidal category, or the primitive types and functions of a domain-specific programming language. We call such a finite hypergraph a *monoidal signature*. Definition 30 formally defines the copy/delete category freely generated by a signature Σ, which Definition 31 will restrict to free Markov categories.

Definition 30 (Free copy/delete category for the signature Σ [15])**.** *The* free CD category $\mathbf{FreeCD}_\Sigma$ *for* $\Sigma \in \mathbf{FinHyp}$ *is a subcategory* $\mathbf{FreeCD}_\Sigma \subseteq \mathbf{cospan}(\mathbf{FinHyp}/\Sigma)$ *where*

- *Objects are the pairs* $(n, \sigma) \in \mathbb{N} \times \underline{n} \to \Sigma$ *assigning outer wires of a string diagram to wires in* Σ*;*
- *Morphisms are isomorphism classes of cospans, given combinatorially*

$$\mathbf{FreeCD}_\Sigma((n,\sigma_n),(m,\sigma_m)) = \{p \to \mathrm{dom}(\tau) \leftarrow q \in \mathbf{FinHyp}(\underline{n}, \mathrm{dom}(\tau)) \times Ob(\mathbf{FinHyp}/\Sigma) \times \mathbf{FinHyp}(\underline{m}, \mathrm{dom}(\tau))\},$$

such that $\tau : G \to \Sigma \in Ob(\mathbf{FinHyp}/\Sigma)$ *is a hypergraph morphism from an acyclic G and every wire* $w \in W(G)$ *has at most one "starting place" as the diagram's input or a box's output*

$$|p^{-1}(w)| + \sum_{b \in B(G)} \sum_{w' \in \mathrm{cod}(b)} \mathbb{I}[w' = w] \leq 1.$$

Intuitively, a morphism in $\mathbf{FreeCD}_\Sigma$ is syntax specifying a string diagram with no looping or merging wires, whose boxes and wires are labeled by Σ. Definition 31 passes to the free Markov category $\mathbf{FreeMarkov}_\Sigma$ just by syntactically enforcing the naturality of $\mathbf{del}_Z$.

Definition 31 (Free Markov category for the signature Σ)**.** *The* free Markov category $\mathbf{FreeMarkov}_\Sigma$ *for* $\Sigma \in \mathbf{FinHyp}$ *is the wide subcategory of* $\mathbf{FreeCD}_\Sigma$ *restricted to morphisms in which every output from every box connects to somewhere else*

$$\mathrm{connects}(w, G, q) := \mathbb{I}[\exists b \in B(G) : w \in \mathrm{cod}(b) \implies q^{-1}(w) \neq \emptyset \vee \exists b' \in B(G) : w \in \mathrm{dom}(b')]$$

$$\mathbf{FreeMarkov}_\Sigma(n,m) := \{p \to \mathrm{dom}(\tau) \leftarrow q \in \mathbf{FreeCD}_\Sigma(n,m) \mid \forall w \in W(\mathrm{dom}(\tau)), \mathrm{connects}(w, \mathrm{dom}(\tau), q)\},$$

and with composition redefined to syntactically enforce this by iterating the deletion of discarded boxes to a fixed-point after composition in $\mathbf{FreeCD}_\Sigma$*.*

Approximate Inference via Fibrations of Statistical Games

Toby St Clere Smithe
VERSES Research
Topos Institute
q@tsmithe.net

We characterize a number of well known systems of approximate inference as *loss models*: lax sections of 2-fibrations of statistical games, constructed by attaching internally-defined loss functions to Bayesian lenses. Our examples include the relative entropy, which constitutes a *strict* section, and whose chain rule is formalized by the horizontal composition of the 2-fibration. In order to capture this compositional structure, we first introduce the notion of 'copy-composition', alongside corresponding bicategories through which the composition of copy-discard categories factorizes. These bicategories are a variant of the **Copara** construction, and so we additionally introduce coparameterized Bayesian lenses, proving that coparameterized Bayesian updates compose optically, as in the non-coparameterized case.

1 Introduction

In previous work [1], we introduced *Bayesian lenses*, observing that the Bayesian inversion of a composite stochastic channel is (almost surely) equal to the 'lens composite' of the inversions of the factors; that is, *Bayesian updates compose optically* ('BUCO') [2]. Formalizing this statement for a given category $\mathscr{C}$ all of whose morphisms ('channels') admit Bayesian inversion, we can observe that there is (almost surely) a functor $(-)^\dagger : \mathscr{C} \to \mathbf{BayesLens}(\mathscr{C})$ from $\mathscr{C}$ to the category $\mathbf{BayesLens}(\mathscr{C})$ whose morphisms $(X,A) \nrightarrow (Y,B)$ are Bayesian lenses: pairs (c,c') of a channel $X \twoheadrightarrow Y$ with a 'state-dependent' inverse $c' : \mathscr{C}(I,X) \to \mathscr{C}(B,A)$. Bayesian lenses constitute the morphisms of a fibration $\pi_{\mathsf{Lens}} : \mathbf{BayesLens}(\mathscr{C}) \to \mathscr{C}$, since $\mathbf{BayesLens}(\mathscr{C})$ is obtained as the Grothendieck construction of (the pointwise opposite of) an indexed category $\mathsf{Stat} : \mathscr{C}^{\,\mathrm{op}} \to \mathbf{Cat}$ of 'state-dependent channels' (recalled in Appendix A), and the functor $(-)^\dagger$ is in fact a section of π_{Lens}, taking $c : X \twoheadrightarrow Y$ to the lens $(c,c^\dagger) : (X,X) \nrightarrow (Y,Y)$, where $c^\dagger$ is the almost-surely unique Bayesian inversion of c (so that the projection π_{Lens} can simply forget the inversion, returning again the channel c).

The functor $(-)^\dagger$ picks out a special class of Bayesian lenses, which we may call *exact* (as they compute 'exact' inversions), but although the category $\mathbf{BayesLens}(\mathscr{C})$ has many other morphisms, the construction is not extravagant: by comparison, we can think of the non-exact lenses as representing *approximate* inference systems. This is particularly necessary in computational applications, because computing exact inversions is usually intractable, but this creates a new problem: choosing an approximation, and measuring its performance. In this paper, we formalize this process, by attaching *loss functions* to Bayesian lenses, thus creating another fibration, of *statistical games*. Sections of this latter fibration encode compositionally well-behaved systems of approximation that we call *loss models*.

A classic example of a loss model will be supplied by the relative entropy, which in some sense measures the 'divergence' between distributions: the game here is then to minimize the divergence between the approximate and exact inversions. If π and π' are two distributions on a space X, with corresponding density functions p_π and $p_{\pi'}$ (both with respect to a common measure), then their relative

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 279–298, doi:10.4204/EPTCS.397.17

entropy $D(\pi,\pi')$ is the real number given by $\mathbb{E}_{x\sim\pi}[\log p_\pi(x) - \log p_{\pi'}(x)]$[1]. Given a pair of channels $\alpha,\alpha' : A \twoheadrightarrow B$ (again commensurately associated with densities), we can extend D to a map $D_{\alpha,\alpha'} : A \to \mathbb{R}_+$ in the natural way, writing $a \mapsto D\big(\alpha(a),\alpha'(a)\big)$. We can assign such a map $D_{\alpha,\alpha'}$ to any such parallel pair of channels, and so, following the logic of composition in $\mathscr{C}$, we might hope for the following equation to hold for all $a : A$ and composable parallel pairs $\alpha,\alpha' : A \twoheadrightarrow B$ and $\beta,\beta' : B \twoheadrightarrow C$,:

$$D_{\beta\bullet\alpha,\beta'\bullet\alpha'}(a) = \mathop{\mathbb{E}}_{b\sim\alpha(a)}\big[D_{\beta,\beta'}(b)\big] + D_{\alpha,\alpha'}(a)$$

The right-hand side is known as the *chain rule* for relative entropy, but, unfortunately, the equation does *not* hold in general, because the composites $\beta\bullet\alpha$ and $\beta'\bullet\alpha'$ involve an extra expectation (by the 'Chapman-Kolmogorov' rule for channel composition). However, we *can* satisfy an equation of this form by using 'copy-composition': if we write $\curlyvee_B$ to denote the canonical 'copying' comultiplication on B, and define $\beta\bullet^2\alpha := (\mathsf{id}_B\otimes\beta)\bullet\curlyvee_B\bullet\alpha$, then $D_{\beta\bullet^2\alpha,\beta'\bullet^2\alpha'}(a)$ *does* equal the chain-rule form on the right-hand side. This result exhibits a general pattern about "copy-discard categories" [3] such as $\mathscr{C}$: composition $\bullet$ can be decomposed into first copying $\curlyvee$, and then discarding $\top$. If we don't discard, then we retain the 'intermediate' variables, and this results in a functorial assignment of relative entropies to channels.

The rest of this paper is dedicated to making use of this observation to construct loss models, including (but not restricted to) the relative entropy. The first complication that we encounter is that copy-composition is not strictly unital, because composing with an identity retains an extra variable. To deal with this, in §2, we construct a *bicategory* of copy-composite channels, extending the **Copara** construction [4, §2], and build coparameterized (copy-composite) Bayesian lenses accordingly; we also prove a corresponding BUCO result. In §3, we then construct 2-fibrations of statistical games, defining loss functions internally to any copy-discard category $\mathscr{C}$ that admits "bilinear effects". Because we are dealing with approximate systems, the 2-dimensional structure of the construction is useful: loss models are allowed to be *lax* sections. We then characterize the relative entropy, maximum likelihood estimation, the free energy, and the 'Laplacian' free energy as such loss models.

Assuming $\mathscr{C}$ is symmetric monoidal, the constructions here result in monoidal (2-)fibrations, but due to space constraints we defer the presentation of this structure (and its exemplification by the foregoing loss models) to Appendix B.

Remark 1.1. Much of this work is situated amongst monoidal fibrations of bicategories, the full theory of which is not known to the present author. Fortunately, enough structure is known for the present work to have been possible, and where things become murkier—such as in the context of monoidal indexed bicategories and their lax homomorphisms—the way largely seems clear. For this, we are grateful to Baković [5], Johnson and Yau [6], and Moeller and Vasilakopoulou [7] in particular for lighting the way; and we enthusiastically encourage the further elucidation of these structures by category theorists.

Remark 1.2. For reasons of space, detailed proofs are not included in the proceedings version of this paper; however, they are included in an appendix to the conference submission, which is available on the arXiv repository with the paper ID 2306.17009v1.

2 'Copy-composite' Bayesian lenses

2.1 Copy-composition by coparameterization

In a locally small copy-discard category $\mathscr{C}$, every object A is equipped with a canonical comonoid structure $(\curlyvee_A, \top_A)$, and so, by the comonoid laws, it is almost a triviality that the composition function

[1] For details about this 'expectation' notation $\mathbb{E}$, see 3.11.

$\bullet : \mathscr{C}(B,C) \times \mathscr{C}(A,B) \to \mathscr{C}(A,C)$ factorizes as

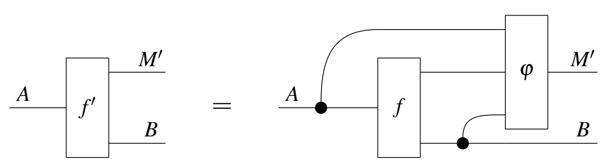
$$\mathscr{C}(B,C) \times \mathscr{C}(A,B) \xrightarrow{(\mathsf{id}_B \otimes -) \times \mathscr{C}\left(\mathsf{id}_A, \curlyvee_B\right)} \mathscr{C}(B \otimes B, B \otimes C) \times \mathscr{C}(A, B \otimes B) \cdots$$
$$\cdots \xrightarrow{\bullet} \mathscr{C}(A, B \otimes C) \xrightarrow{\mathscr{C}(\mathsf{id}_A, \mathsf{proj}_C)} \mathscr{C}(A,C)$$

where the first factor copies the B output of the first morphism and tensors the second morphism with the identity on B, the second factor composes the latter tensor with the copies, and the third discards the extra copy of B^2. This is, however, only *almost* trivial, since it witnesses the structure of 'Chapman-Kolmogorov' style composition in categories of stochastic channels such as $\mathscr{K}\ell(\mathscr{D})$, the Kleisli category of the (finitary) distributions monad $\mathscr{D} : \mathbf{Set} \to \mathbf{Set}$. There, given channels $c : A \twoheadrightarrow B$ and $d : B \twoheadrightarrow C$, the composite $d \bullet c$ is formed first by constructing the 'joint' channel $d \bullet^2 c$ defined by $(d \bullet^2 c)(b,c|a) := d(c|b)c(b|a)$, and then discarding (marginalizing over) $b : B$, giving

$$(d \bullet c)(c|a) = \sum_{b:B} (d \bullet^2 c)(b,c|a) = \sum_{b:B} d(c|b)c(b|a)\,.$$

Of course, the channel $d \bullet^2 c$ is not a morphism $A \twoheadrightarrow C$, but rather $A \twoheadrightarrow B \otimes C$; that is, it is *coparameterized* by B. Moreover, as noted above, $\bullet^2$ is not strictly unital: we need a 2-cell that discards the coparameter, and hence a bicategory, in order to recover (weak) unitality. We therefore construct a bicategory $\mathbf{Copara}_2(\mathscr{C})$ as a variant of the **Copara** construction [4, §2], in which a 1-cell $A \to B$ may be any morphism $A \twoheadrightarrow M \otimes B$ in $\mathscr{C}$, and where horizontal composition is precisely copy-composition.

Theorem 2.1. Let $(\mathscr{C}, \otimes, I)$ be a copy-discard category. Then there is a bicategory $\mathbf{Copara}_2(\mathscr{C})$ as follows. Its 0-cells are the objects of $\mathscr{C}$. A 1-cell $f : A \xrightarrow[M]{} B$ is a morphism $f : A \to M \otimes B$ in $\mathscr{C}$. A 2-cell $\varphi : f \Rightarrow f'$, with $f : A \xrightarrow[M]{} B$ and $f' : A \xrightarrow[M']{} B$, is a morphism $\varphi : A \otimes M \otimes B \to M'$ of $\mathscr{C}$, satisfying the *change of coparameter* axiom:

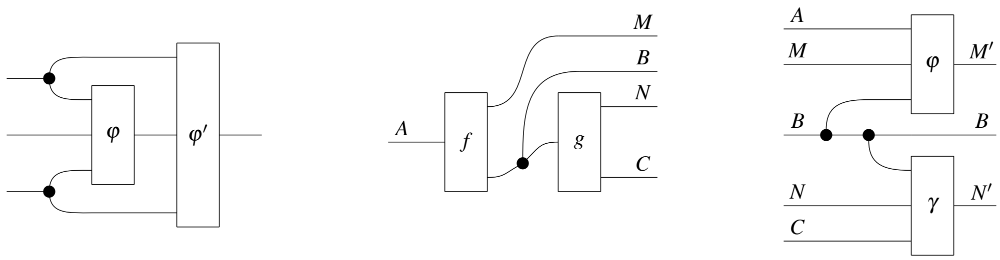


The identity 2-cell $\mathsf{id}_f : f \Rightarrow f$ on $f : A \xrightarrow[M]{} B$ is given by the projection morphism $\mathsf{proj}_M : A \otimes M \otimes B \to M$ obtained by discarding A and B, as in footnote 2. The identity 1-cell id_A on A is given by the inverse of the left unitor of the monoidal structure on $\mathscr{C}$, *i.e.* $\mathsf{id}_A := \lambda_A^{-1} : A \xrightarrow[I]{} A$, with coparameter thus given by the unit object I.

Given 2-cells $\varphi : f \Rightarrow f'$ and $\varphi' : f' \Rightarrow f''$, their vertical composite $\varphi' \odot \varphi : f \Rightarrow f''$ is given by the string diagram on the left below. Given 1-cells $f : A \xrightarrow[M]{} B$ then $g : B \xrightarrow[N]{} C$, the horizontal composite $g \circ f : A \xrightarrow[(M \otimes B) \otimes N]{} C$ is given by the middle string diagram below. Given 2-cells $\varphi : f \Rightarrow f'$ and $\gamma : g \Rightarrow g'$ between 1-cells $f, f' : A \xrightarrow[M]{} B$ and $g, g' : B \xrightarrow[N]{} C$, their horizontal composite $\gamma \circ \varphi : (g \circ f) \Rightarrow (g' \circ f')$ is

[2] We define $\mathsf{proj}_C := B \otimes C \xrightarrow{\top_B \otimes \mathsf{id}_C} I \otimes C \xrightarrow{\lambda_C} C$, using the comonoid counit and the left unitor of $\mathscr{C}$'s monoidal structure.

defined by the string diagram on the right below.



Remark 2.2. When $\mathscr{C}$ is symmetric monoidal, $\mathbf{Copara}_2(\mathscr{C})$ inherits a monoidal structure, elaborated in Proposition B.1.

Remark 2.3. In order to capture the bidirectionality of Bayesian inversion we will need to conside left- and right-handed versions of the $\mathbf{Copara}_2$ construction. These are formally dual, and when $\mathscr{C}$ is symmetric monoidal (as in most examples) they are isomorphic. Nonetheless, it makes formalization easier if we explicitly distinguish $\mathbf{Copara}_2^l(\mathscr{C})$, in which the coparameter is placed on the left of the codomain (as above), from $\mathbf{Copara}_2^r(\mathscr{C})$, in which it is placed on the right. Aside from the swapping of this handedness, the rest of the construction is the same.

We end this section with three easy (and ambidextrous) propositions relating $\mathscr{C}$ and $\mathbf{Copara}_2(\mathscr{C})$.

Proposition 2.4. There is an identity-on-objects lax embedding $\iota : \mathscr{C} \hookrightarrow \mathbf{Copara}_2(\mathscr{C})$, mapping $f : X \to Y$ to $f : X \underset{I}{\to} Y$ (using the unitor of the monoidal structure on $\mathscr{C}$). The laxator $\iota(g) \circ \iota(f) \to \iota(g \circ f)$ discards the coparameter obtained from copy-composition.

Proposition 2.5. There is a 'discarding' functor $(-)^{\dagger} : \mathbf{Copara}_2(\mathscr{C}) \to \mathscr{C}$, which takes any coparameterized morphism and discards the coparameter.

Proposition 2.6. ι is a section of $(-)^{\dagger}$. That is, $\mathsf{id}_{\mathscr{C}} = \mathscr{C} \xhookrightarrow{\iota} \mathbf{Copara}_2(\mathscr{C}) \xrightarrow{(-)^{\dagger}} \mathscr{C}$.

2.2 Coparameterized Bayesian lenses

In order to define (bi)categories of statistical games, coherently with loss functions like the relative entropy that compose by copy-composition, we first need to define coparameterized (copy-composite) Bayesian lenses. Analogously to non-coparameterized Bayesian lenses, these will be obtained by applying a Grothendieck construction to an indexed bicategory [5, Def. 3.5] of state-dependent channels.

Definition 2.7. We define the indexed bicategory $\mathsf{Stat}_2 : \mathbf{Copara}_2^l(\mathscr{C})^{\mathrm{co\,op}} \to \mathbf{Bicat}$ fibrewise as follows.

(i) The 0-cells $\mathsf{Stat}_2(X)_0$ of each fibre $\mathsf{Stat}_2(X)$ are the objects $\mathscr{C}_0$ of $\mathscr{C}$.

(ii) For each pair of 0-cells A, B, the hom-category $\mathsf{Stat}_2(X)(A,B)$ is defined to be the functor category $\mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X), \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big)$, where disc denotes the functor taking a set to the associated discrete category.

(iii) For each 0-cell A, the identity functor $\mathsf{id}_A : \mathbf{1} \to \mathsf{Stat}_2(X)(A,A)$ is the constant functor on the identity on A in $\mathbf{Copara}_2^r(\mathscr{C})$; *i.e.* $\mathsf{disc}\,\mathscr{C}(I,X) \xrightarrow{!} 1 \xrightarrow{\mathsf{id}_A} \mathbf{Copara}_2^r(\mathscr{C})(A,A)$.

(iv) For each triple A,B,C of 0-cells, the horizontal composition functor $\circ_{A,B,C}$ is defined by

$$\begin{aligned}\circ_{A,B,C} : \mathsf{Stat}_2(X)(B,C) \times \mathsf{Stat}_2(X)(A,B) \cdots \\ \cdots \xrightarrow{=} \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X), \mathbf{Copara}_2^r(\mathscr{C})(B,C)\big) \times \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X), \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big) \cdots \\ \cdots \xrightarrow{\times} \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X)^2, \mathbf{Copara}_2^r(\mathscr{C})(B,C) \times \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big) \cdots \\ \cdots \xrightarrow{\mathbf{Cat}(\curlyvee,\circ)} \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X), \mathbf{Copara}_2^r(\mathscr{C})(A,C)\big) \cdots \\ \cdots \xrightarrow{=} \mathsf{Stat}_2(X)(A,C)\end{aligned}$$

where $\mathbf{Cat}(\curlyvee,\circ)$ indicates pre-composition with the universal (Cartesian) copying functor in $(\mathbf{Cat},\times,\mathbf{1})$ and post-composition with the horizontal composition functor of $\mathbf{Copara}_2^r(\mathscr{C})$.

For each pair of 0-cells X,Y in $\mathbf{Copara}^l(\mathscr{C})$, we define the reindexing pseudofunctor $\mathsf{Stat}_{2,X,Y}$: $\mathbf{Copara}^l(\mathscr{C})(X,Y)^{\mathrm{op}} \to \mathbf{Bicat}\big(\mathsf{Stat}_2(Y), \mathsf{Stat}_2(X)\big)$ as follows.

(a) For each 1-cell f in $\mathbf{Copara}^l(\mathscr{C})(X,Y)$, we obtain a pseudofunctor $\mathsf{Stat}_2(f) : \mathsf{Stat}_2(Y) \to \mathsf{Stat}_2(X)$ which acts as the identity on 0-cells.

(b) For each pair of 0-cells A,B in $\mathsf{Stat}_2(Y)$, the functor $\mathsf{Stat}_2(f)_{A,B}$ is defined as the precomposition functor $\mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,f^{\dagger}), \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big)$, where $(-)^{\dagger}$ is the discarding functor $\mathbf{Copara}_2^l(\mathscr{C}) \to \mathscr{C}$ of Proposition 2.5.

(c) For each 2-cell $\varphi : f \Rightarrow f'$ in $\mathbf{Copara}_2^l(\mathscr{C})(X,Y)$, the pseudonatural transformation $\mathsf{Stat}_2(\varphi)$: $\mathsf{Stat}_2(f') \Rightarrow \mathsf{Stat}_2(f)$ is defined on 0-cells $A : \mathsf{Stat}_2(Y)$ by the discrete natural transformation with components $\mathsf{Stat}_2(\varphi)_A := \mathsf{id}_A$, and on 1-cells $c : \mathsf{Stat}_2(Y)(A,B)$ by the substitution natural transformation with constitutent 2-cells $\mathsf{Stat}_2(\varphi)_c : \mathsf{Stat}_2(f)(c) \Rightarrow \mathsf{Stat}_2(f')(c)$ in $\mathsf{Stat}_2(X)$ which acts by replacing $\mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,f^{\dagger}), \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big)$ by $\mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,f'^{\dagger}), \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big)$; and which we might alternatively denote by $\mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,\varphi^{\dagger}), \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big)$.

Notation 2.8. We will write $f : A \overset{X}{\underset{M}{\rightsquigarrow}} B$ to denote a state-dependent coparameterized channel f with coparameter M and state-dependence on X.

In 1-category theory, lenses are morphisms in the fibrewise opposite of a fibration [8]. Analogously, our bicategorical Bayesian lenses are obtained as 1-cells in the bicategorical Grothendieck construction [5, §6] of (the pointwise opposite of) the indexed bicategory Stat_2.

Definition 2.9. Fix a copy-discard category $(\mathscr{C},\otimes,I)$. We define the bicategory of coparameterized Bayesian lenses in $\mathscr{C}$, denoted $\mathbf{BayesLens}_2(\mathscr{C})$ or simply $\mathbf{BayesLens}_2$, to be the bicategorical Grothendieck construction of the pointwise opposite of the corresponding indexed bicategory Stat_2, with the following data:

(i) A 0-cell in $\mathbf{BayesLens}_2$ is a pair (X,A) of an object X in $\mathbf{Copara}_2^l(\mathscr{C})$ and an object A in $\mathsf{Stat}_2(X)$; equivalently, a 0-cell in $\mathbf{BayesLens}_2$ is a pair of objects in $\mathscr{C}$.

(ii) The hom-category $\mathbf{BayesLens}_2\big((X,A),(Y,B)\big)$ is the product category $\mathbf{Copara}_2^l(\mathscr{C})(X,Y) \times \mathsf{Stat}_2(X)(B,A)$.

(iii) The identity on (X,A) is given by the pair $(\mathsf{id}_X, \mathsf{id}_A)$.

(iv) For each triple of 0-cells $(X,A),(Y,B),(Z,C)$, the horizontal composition functor is given by

$$
\begin{aligned}
& \mathbf{BayesLens}_2\big((Y,B),(Z,C)\big) \times \mathbf{BayesLens}_2\big((X,A),(Y,B)\big) \\
= \quad & \mathbf{Copara}_2^l(\mathscr{C})(Y,Z) \times \mathsf{Stat}_2(Y)(C,B) \times \mathbf{Copara}_2^l(\mathscr{C})(X,Y) \times \mathsf{Stat}_2(X)(B,A) \\
\xrightarrow{\sim} \quad & \sum_{g:\mathbf{Copara}_2^l(\mathscr{C})(Y,Z)} \; \sum_{f:\mathbf{Copara}_2^l(\mathscr{C})(X,Y)} \mathsf{Stat}_2(Y)(C,B) \times \mathsf{Stat}_2(X)(B,A) \\
\xrightarrow{\sum_g \sum_f \mathsf{Stat}_2(f)_{C,B} \times \mathsf{id}} \quad & \sum_{g:\mathbf{Copara}_2^l(\mathscr{C})(Y,Z)} \; \sum_{f:\mathbf{Copara}_2^l(\mathscr{C})(X,Y)} \mathsf{Stat}_2(X)(C,B) \times \mathsf{Stat}_2(X)(B,A) \\
\xrightarrow{\sum_{\circ_{\mathbf{Copara}_2^l(\mathscr{C})}} \circ^{\mathsf{Stat}_2(X)}} \quad & \sum_{g\circ f:\mathbf{Copara}_2^l(\mathscr{C})(X,Z)} \mathsf{Stat}_2(X)(C,A) \\
\xrightarrow{\sim} \quad & \mathbf{BayesLens}_2\big((X,A),(Z,C)\big)
\end{aligned}
$$

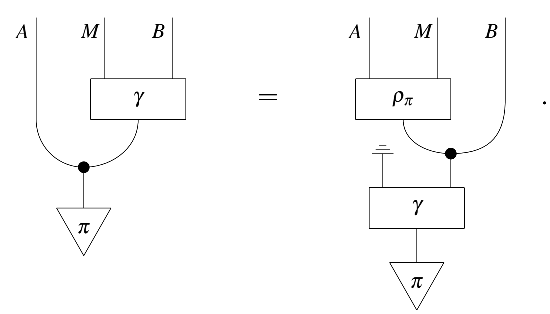
where the functor in the penultimate line amounts to the product of the horizontal composition functors on $\mathbf{Copara}_2^l(\mathscr{C})$ and $\mathsf{Stat}_2(X)$.

Remark 2.10. When $\mathscr{C}$ is symmetric monoidal, Stat_2 acquires the structure of a monoidal indexed bicategory (Definition B.2 and Theorem B.3), and hence $\mathbf{BayesLens}_2$ becomes a monoidal bicategory (Corollary B.4).

2.3 Coparameterized Bayesian updates compose optically

So that our generalized Bayesian lenses are worthy of the name, we should also confirm that Bayesian inversions compose according to the lens pattern ('optically') also in the coparameterized setting. Such confirmation is the subject of the present section, and so we first introduce a new "coparameterized Bayes' rule".

Definition 2.11. We say that a coparameterized channel $\gamma : A \to\!\!\!\!\!\bullet\, M \otimes B$ *admits Bayesian inversion* if there exists a dually coparameterized channel $\rho_\pi : B \to\!\!\!\!\!\bullet\, A \otimes M$ satisfying the graphical equation (with string diagrams read from bottom to top)



In this case, we say that ρ_π is the *Bayesian inversion of* γ *with respect to* π.

With this definition, we can supply the desired result that "coparameterized Bayesian updates compose optically".

Theorem 2.12. Suppose $(\gamma,\gamma^\dagger) : (A,A) \underset{M}{\rightarrow\!\!\!\!\!\!\!\mapsto} (B,B)$ and $(\delta,\delta^\dagger) : (B,B) \underset{N}{\rightarrow\!\!\!\!\!\!\!\mapsto} (C,C)$ are coparameterized Bayesian lenses in $\mathbf{BayesLens}_2$. Suppose also that $\pi : I \to\!\!\!\!\!\bullet\, A$ is a state on A in the underlying category of channels $\mathscr{C}$, such that $\gamma_\pi^\dagger$ is a Bayesian inversion of γ with respect to π, and such that $\delta_{\gamma\pi}^\dagger$ is a Bayesian inversion of δ with respect to $(\gamma\pi)^{\dagger}$; where the notation $(-)^{\dagger}$ represents discarding coparameters. Then $\gamma_\pi^\dagger \bullet \delta_{\gamma\pi}^\dagger$ is a Bayesian inversion of $\delta \bullet \gamma$ with respect to π. (Here $\bullet$ denotes copy-composition.) Moreover, if $(\delta \bullet \gamma)_\pi^\dagger$ is any Bayesian inversion of $\delta \bullet \gamma$ with respect to π, then $\gamma_\pi^\dagger \bullet \delta_{\gamma\pi}^\dagger$ is $(\delta\gamma\pi)^{\dagger}$-almost-surely equal to $(\delta \bullet \gamma)_\pi^\dagger$: that is, $(\delta \bullet \gamma)_\pi^\dagger \overset{(\delta\gamma\pi)^{\dagger}}{\sim} \gamma_\pi^\dagger \bullet \delta_{\gamma\pi}^\dagger$.

In order to satisfy this coparameterized Bayes' rule, a Bayesian lens must be of 'simple' type.

Definition 2.13. We say that a coparameterized Bayesian lens (c, c') is *simple* if its domain and codomain are 'diagonal' (duplicate pairs of objects) and if the coparameter of c is equal to the coparameter of c'. In this case, we can write the type of (c, c') as $(X, X) \underset{M}{\nrightarrow} (Y, Y)$ or simply $X \underset{M}{\nrightarrow} Y$.

3 Statistical games for local approximate inference

3.1 Losses for lenses

Statistical games are obtained by attaching to Bayesian lenses *loss functions*, representing 'local' quantifications of the performance of approximate inference systems. Because this performance depends on the system's context (*i.e.*, the prior $\pi : I \twoheadrightarrow X$ and the observed data $b : B$), a loss function at its most concrete will be a function $\mathscr{C}(I, X) \times B \to \mathbb{R}_+$. To internalize this type in $\mathscr{C}$, we may recall that, when $\mathscr{C}$ is the category **sfKrn** of s-finite kernels or the Kleisli category $\mathscr{K}\ell(\mathscr{D}_{\leqslant 1})$ of the subdistribution monad, a density function $p_c : X \times Y \to [0, 1]$ for a channel $c : X \twoheadrightarrow Y$ corresponds to an *effect* (or *costate*) $X \otimes Y \twoheadrightarrow I$. In this way, we can see a loss function as a kind of *state-dependent effect* $B \overset{X}{\twoheadrightarrow} I$.

Loss functions will compose by sum, and so we need to ask for the effects in $\mathscr{C}$ to form a monoid. Moreover, we need this monoid to be 'bilinear' with respect to channels, so that Stat-reindexing (*cf.* Definition A.1) preserves sums. These conditions are formalized in the following definition.

Definition 3.1. Suppose $(\mathscr{C}, \otimes, I)$ is a copy-discard category. We say that $\mathscr{C}$ *has bilinear effects* if the following conditions are satisfied:

(i) *effect monoid*: there is a natural transformation $+ : \mathscr{C}(-, I) \times \mathscr{C}(=, I) \Rightarrow \mathscr{C}(- \otimes =, I)$ making $\sum_{A:\mathscr{C}} \mathscr{C}(A, I)$ into a commutative monoid with unit $0 : I \twoheadrightarrow I$;

(ii) *bilinearity*: $(g + g') \bullet \curlyvee \bullet f = g \bullet f + g' \bullet f$ for all effects g, g' and morphisms f such that $(g + g') \bullet \curlyvee \bullet f$ exists.

A trivial example of a category with bilinear effects is supplied by any Cartesian category, such as **Set**. If M is any monoid in **Set**, then a less trivial example is supplied by the Kleisli category of the corresponding free module monad; bilinearity follows from the module structure. A related non-example is $\mathscr{K}\ell(\mathscr{D}_{\leqslant 1})$: the failure here is that the effects only form a *partial* monoid[3]. More generally, the category **sfKrn** of s-finite kernels [10] has bilinear effects (owing to the linearity of integration), and we will assume this as our ambient $\mathscr{C}$ for the examples below.

Given such a category $\mathscr{C}$ with bilinear effects, we can lift the natural transformation $+$, and hence the

[3]Indeed, an *effect algebra* is a kind of partial monoid [9, §2], but we do not need the extra complication here.

bilinear effect structure, to the fibres of $\mathsf{Stat}_{\mathscr{C}}$, using the universal property of the product of categories:

$$\begin{aligned}
+_X : \mathsf{Stat}(X)(-,I) \times \mathsf{Stat}(X)(=,I) &= \mathbf{Set}\big(\mathscr{C}(I,X),\mathscr{C}(-,I)\big) \times \mathbf{Set}\big(\mathscr{C}(I,X),\mathscr{C}(=,I)\big) \\
&\xRightarrow{(\cdot,\cdot)} \mathbf{Set}\big(\mathscr{C}(I,X),\mathscr{C}(-,I) \times \mathscr{C}(=,I)\big) \\
&\xRightarrow{\mathbf{Set}\left(\mathscr{C}(I,X),+\right)} \mathbf{Set}\big(\mathscr{C}(I,X),\mathscr{C}(-\otimes=,I)\big) \\
&\xRightarrow{=} \mathsf{Stat}(X)(-\otimes=,I)
\end{aligned}$$

Here, $(\cdot,\cdot)$ denotes the pairing operation obtained from the universal property. In this way, each $\mathsf{Stat}(X)$ has bilinear effects. Note that this lifting is (strictly) compatible with the reindexing of Stat, so that $+_{(-)}$ defines an indexed natural transformation. This means in particular that *reindexing distributes over sums*: given state-dependent effects $g,g' : B \overset{Y}{\twoheadrightarrow} I$ and a channel $c : X \twoheadrightarrow Y$, we have $(g +_Y g')_c = g_c +_X g'_c$. We will thus generally omit the subscript from the lifted sum operation, and just write $+$.

We are now ready to construct the bicategory of statistical games.

Definition 3.2. Suppose $(\mathscr{C},\otimes,I)$ has bilinear effects, and let $\mathbf{BayesLens}_2$ denote the corresponding bicategory of (copy-composite) Bayesian lenses. We will write $\mathbf{SGame}_{\mathscr{C}}$ to denote the following *bicategory of (copy-composite) statistical games* in $\mathscr{C}$:

- The 0-cells are the 0-cells (X,A) of $\mathbf{BayesLens}_2$;
- the 1-cells, called *statistical games*, $(X,A) \to (Y,B)$ are pairs (c,L^c) of a 1-cell $c : (X,A) \twoheadrightarrow (Y,B)$ in $\mathbf{BayesLens}_2$ and a *loss* $L^c : B \overset{X}{\twoheadrightarrow} I$ in $\mathsf{Stat}(X)(B,I)$;
- given 1-cells $(c,L^c),(c',L^{c'}) : (X,A) \to (Y,B)$, the 2-cells $(c,L^c) \Rightarrow (c',L^{c'})$ are pairs (α,K^α) of a 2-cell $\alpha : c \Rightarrow c'$ in $\mathbf{BayesLens}_2$ and a loss $K^\alpha : B \overset{X}{\twoheadrightarrow} I$ such that $L^c = L^{c'} + K^\alpha$;
- the identity 2-cell on (c,L^c) is $(\mathsf{id}_c,0)$;
- given 2-cells $(\alpha,K^\alpha) : (c,L^c) \Rightarrow (c',L^{c'})$ and $(\alpha',K^{\alpha'}) : (c',L^{c'}) \Rightarrow (c'',L^{c''})$, their vertical composite is $(\alpha' \circ \alpha, K^{\alpha'} + K^\alpha)$, where $\circ$ here denotes vertical composition in $\mathbf{BayesLens}_2$;
- given 1-cells $(c,L^c) : (X,A) \to (Y,B)$ and $(d,L^d) : (Y,B) \to (Z,C)$, their horizontal composite is $(c \mathbin{\varphi} d, L^d_c + L^c \circ \overline{d}_c)$; and
  - given 2-cells $(\alpha,K^\alpha) : (c,L^c) \Rightarrow (c',L^{c'})$ and $(\beta,K^\beta) : (d,L^d) \Rightarrow (d',L^{d'})$, their horizontal composite is $(\beta \mathbin{\varphi} \alpha, K^\beta_c + K^\alpha \circ \overline{d}_c)$, where φ here denotes horizontal composition in $\mathbf{BayesLens}_2$.

Theorem 3.3. Definition 3.2 generates a well-defined bicategory.

The proof of this result is that $\mathbf{SGame}_{\mathscr{C}}$ is obtained via a pair of bicategorical Grothendieck constructions [5]: first to obtain Bayesian lenses; and then to attach the loss functions. The proof depends on the intermediate result that our effect monoids can be 'upgraded' to monoidal categories; we then use the delooping of this structure to associate (state-dependent) losses to (state-dependent) channels, after discarding the coparameters of the latter.

Lemma 3.4. Suppose $(\mathscr{C},\otimes,I)$ has bilinear effects. Then, for each object B, $\mathscr{C}(B,I)$ has the structure of a symmetric monoidal category. The objects of $\mathscr{C}(B,I)$ are its elements, the effects. If g,g' are two effects, then a morphism $\kappa : g \to g'$ is an effect such that $g = g' + \kappa$; the identity morphism for each effect id_g is then the constant 0 effect. Likewise, the tensor of two effects is their sum, and the corresponding unit is the constant 0. Precomposition by any morphism $c : A \twoheadrightarrow B$ preserves the monoidal category structure, making the presheaf $\mathscr{C}(-,I)$ into a fibrewise-monoidal indexed category $\mathscr{C}^{\mathrm{op}} \to \mathbf{MonCat}$.

As already indicated, this structure lifts to the fibres of Stat.

Corollary 3.5. For each object X in a category with bilinear effects, and for each object B, $\mathsf{Stat}(X)(B,I)$ inherits the symmetric monoidal structure of $\mathscr{C}(B,I)$; note that morphisms of state-dependent effects are likewise state-dependent, and that tensoring (summing) state-dependent effects involves copying the parameterizing state. Moreover, $\mathsf{Stat}(-)(=,I)$ is a fibrewise-monoidal indexed category $\sum_{X:\mathscr{C}^{\mathrm{op}}} \mathsf{Stat}(X)^{\mathrm{op}} \to \mathbf{MonCat}$.

3.2 Local inference models

In the context of approximate inference, one often does not have a single statistical model to evaluate, but a whole family of them. In particularly nice situations, this family is actually a subcategory $\mathscr{D}$ of $\mathscr{C}$, with the family of statistical models being all those that can be composed in $\mathscr{D}$. The problem of approximate inference can then be formalized as follows. Since both $\mathbf{BayesLens}_2$ and $\mathbf{SGame}_{\mathscr{C}}$ were obtained by bicategorical Grothendieck constructions, we have a pair of 2-fibrations $\mathbf{SGame}_{\mathscr{C}} \xrightarrow{\pi_{\mathsf{Loss}}} \mathbf{BayesLens}_2 \xrightarrow{\pi_{\mathsf{Lens}}} \mathbf{Copara}_2^l(\mathscr{C})$. Each of π_{Loss}, π_{Lens}, and the discarding functor $(-)^{\bar{\top}}$ can be restricted to the subcategory $\mathscr{D}$. The inclusion $\mathscr{D} \hookrightarrow \mathbf{Copara}_2^l(\mathscr{D})$ is a section of this restriction of $(-)^{\bar{\top}}$; the assignment of inversions to channels in $\mathscr{D}$ then corresponds to a 2-section of the 2-fibration π_{Lens} (restricted to $\mathscr{D}$); and the subsequent assignment of losses is a further 2-section of π_{Loss}. This situation is depicted in the following diagram of bicategories:

$$\begin{tikzcd}
\mathbf{SGame}_{\mathscr{D}} \arrow[r, hook] \arrow[d, "\pi_{\mathsf{Loss}}|_{\mathscr{D}}"] & \mathbf{SGame}_{\mathscr{C}} \arrow[d, "\pi_{\mathsf{Loss}}"] \\
\mathbf{BayesLens}_2|_{\mathscr{D}} \arrow[r, hook] \arrow[d, "\pi_{\mathsf{Lens}}|_{\mathscr{D}}"] \arrow[u, bend left] & \mathbf{BayesLens}_2 \arrow[d, "\pi_{\mathsf{Lens}}"] \\
\mathbf{Copara}_2^l(\mathscr{D}) \arrow[r, hook] \arrow[d, "\bar{\top}|_{\mathscr{D}}"] \arrow[u, bend left] & \mathbf{Copara}_2^l(\mathscr{C}) \arrow[d, "\bar{\top}"] \\
\mathscr{D} \arrow[r, hook] \arrow[u, bend left] & \mathscr{C}
\end{tikzcd}$$

This motivates the following definitions of *inference system* and *loss model*, although, for the sake of our examples, we will explicitly allow the loss-assignment to be lax. Before giving these new definitions, we recall the notion of *essential image* of a functor.

Definition 3.6 ([11])**.** Suppose $F : \mathscr{C} \to \mathscr{D}$ is an n-functor (a possibly weak homomorphism of weak n-categories). The *image* of F is the smallest sub-n-category of $\mathscr{D}$ that contains $F(\alpha)$ for all k-cells α in $\mathscr{C}$, along with any $(k+1)$-cells relating images of composites and composites of images, for all $0 \leqslant k \leqslant n$. We say that a sub-n-category $\mathscr{D}$ is *replete* if, for any k-cells α in $\mathscr{D}$ and β in $\mathscr{C}$ (with $0 \leqslant k < n$) such that $f : \alpha \Rightarrow \beta$ is a $(k+1)$-isomorphism in $\mathscr{C}$, then f is also a $(k+1)$-isomorphism in $\mathscr{D}$. The *essential image* of F, denoted $\mathsf{im}(F)$, is then the smallest replete sub-n-category of $\mathscr{D}$ containing the image of F.

Definition 3.7. Suppose $(\mathscr{C}, \otimes, I)$ is a copy-delete category. An *inference system* in $\mathscr{C}$ is a pair $(\mathscr{D}, \ell)$ of a subcategory $\mathscr{D} \hookrightarrow \mathscr{C}$ along with a section $\ell : \mathsf{im}(\iota) \to \mathbf{BayesLens}_2|_{\mathscr{D}}$ of $\pi_{\mathsf{Lens}}|_{\mathscr{D}}$, where $\mathsf{im}(\iota)$ is the essential image of the canonical lax inclusion $\iota : \mathscr{D} \hookrightarrow \mathbf{Copara}_2^l(\mathscr{D})$.

Definition 3.8. Suppose $(\mathscr{C}, \otimes, I)$ has bilinear effects and $\mathscr{B}$ is a subbicategory of $\mathbf{BayesLens}_2$. A *loss model* for $\mathscr{B}$ is a lax section L of the restriction $\pi_{\mathsf{Loss}}|_{\mathscr{B}}$ of π_{Loss} to $\mathscr{B}$. We say that L is a *strict* loss model if it is in fact a strict 2-functor, and a *strong* loss model if it is in fact a pseudofunctor.

Remark 3.9. We may often be interested in loss models for which $\mathscr{B}$ is in fact the essential image of an inference system, but we do not stipulate this requirement in the definition as it is not necessary for the following development.

Since lax functors themselves collect into categories, and using the monoidality of $+$, we have the following easy proposition that will prove useful below.

Proposition 3.10. Loss models for $\mathscr{B}$ constitute the objects of a symmetric monoidal category $\big(\mathsf{Loss}(\mathscr{B}), +, 0\big)$. The morphisms of $\mathsf{Loss}(\mathscr{B})$ are icons (identity component oplax transformations [6, §4.6]) between the corresponding lax functors, and they compose accordingly. The monoidal structure is given by sums of losses.

3.3 Examples

Each of our examples involves taking expectations of log-densities, and so to make sense of them it first helps to understand what we mean by "taking expectations".

Notation 3.11 (Expectations)**.** Written as a function, a density p on X has the type $X \to \mathbb{R}_+$; written as an effect, the type is $X \mathrel{-\!\!\bullet} I$. Given a measure or distribution π on X (equivalently, a state $\pi : I \mathrel{-\!\!\bullet} X$), we can compute the expectation of p under π as the composite $p \bullet \pi$. We write the resulting quantity as $\mathbb{E}_\pi[p]$, or more explicitly as $\mathbb{E}_{x\sim\pi}\big[p(x)\big]$. We can think of this expectation as representing the 'validity' (or truth value) of the 'predicate' p given the state π [12].

3.3.1 Relative entropy and Bayesian inference

For our first example, we return to the subject with which we opened this paper: the compositional structure of the relative entropy. We begin by giving a precise definition.

Definition 3.12. Suppose α, β are both measures on X, with α absolutely continuous with respect to β. Then the *relative entropy* or *Kullback-Leibler divergence* from α to β is the quantity $D_{KL}(\alpha,\beta) := \mathbb{E}_\alpha\left[\log \frac{\alpha}{\beta}\right]$, where $\frac{\alpha}{\beta}$ is the Radon-Nikodym derivative of α with respect to β.

Remark 3.13. When α and β admit density functions p_α and p_β with respect to the same base measure $\mathrm{d}x$, then $D_{KL}(\alpha,\beta)$ can equally be computed as $\mathbb{E}_{x\sim\alpha}\big[\log p_\alpha(x) - \log p_\beta(x)\big]$. It it this form that we will adopt henceforth.

Proposition 3.14. Let $\mathscr{B}$ be a subbicategory of simple lenses in $\mathbf{BayesLens}_2$, all of whose channels admit density functions with respect to a common measure and whose forward channels admit Bayesian inversion (and whose forward and backward coparameters coincide), and with only structural 2-cells. Then the relative entropy defines a strict loss model $\mathsf{KL} : \mathscr{B} \to \mathbf{SGame}$. Given a lens $(c, c') : (X, X) \nrightarrow (Y, Y)$, KL assigns the loss function $\mathsf{KL}(c, c') : Y \overset{X}{\mathrel{-\!\!\bullet}} I$ defined, for $\pi : I \mathrel{-\!\!\bullet} X$ and $y : Y$, by the relative entropy $\mathsf{KL}(c, c')_\pi(y) := D_{KL}\big(c'_\pi(y), c^\dagger_\pi(y)\big)$, where $c^\dagger$ is the exact inversion of c.

Successfully playing a relative entropy game entails minimizing the divergence from the approximate to the exact posterior. This divergence is minimized when the two coincide, and so KL represents a form of approximate Bayesian inference.

3.3.2 Maximum likelihood estimation

A statistical system may be more interested in predicting observations than updating beliefs. This is captured by the process of *maximum likelihood estimation*.

Definition 3.15. Let $(c, c') : (X, X) \nrightarrow (Y, Y)$ be a simple lens whose forward channel c admits a density function p_c. Then its *log-likelihood* is the loss function defined by $\mathsf{MLE}(c, c')_\pi(y) := -\log p_{c^\dagger \bullet \pi}(y)$.

Proposition 3.16. Let $\mathscr{B}$ be a subbicategory of lenses in $\mathbf{BayesLens}_2$ all of which admit density functions with respect to a common measure, and with only structural 2-cells. Then the assignment $(c,c') \mapsto \mathsf{MLE}(c,c')$ defines a lax loss model $\mathsf{MLE} : \mathscr{B} \to \mathbf{SGame}$.

Successfully playing a maximum likelihood game involves maximizing the log-likelihood that the system assigns to its observations $y : Y$. This process amounts to choosing a channel c that assigns high likelihood to likely observations, and thus encodes a valid model of the data distribution.

3.3.3 Autoencoders via the free energy

Many adaptive systems neither just infer nor just predict: they do both, building a model of their observations that they also invert to update their beliefs. In machine learning, such systems are known as *autoencoders*, as they 'encode' (infer) and 'decode' (predict), 'autoassociatively' [13]. In a Bayesian context, they are known as *variational autoencoders* [14], and their loss function is the *free energy* [15].

Definition 3.17. The *free energy* loss model is the sum of the relative entropy and the likelihood loss models: $\mathsf{FE} := \mathsf{KL} + \mathsf{MLE}$. Given a simple lens $(c,c') : (X,X) \nrightarrow (Y,Y)$ admitting Bayesian inversion and with densities, FE assigns the loss function

$$\begin{aligned}
\mathsf{FE}(c,c')_\pi(y) &= (\mathsf{KL}+\mathsf{MLE})(c,c')_\pi(y) \\
&= D_{KL}\big(c'_\pi(y), c^\dagger_\pi(y)\big) - \log p_{c^{\overline{\top}} \bullet \pi}(y)
\end{aligned}$$

Remark 3.18. Beyond its autoencoding impetus, another important property of the free energy is its improved computational tractability compared to either the relative entropy or the likelihood loss. This property is a consequence of the following fact: although obtained as the sum of terms which both depend on an expensive marginalization[4], the free energy itself does not. This can be seen by expanding the definitions of the relative entropy and of $c^\dagger_\pi$ and rearranging terms:

$$\begin{aligned}
\mathsf{FE}(c,c')_\pi(y) &= D_{KL}\big(c'_\pi(y), c^\dagger_\pi(y)\big) - \log p_{c^{\overline{\top}} \bullet \pi}(y) \\
&= \mathop{\mathbb{E}}_{(x,m)\sim c'_\pi(y)} \big[\log p_{c'_\pi}(x,m|y) - \log p_{c^\dagger_\pi}(x,m|y)\big] - \log p_{c^{\overline{\top}} \bullet \pi}(y) \\
&= \mathop{\mathbb{E}}_{(x,m)\sim c'_\pi(y)} \big[\log p_{c'_\pi}(x,m|y) - \log p_{c^\dagger_\pi}(x,m|y) - \log p_{c^{\overline{\top}} \bullet \pi}(y)\big] \\
&= \mathop{\mathbb{E}}_{(x,m)\sim c'_\pi(y)} \Big[\log p_{c'_\pi}(x,m|y) - \log \frac{p_c(m,y|x)\,p_\pi(x)}{p_{c^{\overline{\top}} \bullet \pi}(y)} - \log p_{c^{\overline{\top}} \bullet \pi}(y)\Big] \\
&= \mathop{\mathbb{E}}_{(x,m)\sim c'_\pi(y)} \big[\log p_{c'_\pi}(x,m|y) - \log p_c(m,y|x) - \log p_\pi(x)\big] \\
&= D_{KL}\big(c'_\pi(y), \pi \otimes \mathbf{1}\big) - \mathop{\mathbb{E}}_{(x,m)\sim c'_\pi(y)} \big[\log p_c(m,y|x)\big]
\end{aligned}$$

Here, $\mathbf{1}$ denotes the measure with density 1 everywhere. Note that when the coparameter is trivial, $\mathsf{FE}(c,c')_\pi(y)$ reduces to

$$D_{KL}\big(c'_\pi(y), \pi\big) - \mathop{\mathbb{E}}_{x\sim c'_\pi(y)} \big[\log p_c(y|x)\big] .$$

Remark 3.19. The name *free energy* is due to an analogy with the Helmholtz free energy in thermodynamics, as we can write it as the difference between an (expected) energy and an entropy

[4]Evaluating the pushforward $c^{\overline{\top}} \bullet \pi$ involves marginalizing over the intermediate variable; and evaluating $c^\dagger_\pi(y)$ also involves evaluating $c^{\overline{\top}} \bullet \pi$.

term:

$$\begin{aligned}\mathsf{FE}(c,c')_\pi(y) &= \mathop{\mathbb{E}}_{(x,m)\sim c'_\pi(y)}\left[-\log p_c(m,y|x) - \log p_\pi(x)\right] - S_{X\otimes M}\left[c'_\pi(y)\right] \\ &= \mathop{\mathbb{E}}_{(x,m)\sim c'_\pi(y)}\left[E_{(c,\pi)}(x,m,y)\right] - S_{X\otimes M}\left[c'_\pi(y)\right] \;= U - TS\end{aligned}$$

where we call $E_{(c,\pi)} : X \otimes M \otimes Y \xrightarrow{X} I$ the *energy*, and where $S_{X\otimes M} : I \xrightarrow{X\otimes M} I$ is the Shannon entropy. The last equality makes the thermodynamic analogy: U here is the *internal energy* of the system; $T = 1$ is the *temperature*; and S is again the entropy.

3.3.4 The Laplace approximation

Although optimizing the free energy does not necessitate access to exact inversions, it is still necessary to compute an expectation under the approximate inversion, and unless one chooses wisely[5], this might also be difficult. One such wise choice established in the computational neuroscience literature is the Laplace approximation [17], in which one assumes Gaussian channels and posteriors with small variance. Under these conditions, the expectations can be approximated away.

Definition 3.20. We will say that a channel c is *Gaussian* if $c(x)$ is a Gaussian measure for every x in its domain. We will denote the mean and variance of $c(x)$ by $\mu_c(x)$ and $\Sigma_c(x)$ respectively.

Proposition 3.21 (Laplace approximation)**.** Let the ambient category of channels $\mathscr{C}$ be restricted to that generated by Gaussian channels between finite-dimensional Cartesian spaces, and let $\mathscr{B}$ denote the corresponding restriction of $\mathbf{BayesLens}_2$. Suppose $(\gamma,\rho) : (X,X) \nrightarrow (Y,Y)$ is such a lens, for which, for all $y : Y$ and Gaussian priors $\pi : I \nrightarrow X$, the eigenvalues of $\Sigma_{\rho_\pi}(y)$ are small. Then the free energy $\mathsf{FE}(\gamma,\rho)_\pi(y)$ can be approximated by the *Laplacian free energy*

$$\begin{aligned}\mathsf{FE}(\gamma,\rho)_\pi(y) &\approx \mathsf{LFE}(\gamma,\rho)_\pi(y) && (1)\\ &:= E_{(\gamma,\pi)}\left(\mu_{\rho_\pi}(y),y\right) - S_{X\otimes M}\left[\rho_\pi(y)\right] && (2)\\ &= -\log p_\gamma(\mu_{\rho_\pi}(y),y) - \log p_\pi(\mu_{\rho_\pi}(y)|_X) - S_{X\otimes M}\left[\rho_\pi(y)\right]\end{aligned}$$

where we have written the argument of the density p_γ in 'function' style; where $(-)_X$ denotes the projection onto X; and where $S_{X\otimes M}[\rho_\pi(y)] = \mathbb{E}_{(x,m)\sim\rho_\pi(y)}[-\log p_{\rho_\pi}(x,m|y)]$ is the Shannon entropy of $\rho_\pi(y)$. The approximation is valid when Σ_{ρ_π} satisfies

$$\Sigma_{\rho_\pi}(y) = \left(\partial^2_{(x,m)} E_{(\gamma,\pi)}\right)\left(\mu_{\rho_\pi}(y),y\right)^{-1}. \qquad (3)$$

We call $E_{(\gamma,\pi)}$ the *Laplacian energy*.

Remark 3.22. The usual form of the Laplace model in the literature omits the coparameters. It is of course easy to recover the non-coparameterized form by taking $M = 1$.

Proposition 3.23. Let $\mathscr{B}$ be a subbicategory of $\mathbf{BayesLens}_2$ of Gaussian lenses whose backward channels have small variance. Then LFE defines a lax loss model $\mathscr{B} \to \mathbf{SGame}$.

Effectively, this proposition says that, under the stated conditions, the free energy and the Laplacian free energy coincide. Consequently, successfully playing a Laplacian free energy game has the same autoencoding effect.

[5] In machine learning, optimizing variational autoencoders using stochastic gradient descent typically involves a "reparameterization trick" [16, §2.5].

4 Future work

This paper only scratches the surface of the structure of statistical games. One avenue for further investigation is the link between this structure and the similar structure of diegetic open (economic) games [18], a recent reformulation of compositional game theory [19]. Notably, the composition rule for loss functions appears closely related to the Bellman equation, suggesting that algorithms for approximate inference (such as expectation-maximization) and reinforcement learning (such as backward induction) are more than superficially similar.

Another avenue for further investigation concerns mathematical neatness. First, we seek an abstract characterization of copy-composition and $\mathbf{Copara}_2$; it has been suggested to us[6] that the computation by compilers of "static single-assignment form" [20] by compilers may have a similar structure. Second, the explicit constraint defining simple coparameterized Bayesian lenses is inelegant; we expect that using dependent optics [21–23] may help to encode this constraint in the type signature, at the cost of higher-powered mathematical machinery. Finally, we seek further examples of loss models, and more abstract (and hopefully universal) characterizations of those we already have; for example, it is known that the Shannon entropy has a topological origin [24] via a "nonlinear derivation" [25], and we expect that we can follow this connection further.

5 References

[1] Toby St Clere Smithe. "Cyber Kittens, or Some First Steps Towards Categorical Cybernetics". In: *Electronic Proceedings in Theoretical Computer Science*. Applied Category Theory 2020. Vol. 333. 02/08/2021, pp. 108–124. DOI: 10.4204/EPTCS.333.8. (Visited on 09/29/2023).

[2] Toby St Clere Smithe. *Bayesian Updates Compose Optically*. 05/31/2020. arXiv: 2006.01631 [math.CT]. preprint.

[3] 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.

[4] Matteo Capucci et al. "Towards Foundations of Categorical Cybernetics". In: *Electronic Proceedings in Theoretical Computer Science*. Applied Category Theory 2021. Vol. 372. 11/03/2022, pp. 235–248. DOI: 10.4204/EPTCS.372.17. (Visited on 09/29/2023).

[5] Igor Baković. "Fibrations of Bicategories". In: (2010). URL: https://www2.irb.hr/korisnici/ibakovic/groth2fib.pdf.

[6] Niles Johnson and Donald Yau. *2-Dimensional Categories*. 06/17/2020. arXiv: 2002.06055 [math]. URL: http://arxiv.org/abs/2002.06055 (visited on 11/03/2022). preprint.

[7] Joe Moeller and Christina Vasilakopoulou. "Monoidal Grothendieck Construction". In: *Theory and Applications of Categories* 35.31 (2020), pp. 1159–1207. arXiv: 1809.00727v2 [math.CT].

[6] This suggestion is due to Owen Lynch.

[8] David I. Spivak. *Generalized Lens Categories via Functors C Op → Cat*. 08/06/2019. arXiv: `1908.02202 [math.CT]`. preprint.

[9] Bart Jacobs and Bram Westerbaan. "An Effect-Theoretic Account of Lebesgue Integration". In: *Electronic Notes in Theoretical Computer Science* 319 (12/2015), pp. 239–253. DOI: `10.1016/j.entcs.2015.12.015`.

[10] Matthijs Vákár and Luke Ong. *On S-Finite Measures and Kernels*. 10/03/2018. arXiv: `1810.01837 [math.PR]`. preprint.

[11] nLab authors. *Essential Image*. 05/2023. URL: `https://ncatlab.org/nlab/show/essential+image`.

[12] Bart Jacobs and Fabio Zanasi. "The Logical Essentials of Bayesian Reasoning". In: *Foundations of Probabilistic Programming*. Ed. by Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva. 1st ed. Cambridge University Press, 12/03/2020, pp. 295–332. ISBN: 978-1-108-77075-0 978-1-108-48851-8. DOI: `10.1017/9781108770750.010`. (Visited on 09/29/2023).

[13] Mark A. Kramer. "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks". In: *AIChE Journal* 37.2 (02/1991), pp. 233–243. ISSN: 0001-1541, 1547-5905. DOI: `10.1002/aic.690370209`. (Visited on 05/02/2023).

[14] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 12/20/2013. arXiv: `1312.6114 [stat.ML]`. preprint.

[15] Peter Dayan et al. "The Helmholtz Machine". In: *Neural Computation* 7.5 (09/1995), pp. 889–904. DOI: `10.1162/neco.1995.7.5.889`. URL: `http://dx.doi.org/10.1162/neco.1995.7.5.889`.

[16] Diederik P. Kingma. "Variational Inference & Deep Learning. A New Synthesis". PhD thesis. University of Amsterdam, 2017. URL: `https://hdl.handle.net/11245.1/8e55e07f-e4be-458f-a929-2f9bc2d169e8`.

[17] K. Friston et al. "Variational Free Energy and the Laplace Approximation". In: *Neuroimage* 34.1 (01/2007), pp. 220–234. DOI: `10.1016/j.neuroimage.2006.08.035`. pmid: `17055746`.

[18] Matteo Capucci. "Diegetic Representation of Feedback in Open Games". In: *Electronic Proceedings in Theoretical Computer Science*. Applied Category Theory 2022. Vol. 380. 08/07/2023, pp. 145–158. DOI: `10.4204/EPTCS.380.9`. (Visited on 09/29/2023).

[19] Neil Ghani et al. "Compositional Game Theory". In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science. Oxford United Kingdom: ACM, 07/09/2018, pp. 472–481. ISBN: 978-1-4503-5583-4. DOI: `10.1145/3209108.3209165`. (Visited on 09/29/2023).

[20] Richard A. Kelsey. “A Correspondence between Continuation Passing Style and Static Single Assignment Form”. In: *Papers from the 1995 ACM SIGPLAN Workshop on Intermediate Representations*. POPL95: 22nd ACM Symposium on Principles of Programming Languages. San Francisco California USA: ACM, 03/1995, pp. 13–22. ISBN: 978-0-89791-754-4. DOI: `10.1145/202529.202532`. (Visited on 05/02/2023).

[21] Dylan Braithwaite et al. *Fibre Optics*. 12/21/2021. arXiv: `2112.11145 [math.CT]`. preprint.

[22] Pietro Vertechi. “Dependent Optics”. In: *Electronic Proceedings in Theoretical Computer Science*. Applied Category Theory 2022. Vol. 380. 08/07/2023, pp. 128–144. DOI: `10.4204/EPTCS.380.8`. (Visited on 09/29/2023).

[23] Matteo Capucci. *Seeing Double through Dependent Optics*. 04/22/2022. arXiv: `2204.10708 [math.CT]`. preprint.

[24] Tai-Danae Bradley. “Entropy as a Topological Operad Derivation”. In: *Entropy* 23.9 (09/09/2021), p. 1195. ISSN: 1099-4300. DOI: `10.3390/e23091195`. URL: `https://www.mdpi.com/1099-4300/23/9/1195` (visited on 05/02/2023).

[25] Tom Leinster. *Entropy and Diversity: The Axiomatic Approach*. 1st ed. Cambridge University Press, 04/30/2021. ISBN: 978-1-108-96355-8 978-1-108-83270-0 978-1-108-96557-6. DOI: `10.1017/9781108963558`. (Visited on 09/29/2023).

A State-dependent channels

In this section, we review the indexed category $\mathsf{Stat} : \mathscr{C}^{\mathrm{op}} \to \mathbf{Cat}$ of state-dependent channels in $\mathscr{C}$, from which Bayesian lenses are obtained. We can think of Stat as a decategorified, non-coparameterized, version of Stat_2, in which the hom-sets $\mathsf{Stat}(X)(A,B)$ of each fibre are given by $\mathbf{Set}\big(\mathscr{C}(I,X),\mathscr{C}(A,B)\big)$. Reindexing is again by pre-composition, although simplified as there are now no coparameters to discard.

Definition A.1. Let $(\mathscr{C},\otimes,I)$ be a monoidal category. Define the *$\mathscr{C}$-state-indexed* category $\mathsf{Stat} : \mathscr{C}^{\mathrm{op}} \to \mathbf{Cat}$ as follows.

$$\mathsf{Stat} \;:\; \mathscr{C}^{\mathrm{op}} \to \mathbf{Cat}$$

$$X \mapsto \mathsf{Stat}(X) := \left(\begin{array}{rrcl} & \mathsf{Stat}(X)_0 & := & \mathscr{C}_0 \\ & \mathsf{Stat}(X)(A,B) & := & \mathbf{Set}\big(\mathscr{C}(I,X),\mathscr{C}(A,B)\big) \\ \mathsf{id}_A : & \mathsf{Stat}(X)(A,A) & := & \begin{cases} \mathsf{id}_A : \mathscr{C}(I,X) \to \mathscr{C}(A,A) \\ \qquad \rho \quad \mapsto \quad \mathsf{id}_A \end{cases} \end{array} \right) \tag{4}$$

$$f : \mathscr{C}(Y,X) \mapsto \left(\begin{array}{lccc} \mathsf{Stat}(f) : & \mathsf{Stat}(X) & \to & \mathsf{Stat}(Y) \\ & \mathsf{Stat}(X)_0 & = & \mathsf{Stat}(Y)_0 \\ & \mathbf{Set}(\mathscr{C}(I,X),\mathscr{C}(A,B)) & \to & \mathbf{Set}\big(\mathscr{C}(I,Y),\mathscr{C}(A,B)\big) \\ & \alpha & \mapsto & f^*\alpha : \big(\sigma : \mathscr{C}(I,Y)\big) \mapsto \big(\alpha(f \bullet \sigma) : \mathscr{C}(A,B)\big) \end{array} \right)$$

Composition in each fibre $\mathsf{Stat}(X)$ is as in $\mathscr{C}$. Explicitly, indicating morphisms $\mathscr{C}(I,X) \to \mathscr{C}(A,B)$ in

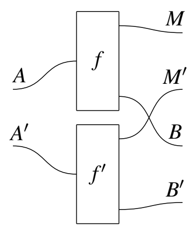
$\mathsf{Stat}(X)$ by $A \overset{X}{\bullet\!\!\!\to} B$, and given $\alpha : A \overset{X}{\bullet\!\!\!\to} B$ and $\beta : B \overset{X}{\bullet\!\!\!\to} C$, their composite is $\beta \circ \alpha : A \overset{X}{\bullet\!\!\!\to} C := \rho \mapsto \beta(\rho) \bullet \alpha(\rho)$, where here we indicate composition in $\mathscr{C}$ by $\bullet$ and composition in the fibres $\mathsf{Stat}(X)$ by $\circ$. Given $f : Y \bullet\!\!\!\to X$ in $\mathscr{C}$, the induced functor $\mathsf{Stat}(f) : \mathsf{Stat}(X) \to \mathsf{Stat}(Y)$ acts by pre-composition.

The category of non-coparameterized Bayesian lenses is then obtained as the (1-categorical) Grothendieck construction of the pointwise opposite of Stat, following Spivak [8].

B Monoidal statistical games

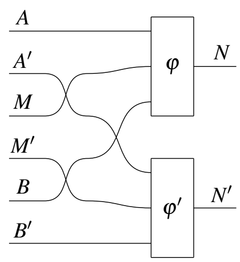
In this section, we exhibit the monoidal structures on copy-composite Bayesian lenses, statistical games, and loss models, as well as demonstrating that each of our loss models is accordingly monoidal. We begin by expressing the monoidal structure on $\mathbf{Copara}_2(\mathscr{C})$.

Proposition B.1. If the monoidal structure on $\mathscr{C}$ is symmetric, then $\mathbf{Copara}_2(\mathscr{C})$ inherits a monoidal structure $(\otimes, I)$, with the same unit object I as in $\mathscr{C}$. On 1-cells $f : A \underset{M}{\longrightarrow} B$ and $f' : A' \underset{M'}{\longrightarrow} B'$, the tensor $f \otimes f' : A \otimes A' \underset{M \otimes M'}{\longrightarrow} B \otimes B'$ is defined by



.

On 2-cells $\varphi : f \Rightarrow g$ and $\varphi' : f' \Rightarrow g'$, the tensor $\varphi \otimes \varphi' : (f \otimes f') \Rightarrow (g \otimes g')$ is given by the string diagram



.

Next, we define the notion of monoidal indexed bicategory.

Definition B.2. Suppose $(\mathscr{B}, \otimes, I)$ is a monoidal bicategory. We will say that $F : \mathscr{B}^{\mathrm{coop}} \to \mathbf{Bicat}$ is a *monoidal indexed bicategory* when it is equipped with the structure of a weak monoid object in the 3-category of indexed bicategories, indexed pseudofunctors, indexed pseudonatural transformations, and indexed modifications.

More explicitly, we will take F to be a monoidal indexed bicategory when it is equipped with

(i) an indexed pseudofunctor $\mu : F(-) \times F(=) \to F(- \otimes =)$ called the *multiplication*, *i.e.*,
   (a) a family of pseudofunctors $\mu_{X,Y} : FX \times FY \to F(X \otimes Y)$, along with
   (b) for any 1-cells $f : X \to X'$ and $g : Y \to Y'$ in $\mathscr{B}$, a pseudonatural isomorphism $\mu_{f,g} : \mu_{X',Y'} \circ (Ff \times Fg) \Rightarrow F(f \otimes g) \circ \mu_{X,Y}$;

(ii) a pseudofunctor $\eta : \mathbf{1} \to FI$ called the *unit*;

as well as three indexed pseudonatural isomorphisms — an associator, a left unitor, and a right unitor — which satisfy weak analogues of the coherence conditions for a monoidal indexed category [7, §3.2], up to invertible indexed modifications.

Using this notion, we can establish that Stat_2 is monoidal. (So as to demonstrate the structure, we do not omit the proof sketch.)

Theorem B.3. Stat_2 is a monoidal indexed bicategory.

Proof sketch. The multiplication μ is given first by the family of pseudofunctors $\mu_{X,Y} : \mathsf{Stat}_2(X) \times \mathsf{Stat}_2(Y) \to \mathsf{Stat}_2(X \otimes Y)$ which are defined on objects simply by tensor

$$\mu_{X,Y}(A,B) = A \otimes B$$

since the objects do not vary between the fibres of Stat_2, and on hom categories by the functors

$$\begin{aligned}
&\mathsf{Stat}_2(X)(A,B) \times \mathsf{Stat}_2(Y)(A',B') \\
&= \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X), \mathbf{Copara}_2^r(\mathscr{C})(A,B)\big) \times \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,Y), \mathbf{Copara}_2^r(\mathscr{C})(A',B')\big) \\
&\cong \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X) \times \mathsf{disc}\,\mathscr{C}(I,Y), \mathbf{Copara}_2^r(\mathscr{C})(A,B) \times \mathbf{Copara}_2^r(\mathscr{C})(A',B')\big) \\
&\xrightarrow{\mathbf{Cat}(\mathsf{disc}\,\mathscr{C}(I,\mathsf{proj}_X) \times \mathsf{disc}\,\mathscr{C}(I,\mathsf{proj}_Y),\otimes)} \mathbf{Cat}\big(\mathsf{disc}\,\mathscr{C}(I,X\otimes Y)^2, \mathbf{Copara}_2^r(\mathscr{C})(A\otimes A', B\otimes B')\big) \\
&\xrightarrow{\mathbf{Cat}(\blacktriangledown,\mathsf{id})} \mathbf{Cat}(\mathsf{disc}\,\mathscr{C}(I,X\otimes Y), \mathbf{Copara}_2^r(C)(A\otimes A', B\otimes B') \\
&= \mathsf{Stat}_2(X\otimes Y)(A\otimes A', B\otimes B')\ .
\end{aligned}$$

where $\mathbf{Cat}(\blacktriangledown,\mathsf{id})$ indicates pre-composition with the universal (Cartesian) copying functor. For all $f : X \to X'$ and $g : Y \to Y'$ in $\mathbf{Copara}_2^l(\mathscr{C})$, the pseudonatural isomorphisms

$$\mu_{f,g} : \mu_{X',Y'} \circ \big(\mathsf{Stat}_2(f) \times \mathsf{Stat}_2(g)\big) \Rightarrow \mathsf{Stat}_2(f \otimes g) \circ \mu_{X,Y}$$

are obtained from the universal property of the product $\times$ of categories. The unit $\eta : \mathbf{1} \to \mathsf{Stat}_2(I)$ is the pseudofunctor mapping the unique object of $\mathbf{1}$ to the monoidal unit I. Associativity and unitality of this monoidal structure follow from the functoriality of the construction, given the monoidal structures on $\mathscr{C}$ and $\mathbf{Cat}$. □

Just as the monoidal Grothendieck construction induces a monoidal structure on categories of lenses for monoidal pseudofunctors [7], we obtain a monoidal structure on the bicategory of copy-composite bayesian lenses.

Corollary B.4. The bicategory of copy-composite Bayesian lenses $\mathbf{BayesLens}_2$ is a monoidal bicategory. The monoidal unit is the object (I,I). The tensor $\otimes$ is given on 0-cells by $(X,A) \otimes (X',A') := (X \otimes X', A \otimes A')$, and on hom-categories by

$$\begin{aligned}
&\mathbf{BayesLens}_2\big((X,A),(Y,B)\big) \times \mathbf{BayesLens}_2\big((X,A),(Y,B)\big) \\
&= \mathbf{Copara}_2^l(\mathscr{C})(X,Y) \times \mathsf{Stat}_2(X)(B,A) \times \mathbf{Copara}_2^l(\mathscr{C})(X',Y') \times \mathsf{Stat}_2(X')(B',A') \\
&\xrightarrow{\sim} \mathbf{Copara}_2^l(\mathscr{C})(X,Y) \times \mathbf{Copara}_2^l(\mathscr{C})(X',Y') \times \mathsf{Stat}_2(X)(B,A) \times \mathsf{Stat}_2(X')(B',A') \\
&\xrightarrow{\otimes \times \mu^{\mathrm{op}}_{X,X'}} \mathbf{Copara}_2^l(\mathscr{C})(X\otimes X', Y\otimes Y') \times \mathsf{Stat}_2(X\otimes X')(B\otimes B', A\otimes A') \\
&= \mathbf{BayesLens}_2\big((X,A)\otimes(X',A'), (Y,B)\otimes(Y',B')\big)\ .
\end{aligned}$$

And similarly, we obtain a monoidal structure on statistical games.

Proposition B.5. The bicategory of copy-composite statistical games **SGame** is a monoidal bicategory. The monoidal unit is the object (I,I). The tensor $\otimes$ is given on 0-cells as for the tensor of Bayesian lenses, and on hom-categories by

$$\begin{aligned}
&\mathbf{SGame}\big((X,A),(Y,B)\big) \times \mathbf{SGame}\big((X',A'),(Y',B')\big) \\
&= \mathbf{BayesLens}_2\big((X,A),(Y,B)\big) \times \mathsf{Stat}(X)(B,I) \\
&\qquad \times \mathbf{BayesLens}_2\big((X',A'),(Y',B')\big) \times \mathsf{Stat}(X')(B',I) \\
&\xrightarrow{\sim} \mathbf{BayesLens}_2\big((X,A),(Y,B)\big) \times \mathbf{BayesLens}_2\big((X',A'),(Y',B')\big) \\
&\qquad \times \mathsf{Stat}(X)(B,I) \times \mathsf{Stat}(X')(B',I) \\
&\xrightarrow{\otimes \times \mu_{X,X'}} \mathbf{BayesLens}_2\big((X,A)\otimes(X',A'),(Y,B)\otimes(Y',B')\big) \times \mathsf{Stat}(X\otimes X')(B\otimes B',I\otimes I) \\
&\xrightarrow{\sim} \mathbf{SGame}\big((X,A)\otimes(X',A'),(Y,B)\otimes(Y',B')\big)
\end{aligned}$$

where here μ indicates the multiplication of the monoidal structure on Stat [**Smithe2022Mathematical**].

We give natural definitions of monoidal inference system and monoidal loss model, which we elaborate below.

Definition B.6. A *(lax) monoidal inference system* is an inference system $(\mathscr{D},\ell)$ for which ℓ is a lax monoidal pseudofunctor. A *(lax) monoidal loss model* is a loss model L which is a lax monoidal lax functor.

Remark B.7. We say 'lax' whenever a morphism (of any structure) only weakly preserves a monoidal operation such as composition of any order; this includes as a special case lax monoidal functors (since a monoidal category is a one-object bicategory). In this respect, we differ from [7, §2.2], who use 'weak' in the latter case; we prefer to maintain consistency. Following [6, Def. 4.2.1], we will continue to say *lax* when the witness to laxness maps composites of images to images of composites (and *oplax* when the witness maps images of composites to composites of images).

These conventions mean that a loss model $L : \mathscr{B} \to \mathbf{SGame}$ is lax monoidal when it is equipped with strong transformations

$$\begin{tikzcd}
\mathscr{B}\times\mathscr{B} \arrow[r, "L\times L"] \arrow[d, "\otimes_{\mathscr{B}}"'] \arrow[dr, phantom, "\swarrow\lambda"] & \mathbf{SGame}\times\mathbf{SGame} \arrow[d, "\otimes_{\mathbf{G}}"] \\
\mathscr{B} \arrow[r, "L"'] & \mathbf{SGame}
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
\mathbf{1} \arrow[d, "(I,I)"'] \arrow[dr, bend left, "(I,I)"] \arrow[dr, phantom, "\swarrow\lambda_0"] & \\
\mathscr{B} \arrow[r, "L"'] & \mathbf{SGame}
\end{tikzcd}$$

where $\otimes_{\mathscr{B}}$ and $\otimes_{\mathbf{G}}$ denote the monoidal products on $\mathscr{B} \hookrightarrow \mathbf{BayesLens}_2$ and **SGame** respectively, and when λ and λ_0 are themselves equipped with invertible modifications satisfying coherence axioms, as in Moeller and Vasilakopoulou [7, §2.2].

Note that, because L must be a (lax) section of the 2-fibration $\pi_{\mathsf{Loss}}|_{\mathscr{B}} : \mathbf{SGame}|_{\mathscr{B}} \to \mathscr{B}$, the unitor λ_0 is forced to be trivial, picking out the identity on the monoidal unit (I,I). Likewise, the laxator $\lambda : L(-)\otimes L(=) \Rightarrow L(-\otimes=)$ must have 1-cell components which are identities:

$$L(X,A)\otimes L(X',A) = (X,A)\otimes(X',A') = (X\otimes X',A\otimes A') = L\big((X,A)\otimes L(X',A)\big)$$

The interesting structure is therefore entirely in the 2-cells. We follow the convention of [6, Def. 4.2.1] that a strong transformation is a lax transformation with invertible 2-cell components. Supposing that $(c,c') : (X,A) \twoheadrightarrow (Y,B)$ and $(d,d') : (X',A') \twoheadrightarrow (Y',B')$ are 1-cells in $\mathscr{B}$, the corresponding 2-cell component of λ has the form $\lambda_{c,d} : L\big((c,c')\otimes(d,d')\big) \Rightarrow L(c,c')\otimes L(d,d')$, hence filling the following square in **SGame**:

$$\begin{tikzcd}
(X,A)\otimes(X',A') \arrow[r, "L(c,c')\otimes L(d,d')"] \arrow[d, equal] & (Y,B)\otimes(Y',B') \arrow[d, equal] \\
(X,A)\otimes(X',A') \arrow[r, "L((c,c')\otimes(d,d'))"'] \arrow[ur, phantom, "\overset{\lambda_{c,d}}{\Rightarrow}"] & (Y,B)\otimes(Y',B')
\end{tikzcd}$$

Intuitively, these 2-cells witness the failure of the tensor $L(c,c')\otimes L(d,d')$ of the parts to account for correlations that may be evident to the "whole system" $L\big((c,c')\otimes(d,d')\big)$.

Just as we have monoidal lax functors, we can have monoidal lax transformations; again, see [7, §2.2].

Proposition B.8. Monoidal loss models and monoidal icons form a subcategory $\mathsf{MonLoss}(\mathscr{B})$ of $\mathsf{Loss}(\mathscr{B})$, and the symmetric monoidal structure $(+,0)$ on the latter restricts to the former.

B.1 Examples

In this section, we present the monoidal structure on the loss models considered above. Because loss models L are (lax) sections, following Remark B.7, this monoidal structure is given in each case by a lax natural family of 2-cells $\lambda_{c,d} : L\big((c,c')\otimes(d,d')\big) \Rightarrow L(c,c')\otimes L(d,d')$, for each pair of lenses $(c,c') : (X,A) \twoheadrightarrow (Y,B)$ and $(d,d') : (X',A') \twoheadrightarrow (Y',B')$. Such a 2-cell $\lambda_{c,d}$ is itself given by a loss function of type $B\otimes B' \xrightarrow[\bullet]{X\otimes X'} I$ satisfying the equation $L\big((c,c')\otimes(d,d')\big) = L(c,c')\otimes L(d,d') + \lambda_{c,d}$. Following [6, Eq. 4.2.3], lax naturality requires that λ satisfy the following equation of 2-cells, where K denotes the laxator (with respect to horizontal composition $\diamond$) with components $K(e,c) : Le \diamond Lc \Rightarrow L(e \obar c)$:

$$\begin{tikzcd}
& (Y,B)\otimes(Y',B') \arrow[dr, "L(e\otimes f)"] \arrow[d, phantom, "\Downarrow K(e\otimes f, c\otimes d)"] & \\
(X,A)\otimes(X',A') \arrow[ur, "L(c\otimes d)"] \arrow[rr, "L\big((e\obar c)\otimes(f\obar d)\big)" description] \arrow[rr, bend right=60, "L(e\obar c)\otimes L(f\obar d)"'] \arrow[rr, phantom, bend right=30, "\Downarrow \lambda(e\obar c, f\obar d)"] & {} & (Z,C)\otimes(Z',C')
\end{tikzcd}$$

$$=$$

$$\begin{tikzcd}
& (Y,B)\otimes(Y',B') \arrow[dr, "L(e\otimes f)"] \arrow[d, equal] & \\
(X,A)\otimes(X',A') \arrow[ur, "L(c\otimes d)"] \arrow[r, "Lc\otimes Ld"] \arrow[ur, phantom, shift right=3, "\Downarrow\lambda(c,d)"] \arrow[rr, bend right=60, "L(e\obar c)\otimes L(f\obar d)"'] \arrow[rr, phantom, bend right=30, "\Downarrow K(e,c)\otimes K(f,d)"] & (Y,B)\otimes(Y',B') \arrow[r, "Le\otimes Lf"] \arrow[ur, phantom, shift right=3, "\lambda(e,f)\Downarrow"] & (Z,C)\otimes(Z',C')
\end{tikzcd}$$

Since vertical composition in **SGame** is given on losses by $+$, we can write this equation as

$$\begin{aligned}
&\lambda(e \circ c, f \circ d) + K(e \otimes f, c \otimes d) \\
&\quad = \lambda(e,f) \diamond \lambda(c,d) + K(e,c) \otimes K(f,d) \\
&\quad = \lambda(e,f)_{c\otimes d} + \lambda(c,d) \circ (e' \otimes f')_{c\otimes d} + K(e,c) \otimes K(f,d) \,.
\end{aligned} \tag{5}$$

In each of the examples below, therefore, we establish the definition of the laxator λ and check that it satisfies equation 5.

We will often use the notation $(-)_X$ to denote projection onto a factor X of a monoidal product.

B.1.1 Relative entropy

Proposition B.9. The loss model KL of Proposition 3.14 is lax monoidal. Supposing that $(c,c') : (X,X) \nrightarrow (Y,Y)$ and $(d,d') : (X',X') \nrightarrow (Y',Y')$ are lenses in $\mathscr{B}$, the corresponding component $\lambda^{\mathsf{KL}}(c,d)$ of the laxator is given, for $\omega : I \twoheadrightarrow X \otimes X'$ and $(y,y') : Y \otimes Y'$, by

$$\lambda^{\mathsf{KL}}(c,d)_\omega(y,y') := \mathop{\mathbb{E}}_{\substack{(x,x',m,m')\sim \\ (c'_{\omega_X} \otimes d'_{\omega_{X'}})(y,y')}} \left[\log \frac{p_{\omega_X \otimes \omega_{X'}}(x,x')}{p_\omega(x,x')} \right] + \log \frac{p_{(c\otimes d)^\dagger \bullet \omega}(y,y')}{p_{(c\otimes d)^\dagger \bullet (\omega_X \otimes \omega_{X'})}(y,y')} \,.$$

B.1.2 Maximum likelihood estimation

Proposition B.10. The loss model MLE of Proposition 3.16 is lax monoidal. Supposing that $(c,c') : (X,X) \nrightarrow (Y,Y)$ and $(d,d') : (X',X') \nrightarrow (Y',Y')$ are lenses in $\mathscr{B}$, the corresponding component $\lambda^{\mathsf{MLE}}(c,d)$ of the laxator is given, for $\omega : I \twoheadrightarrow X \otimes X'$ and $(y,y') : Y \otimes Y'$, by

$$\lambda^{\mathsf{MLE}}(c,d)_\omega(y,y') := \log \frac{p_{(c\otimes d)^\dagger \bullet (\omega_X \otimes \omega_{X'})}(y,y')}{p_{(c\otimes d)^\dagger \bullet \omega}(y,y')} \,.$$

B.1.3 Free energy

Corollary B.11. The loss model FE of Definition 3.17 is lax monoidal. Supposing that $(c,c') : (X,X) \nrightarrow (Y,Y)$ and $(d,d') : (X',X') \nrightarrow (Y',Y')$ are lenses in $\mathscr{B}$, the corresponding component $\lambda^{\mathsf{FE}}(c,d)$ of the laxator is given, for $\omega : I \twoheadrightarrow X \otimes X'$ and $(y,y') : Y \otimes Y'$, by

$$\lambda^{\mathsf{FE}}(c,d)_\omega(y,y') := \mathop{\mathbb{E}}_{(x,x')\sim (c'_{\omega_X} \otimes d'_{\omega_{X'}})(y,y')} \left[\log \frac{p_{\omega_X \otimes \omega_{X'}}(x,x')}{p_\omega(x,x')} \right] .$$

B.1.4 Laplacian free energy

Proposition B.12. The loss model LFE of Propositions 3.21 and 3.23 is lax monoidal. Supposing that $(c,c') : (X,X) \nrightarrow (Y,Y)$ and $(d,d') : (X',X') \nrightarrow (Y',Y')$ are lenses in $\mathscr{B}$, the corresponding component $\lambda^{\mathsf{LFE}}(c,d)$ of the laxator is given, for $\omega : I \twoheadrightarrow X \otimes X'$ and $(y,y') : Y \otimes Y'$, by

$$\lambda^{\mathsf{LFE}}(c,d)_\omega(y,y') := \log \frac{p_{\omega_X \otimes \omega_{X'}}(\mu_{(c\otimes d)'_\omega}(y,y')_{XX'})}{p_\omega(\mu_{(c\otimes d)'_\omega}(y,y')_{XX'})}$$

where $\mu_{(c\otimes d)'_\omega}(y,y')_{XX'}$ is the $(X \otimes X')$-mean of the Gaussian distribution $(c'_{\omega_X} \otimes d'_{\omega_{X'}})(y,y')$.

Unifilar Machines and the Adjoint Structure of Bayesian Filtering

Nathaniel Virgo
Earth-Life Science Institute (ELSI), Tokyo Instutute of Technology
nathanielvirgo+science@gmail.com

We elucidate the mathematical structure of Bayesian filtering, and Bayesian inference more broadly, by applying recent work on category theoretical probability, specifically the concept of a strongly representable Markov category. We show that filtering, along with related concepts such as conjugate priors, arise from an adjunction: the process of taking a hidden Markov process is right adjoint to a forgetful functor. This has an interesting consequence. In practice, filtering is usually implemented using parametrised families of distributions. The Kalman filter is a particularly important example, which uses Gaussians. Rather than calculating a new posterior each time, the implementation only needs to udpate the parameters. This structure arises naturally from our adjunction; the correctness of such a model is witnessed by a map from the model into the system being modelled. Conjugate priors arise from this construction as a special case.

In showing this we define a notion of *unifilar machine*, which has its origins in the literature on ϵ-machines. Unifilar machines are useful as models of the 'observable behaviour' of stochastic systems; we show additionally that in the Kleisli category of the distribution monad there is a terminal unifilar machine, and its elements are controlled stochastic processes, mapping sequences of the input alphabet probabilistically to sequences of the output alphabet.

1 Introduction

This paper is concerned with the mathematical structure of *Bayesian filtering*, which is a common problem in applications of Bayesian inference. The idea is that there is some system with known dynamics (which in general are stochastic) but an unknown hidden state. The goal is to keep track of a Bayesian prior over the states of the system, updating it to a posterior whenever a new observation is made. This is useful if we want to be able to control the hidden state, as in solving a partially observable Markov decision process (POMDP), for example.

To reveal the underlying mathematical structure we make use of recent results in synthetic probability, which allows us to write proofs at the category theoretic level without using measure theory directly. We work in the framework of Markov categories [8], and in particular we make use of the concept of *strongly representable Markov category* as defined in [10]. Strongly representable Markov categories include **BorelStoch** (whose objects are standard Borel spaces and whose morphisms are Markov kernels) and the Kleisli category of the (real-valued) distribution monad, which we refer to as **Dist**. Therefore most of our results apply to both measure-theoretic probability and finitely supported probability.

We model a system with a hidden state as a certain kind of stochastic Moore machine (essentially a hidden Markov model); we refer to this informally as a *dynamical model* of the system. There is then a functor B that takes such a dynamical model and maps it to an *epistemic model*. This lives in a different category of machines that we call *unifilar machines*, whose outputs are stochastic but whose state updates are deterministic. Its state space consists of probability distributions over the hidden states of the system, and its dynamics are given by Bayesian updating.

S. Staton, C. Vasilakopoulou (Eds.):
Applied Category Theory 2023 (ACT2023)
EPTCS 397, 2023, pp. 299–317, doi:10.4204/EPTCS.397.18

Our main technical result is theorem 2.7, which states that this functor is right adjoint to a forgetful functor in the opposite direction. This has an interesting consequence. The functor B maps a dynamical model κ to what could be called its universal epistemic model, $B(\kappa)$. By this we mean that if we consider another unifilar machine α equipped with a morphism $\alpha \to B(\kappa)$, we can also consider this an epistemic model of κ, in a sense that we describe next.

In applications, one doesn't necessarily want to keep track of the Bayesian distribution directly. Instead, one uses a parametrised family of distributions, chosen such that the update step only needs to update the parameters to produce the posterior distribution. For this to work, the Bayesian posterior must always be in the same family of distributions as the prior. An example with enormous practical importance is the Kalman filter, example 2.8. Here the prior is a multivariate Gaussian and the posterior is always also a multivariate Gaussian. The filter's state space consists of the parameters of such a Gaussian, and the update step simply maps them to their new values. In our framework this kind of structure arises from considering a morphism $\alpha \to B(\kappa)$. The state space of $B(\kappa)$ consists of probability distributions, and the state space of α consists of values that parametrise them in a consistent way.

This idea is closely related to the notion of conjugate prior, which was previously studied in a category-theoretic context in [13]. The definition in that paper is essentially our eq. (24), which arises from our framework in a very natural way. The connection between Bayesian filtering and Bayesian inference is explored in section 2.1, where we also briefly touch on connections to recent work on de Finetti's theorem within a category-theoretic context [15, 9].

A secondary contribution of our paper is an exploration of the possible generalisations of Moore machines to the stochastic case. Our result involves two different generalisations of Moore machine, which we term *comb machine* (definition 2.2) and *unifilar machine* (definition 2.4). Unifilar machines in particular are of independent interest. They are based on an idea from the literature on ϵ-machines [2]. They are defined such that their output map is stochastic but their update map is (almost surely) deterministic given their input and their output. This means that their states map more directly to 'behaviours' than the states of a more general stochastic machine. Indeed, we show in section 3 that in **Dist** the category of unifilar machines has a terminal object, which consists of the collection of 'controlled stochastic processes,' also known as 'stochastic streams' [6]. In general, if a category of unifilar machines has a terminal object then it can be seen as an "object of behaviours" of stochastic systems.

Our Bayesian filtering machines have a strong resemblance to Bayesian lenses as presented in [20, 4] (see also [17], which develops the idea in a way that makes all of the relevant maps measurable). Our work is closely related, since the machine $B(\kappa)$ described above has the same data as a Bayesian lens. However, general unifilar machines do not seem to compose like lenses, and working out the precise relationship is a task for future work.

Our work also seems related to the notion of *determinisation* from automata theory. Here one takes a nondeterministic automaton (one that may have more than one transition from a given state, but with no notion of probability assigned to them) and turns it into a deterministic automaton whose state space is the power set of the original automaton. Determinisation has been studied and generalised in a coalgebraic context [19, 1]; understanding the exact relation to our work is also a task for future work.

Bayesian filtering and its connection to conjugate priors was previously considered in a Markov category context by the author and colleagues in [23]. A related approach to Bayesian filtering in Markov categories was presented at the conference [7]; the present work was developed independently. The novel contribution of the present paper is to reveal more of the abstract categorical structure underlying the idea, including the definitions of comb machine, unifilar machine and the adjoint structure involving the functor B, as well as the brief discussion of terminal unifilar machines in section 3.

1.1 Background on Representable Markov categories

We will use the machinery of representable Markov categories and in particular, strongly representable Markov categories, both defined in [10].[1] For general background on Markov categories we refer to [8]. Definitions in this section are from the literature, except for definition 1.2, which is a slight generalisation of the usual category-theoretic definition of 'almost surely.'

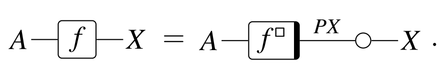
Recall from [10] that given an object X in a Markov category $\mathcal{C}$, a *distribution object* is an object PX equipped with a map $\mathrm{samp}_X\colon PX \to X$ such that for every morphism $f\colon A \to X$ there is a unique deterministic morphism $f^\square\colon A \to PX$ such that $f^\square \fatsemi \mathrm{samp}_\mathrm{X} = f$. A Markov category is then called *representable* if every object has a distribution object. Representable Markov categories often arise as the Kleisli categories of monads obeying conditions spelt out in [10].

The two examples we will use are **BorelStoch** (the Kleisli category of the Giry monad, restricted to standard Borel spaces) and the Kleisli category of the (real-valued) distribution monad, which we will call **Dist**. These are both shown to be representable in [10].

We recall also the following results about representable Markov categories: When every object has a distribution object, P extends to a functor $P\colon \mathcal{C} \to \mathcal{C}_\mathrm{det}$. Restricting the domain of this functor we obtain a functor $P_\mathrm{det}\colon \mathcal{C}_\mathrm{det} \to \mathcal{C}_\mathrm{det}$, which will also be written as P, except when we wish to explicitly disambiguate. The functor P_det can be made into a monad on $\mathcal{C}_\mathrm{det}$, and the Kleisli category of this monad is $\mathcal{C}$. The unit has components $\delta_X = \mathrm{id}_X^\square\colon X \to PX$, and the multiplication has components $\mu_X = P(\mathrm{samp}_X)\colon PPX \to PX$.

This monad arises from an adjunction: the functor P is right adjoint to the inclusion functor $\mathcal{C}_\mathrm{det} \hookrightarrow \mathcal{C}$. Its unit has components δ_X and its counit has components given by the sampling map $\mathrm{samp}_X\colon PX \to P$.

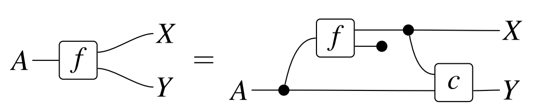
In string diagrams we will draw samp_X as a white dot. Additionally, if a morphism is known to be deterministic ([8], definition 2.2) we indicate this with a black bar at its right-hand edge, so we can write



(1)

We will need the definition of a strongly representable Markov category. For this we first recall some more definitions from [8] and [10].

Definition 1.1 (conditionals; [8], definition 11.5)**.** Let $f\colon A \to X \otimes Y$ be a morphism in a Markov category $\mathcal{C}$. We say that a morphism $c\colon X \otimes A \to Y$ is a *conditional* of f if



(2)

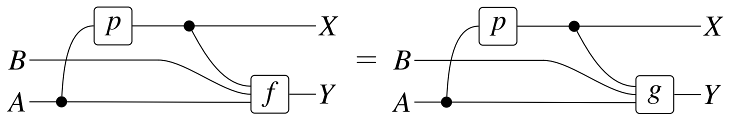
We say $\mathcal{C}$ *has conditionals* if every morphism of the appropriate type has a conditional.

The intuition is that for every value of the parameter A, the morphism f defines a joint distribution between X and Y, and eq. (2) represents a factorisation of this joint distribution into the marginal distribution over X and a conditional distribution of Y given X. Recall from [8] that **Dist** and **BorelStoch** both have conditionals, but **Stoch** does not. Conditionals are in general not unique when they exist; see [8], proposition 11.15 and the discussion surrounding it.

The following is essentially definition 13.1 of [8] ('almost surely'), but we generalise it slightly.

[1]The definitions of strongly representable Markov categories and 'deterministic given X' appear in version 2 of the arXiv preprint [10] but not in the published version of the same paper ([11]) or version 3 of the preprint. The author understands that the removal was for reasons of narrative structure rather than any technical defect.

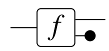
Definition 1.2 (generalised almost surely)**.** Given a morphism $p\colon A\otimes C\to X$ in a Markov category $\mathcal{C}$, we say morphisms $f,g\colon X\otimes A\to Y$ are *p-generalised-almost-surely equal* or *p-g.a.s. equal* if

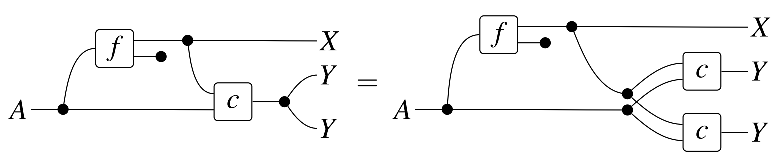


(3)

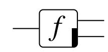
The idea is that in a measure-theoretic context such as **Stoch** or **BorelStoch**, for a values A, the values of f and g can differ only on subsets of X that have measure zero according to p. In **Dist** it means that $f(y\mid x,b,a)=g(y\mid x,b,a)$ whenever $p(x\mid a)>0$.

The category-theoretic definition of almost-surely in [8] allows p but not f or g to depend on a parameter, so this is a slight generalisation. We avoid calling our version "almost surely" to avoid confusion with the usual definition, as used in [8], [10] and other works.

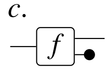
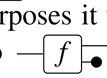
Definition 1.3 (deterministic given X; [10], definition 6.4)**.** Let $f\colon A\to X\otimes Y$ be a morphism in a Markov category $\mathcal{C}$, and let f be such that a conditional $c\colon X\otimes A\to Y$ exists. The morphism f is said to be *deterministic given* X if the conditional is  -generalised-almost-surely deterministic, in the sense that



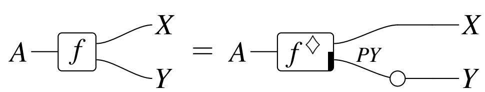
(4)

If $f\colon A\to X\otimes Y$ is known to be deterministic given X we write it as  .

In [10] it is shown that if eq. (4) holds for one conditional of f then it holds for all conditionals, so that this definition is independent of the choice of conditional c.

For both **BorelStoch** and **Dist**, if eq. (4) holds then c is  -generalised-almost-surely equal to a deterministic morphism ([10], example 6.12), so for most purposes it will not hurt to think of such conditionals as genuinely deterministic, though only defined up to  -g.a.s. equivalence.

Definition 1.4. [Strongly representable Markov category; [10], definition 6.7] A strongly representable Markov category is a representable Markov category in which for every morphism $f\colon A\to X\otimes Y$ there is a unique morphism $f^{\diamondsuit}\colon A\to X\otimes PY$ such that (i) $f^{\diamondsuit}$ is deterministic given X, and (ii)

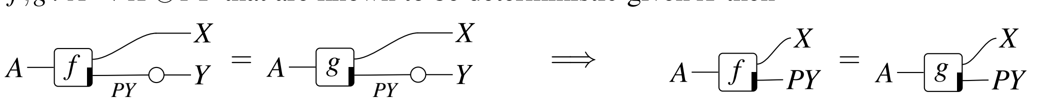


(5)

(This definition is less efficient than the one given in [10], which doesn't include an assumption that the category is representable, since this can be proven from weaker assumptions.) A strongly representable Markov category necessarily has conditionals, because $f^{\diamondsuit}$ has a conditional by the definition of deterministic given X, and if $c\colon X\times A\to PY$ is such a conditional then $c\,\mathbin{⨾}\,\mathrm{samp}_Y$ is a conditional of f.

BorelStoch is shown to be strongly representable in example 6.12 of [10]. For completeness we provide a proof that **Dist** is strongly representable in appendix A.1, where we also give an explicit construction for $f^{\diamondsuit}$ in **Dist**.

As a consequence of definition 1.4, in a strongly representable Markov category, if we have morphisms $f,g\colon A\to X\otimes PY$ that are known to be deterministic given X then



(6)

This will be used in the proofs of proposition 2.6 and theorem 2.7. The condition that f and g are deterministic given X is needed to cancel sampling maps in this way, since sampling maps are not usually epimorphisms.

2 Machines and Bayesian Filtering

The following definitions are all relative to a Markov category $(\mathcal{C}, \otimes, 1)$ and a choice of objects called the *input space* I and *output space* O, which we will assume to be fixed throughout this section.

For most of the following we will work with what we call "comb machines," which are a generalisation of Moore machines. However, many of the results also carry over to the case of Mealy machines, which we define first because they are simpler. The following definition is standard:

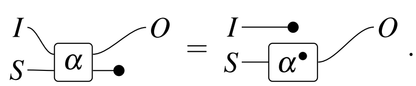
Definition 2.1 (Stochastic Mealy machine)**.** A *stochastic Mealy machine* is an object S of $\mathcal{C}$ called the *state space*, together with a morphism $\alpha\colon I\otimes S \to O\otimes S$ in $\mathcal{C}$. A morphism of Mealy machines $(S,\alpha)\to(T,\beta)$ is a morphism $f\colon S\to T$ in $\mathcal{C}$ such that $I\otimes S \xrightarrow{\alpha} O\otimes S \xrightarrow{\mathrm{id}_O\otimes f} O\otimes T = I\otimes S \xrightarrow{\mathrm{id}_I\otimes f} I\otimes T \xrightarrow{\beta} O\otimes T$. The category of Mealy machines will be written $\mathbf{Mealy}(I,O)$.

The idea is that a Mealy machine starts in some state in S, receives an input in I, and then produces an output in O while simultaneously transitioning to a new state. The output may depend on the input and may be correlated with the new state. We don't require morphisms of Mealy machines to be deterministic.

We now briefly discuss Moore machines and their generalisation to the stochastic context. In a Cartesian category, a Moore machine consists of a state space S and two maps: a *readout map* $S\to O$ and an *update map* $I\times S\to S$. An obvious way to generalise this to the stochastic case is to let both maps be stochastic, so that the update map has type $I\otimes S\to S$. However, machines with this definition tend not to be very well behaved, and in practice other definitions tend to be used.

One way to make stochastic Moore machines well behaved is to make the readout map deterministic. Machines of this kind can be expressed in terms of generalised lenses [21]; this is the approach taken in [18], for example. Intuitively, requiring a deterministic readout map allows the update map to "know" what the machine's last output was, since this can be inferred from the current value of S. However, for the present work we need the readout map to be stochastic, so we take a different approach:

Definition 2.2 (Comb machine)**.** A *comb machine* in a Markov category $\mathcal{C}$ is an object S of $\mathcal{C}$ (the state space), together with a morphism $\alpha\colon I\otimes S\to O\otimes S$ in $\mathcal{C}$ and a morphism $\alpha^\bullet\colon S\to O$ such that

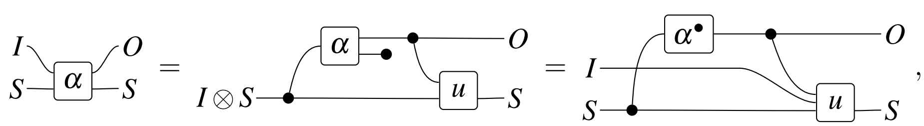


(7)

A morphism of comb machines $(S,\alpha)\to(T,\beta)$ is a morphism $f\colon S\to T$ in $\mathcal{C}$ that commutes with α and β in the same way as for a Mealy machine. The category of comb machines will be written $\mathbf{CombMachine}(I,O)$.

Comb machines take their name from comb elements, as defined in a probability context in [14]. We avoid calling them Moore machines in order to avoid confusion with the more usual definition.

Equation (7) expresses the idea that the output of a comb machine cannot directly depend on the input. Consequently a comb machine α could be seen as a Mealy machine that obeys an extra condition, namely the existence of $\alpha^\bullet$ such that eq. (7) holds. However, we will often think of them differently. If $\mathcal{C}$ has conditionals then a comb machine α can always be factored as

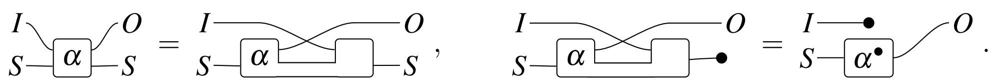


(8)

where u is a conditional of α as shown. We refer to $\alpha^\bullet$ as the *readout map* and u as an *update map* of the comb machine (S,α), analogously to the maps that define a Moore machine. If $f\colon (S,\alpha)\to(T,\beta)$ is a morphism of comb machines, then we have $f\,\mathbin{⨟}\,\alpha^\bullet=\beta^\bullet$, which we show in appendix A.2.

The readout map $\alpha^\bullet$ has the same type as in a Moore machine, $S \to O$, but the update map has type $O \otimes I \otimes S \to S$ and is only defined up to $\alpha^\bullet$-g.a.s. equality. This allows the next state and the output to be correlated for a given previous state and input, while still requiring the output to be independent of the input. Although update maps are not uniquely defined, their behaviour can only differ on measure zero subsets of the output space. In **Dist** this means their behaviour can differ only on outputs $o \in O$ that cannot occur at all in a given state, i.e. for which $\alpha^\bullet(o \mid s) = 0$.

We think of comb machines as giving their output first and then receiving their input, in contrast to Mealy machines, which first receive an input and then give an output.[2] The picture to have in mind for a comb machine is this:



(9)

We now introduce the concept of a *unifilar* machine. A unifilar machine has a stochastic readout map but a deterministic update map. (Or at least, a generalised-almost-surely deterministic one.) The term "unifilar" comes from the literature on computational mechanics, where it can be used to define ϵ-machines [22]. In particular it appears in a machine-like context in [2], proposition 5. The formal context is different, in part because we don't assume stationarity or irreducibility, but our definition achieves the same idea. We define unifilar machines in Mealy machine and comb machine flavours:

Definition 2.3 (unifilar Mealy machine)**.** A *unifilar Mealy machine* in a Markov category is a Mealy machine (S, α) with the condition that α is deterministic given O. Additionally, we require morphisms of unifilar mealy machines to be deterministic. The category of unifilar Mealy machines will be written as $\mathbf{UnifilarMealy}(I, O)$.

Definition 2.4 (unifilar comb machine)**.** A *unifilar comb machine* in a Markov category is a comb machine $(S, \alpha, \alpha^\bullet)$ with the condition that α is deterministic given O. As with unifilar Mealy machines, we require morphisms of unifilar comb machines to be deterministic. The category of unifilar comb machines will be written as $\mathbf{UnifilarComb}(I, O)$.

Note that α must admit a conditional $O \otimes I \otimes S \to S$ in order to satisfy either definition.

When we say "unifilar machine" without qualification we mean a unifilar comb machine.

The idea of a unifilar machine (of either type) is that all of the randomness comes from the choice of output. A unifilar comb machine factors according to eq. (8), with the additional feature that the conditional u is $\alpha^\bullet$-generalised-almost-surely deterministic. We interpret this as follows: first the output O is chosen stochastically (via $\alpha^\bullet \colon S \to O$), and then the state updates $\alpha^\bullet$-g.a.s. deterministically as a function of the output and the input. As for comb machines in general, the behaviour of an update map is uniquely specified on all but $\alpha^\bullet$-measure-zero subsets of the output space.

If $\mathcal{C}$ is Cartesian then Mealy machines and unifilar Mealy machines coincide, as do comb machines and unifilar comb machines, both of which coincide with Moore machines. So both comb machines and unifilar comb machines can claim to be a generalisation of Moore machines to the stochastic case.

It is worth saying something about the meaning of morphisms in these categories. The following can be made formal using the machinery we introduce in section 3, but for now we state it informally. We can think of a non-unifilar machine (of either flavour) as inducing a stochastic map from infinite sequences of inputs to infinite sequences of outputs, subject to a *causality condition* that each output can

[2] This raises the question of whether we can interpose some other morphism in between $\alpha^\bullet$ and u, so that the machine receives an input that can depend on its output, and perhaps also on the outputs of other machines. Answering this in the most general case is rather involved and we will not address it in this paper. However, in the case where $\mathcal{C}$ is **FinStoch**, [14] provides a way to compose 2-combs, of which comb machines are a special case.

only depend on inputs that were received at earlier points in time. (Recall that for Mealy machines we consider the input to be received before the output, and vice versa for comb machines.) We refer to this map as the machine's behaviour. For Mealy machines and comb machines, a morphism $(S,\alpha) \to (T,\beta)$ witnesses that β is capable of exhibiting all of the externally observable behaviours that α can exhibit. Using a stochastic map makes sense because the states are unobserved and change randomly; we consider distributions over states to exhibit behaviours, as well as states themselves.

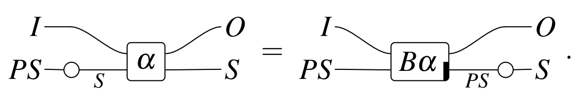
The interpretation of morphisms between unifilar machines is similar, but we require the morphisms to be deterministic. A morphism of unifilar machines witnesses not only that their externally observable behaviour is the same, but also that there is a mapping between their internal states that preserves this behaviour. This makes sense conceptually because we will generally consider the state of a unifilar machine to be observable.

Our first result concerns the existence of an adjunction between the categories $\mathbf{CombMachine}(I,O)$ and $\mathbf{UnifilarComb}(I,O)$, from which Bayesian filtering arises. A similar result holds for $\mathbf{Mealy}(I,O)$ and $\mathbf{UnifilarMealy}(I,O)$, which we will state at the end. Its proof is largely the same.

We first note that there is a forgetful functor $F\colon \mathbf{UnifilarComb}(I,O) \to \mathbf{CombMachine}(I,O)$ that embeds unifilar comb machines into comb machines. On objects it forgets that the machine obeys the deterministic-given-O condition, and it also forgets that morphisms are deterministic.

If $\mathcal{C}$ is strongly representable we can construct a functor in the opposite direction. We first define it and then prove that it lands in $\mathbf{UnifilarComb}(I,O)$ and is a functor.

Definition 2.5. Suppose that $\mathcal{C}$ is a strongly representable Markov category. Then we define a putative functor $B\colon \mathbf{CombMachine}(I,O) \to \mathbf{UnifilarComb}(I,O)$. On objects it maps $(S,\alpha) \mapsto (PS,B\alpha)$, where $B\alpha = (\mathrm{id}_I \otimes \mathrm{samp}_S \mathbin{⨾} \alpha)^{\diamondsuit}$ is the unique morphism such that $B\alpha$ is deterministic given O and

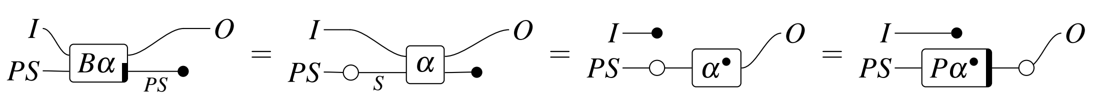


(10)

On morphisms, B maps a morphism of comb machines with underlying map $f\colon S \to T$ to a morphism of unifilar machines with underlying deterministic map $Pf\colon PS \to PT$.

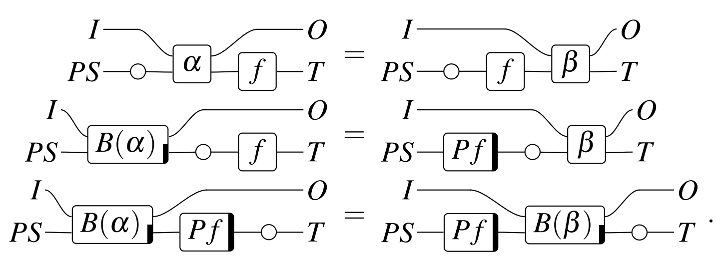
Proposition 2.6. *Let $\mathcal{C}$ be a strongly representable Markov category. Then definition 2.5 yields a functor* $B\colon \mathbf{CombMachine}(I,O) \to \mathbf{UnifilarComb}(I,O)$.

Proof. The mapping respects composition and identities by functoriality of P, but to prove B is a functor we have to show (i) that $B\alpha$ is indeed a unifilar comb machine, and (ii) that that Pf is indeed a morphism of unifilar comb machines. For (i) we show that if eq. (7) holds for α then it holds for $B\alpha$:



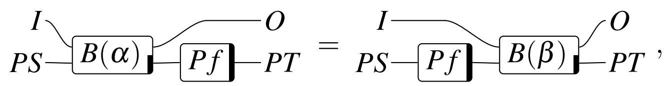
(11)

For (ii), since f is a morphism of comb machines we have



(12)

We can then use the defining property of a strongly representable Markov category in the form of eq. (6) to cancel off the sampling maps and conclude

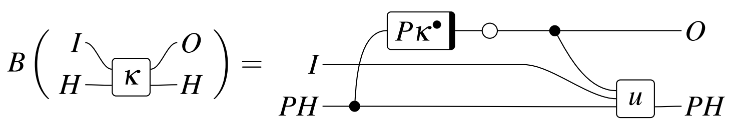


(13)

i.e. Pf is a morphism of unifilar machines. $\square$

We think of the functor B as taking a dynamical model (in the form of a comb machine) and converting it into an epistemic model in the form of a unifilar machine. To unpack this, first consider a comb machine (H,κ) as a dynamical model: we think of H as a set of hidden states and κ as a dynamical process that emits outputs and stochastically changes the hidden state as a function of the input.

Then $B((H,\kappa))$ is a unifilar machine. In order to view it as an epistemic model, we write it using eq. (11) as



(14)

in which the conditional u is $(P\kappa^\bullet \mathbin{⨾} \mathrm{samp})$-g.a.s. deterministic as well as $(P\kappa^\bullet \mathbin{⨾} \mathrm{samp})$-g.a.s. unique. We will think of the state space PH as the space of "beliefs about H," and the update map u as updating those beliefs using Bayesian filtering.

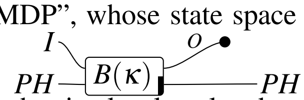
We imagine these beliefs to be held by an idealised Bayesian reasoner, whose prior at any given time is an element of the state space, PH. This Bayesian reasoner does not interact with the machine κ, it only observes the inputs that κ receives and the outputs it emits in response, updating its prior to a posterior at each time step.

The output map $PH \xrightarrow{\mathrm{samp}_H} H \xrightarrow{\kappa^\bullet} O = PH \xrightarrow{P\kappa^\bullet} PO \xrightarrow{\mathrm{samp}_O} O$ "simulates" the output of κ. The map $P\kappa^\bullet$ maps the reasoner's prior beliefs about H to its beliefs about the next output it will observe.

The update map u performs Bayesian filtering. It takes as input a probability measure over the hidden states along with an input and an output, and it returns a new probability measure over hidden states. We think of it as taking a prior over the *current* value hidden state and returning a posterior distribution over the *next* value of the hidden state, conditioned on the observed output. It thus combines Bayesian updating with "simulating" the stochastic change in H.

The output from u is the posterior distribution. It is only defined up to almost sure equivalence. In the case where O is finite this is because for a given output $o \in O$ and a given belief $b \in PH$ we might have $(b \mathbin{⨾} P\kappa^\bullet)(o) = 0$, i.e. the output o is "subjectively impossible" according to the agent's current epistemic state. In this case calculating the Bayesian posterior in the usual way would lead to a division by zero, so there is no consistent value that the posterior distribution could take. Since the update map u is only defined up to $(P\kappa^\bullet \mathbin{⨾} \mathrm{samp})$-g.a.s. equality its output only matters in those cases where this doesn't happen.

As one would expect from a Bayesian filter, instances of the map u can be chained together in such a way that, given an initial distribution over H and a sequence of inputs, we can recover the posterior over H for a given observed sequence of outputs. We give a precise statement of this in appendix A.3, though we omit the proof for reasons of space.

We can thus regard the functor B as taking a dynamical model as input and turning it into an epistemic model. We remark that a similar operation is performed in the process of solving a partially observable Markov decision process (POMDP). A POMDP consists of some kind of machine — for simplicity let us say a comb machine (H,κ) — together with a reward function. This machine is a dynamical model of some environment, and the goal is to find a "policy" that maximises the expected amount of reward that is accumulated over time, usually with an exponential discounting factor. (We will not consider reward functions in the present work.) A common solution technique involves converting the POMDP into a Markov decision process (MDP), which is a simpler class of problem. In an MDP the state space is assumed to be fully observed, so that there is no need to consider outputs. In an MDP the machine only takes inputs, and changes state stochastically as a function of its input, so it can be seen as an object of **CombMachine**$(I,1)$. Again there is an associated reward function, which we will not consider in detail. To turn a POMDP into an MDP one forms the so-called "belief MDP", whose state space is given by probability distributions over H. In our framework it is given by 

. Note that this is a stochastic map in general. For an approach to POMDPs that is closely related to the present work, see [3].

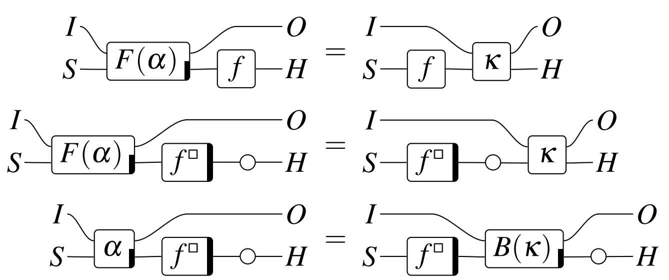
The following is our main technical result.

Theorem 2.7. *When $\mathcal{C}$ is a strongly representable Markov category, the functor B is right adjoint to the forgetful functor F,*

$$\mathbf{CombMachine}(I,O) \underset{B}{\overset{F}{\underset{\longrightarrow}{\overset{\longleftarrow}{\perp}}}} \mathbf{UnifilarComb}(I,O).$$

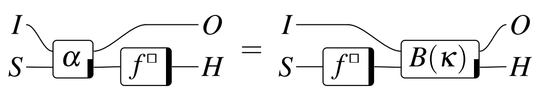
Proof. We show that if $f\colon S \to H$ is the map in $\mathcal{C}$ underlying a morphism $F((S,\alpha)) \to (H,\kappa)$ in **CombMachine**(I,O) then $f^{\square}\colon S \to PH$ is the deterministic map underlying a morphism $(S,\alpha) \to B((H,\kappa))$ in **UnifilarComb**(I,O), and vice versa. This will form the natural isomorphism of hom-sets needed for an adjunction.

Suppose $f\colon S \to H$ is the map underlying a morphism $F((S,\alpha)) \to (H,\kappa)$ in **CombMachine**(I,O). Then we have the following (where, as always, all diagrams are in $\mathcal{C}$):



(15)

We can then use representability of $\mathcal{C}$ in the form of eq. (6) again to conclude that



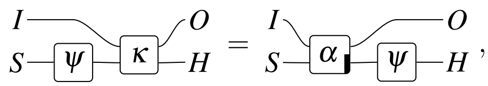
(16)

so that $f^{\square}$ underlies a morphism $(S,\alpha) \to B((H,\kappa))$ in **UnifilarComb**(I,O). Each of these steps can be reversed, so this gives a bijection **CombMachine**$(I,O)(F(-),=) \cong$ **UnifilarComb**$(I,O)(-,B(=))$. Naturality follows from the naturality of the sampling map. $\square$

This adjunction is related to the one between $P\colon \mathcal{C}_{\mathrm{det}} \to \mathcal{C}$ and $\mathcal{C}_{\mathrm{det}} \hookrightarrow \mathcal{C}$ in a representable Markov category, and it shares the same unit and counit. The unit has components $\delta_X\colon X \to PX$ and the counit has components $\mathrm{samp}_X\colon PX \to X$, where $PX = BX$ on objects.

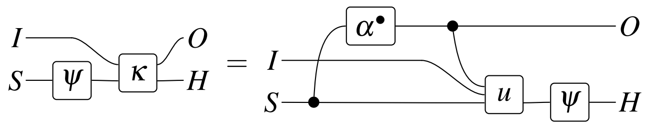
The existence of this adjunction has some interesting consequences. We have already established that the unifilar machine $B((H,\kappa))$ can be seen as an epistemic model of the comb machine (H,κ), seen as a dynamical model. But now consider a morphism $(S,\alpha) \to B((H,\kappa))$ in **UnifilarComb**(I,O) from some other unifilar machine into $B((H,\kappa))$. We argue that when equipped with such a morphism, (S,α) *also* deserves to be seen as modelling (H,κ).

To see this we consider its adjoint map $F((S,\alpha)) \to (H,\kappa)$, which is given by an underlying map $\psi\colon S \to H$ in $\mathcal{C}$ such that



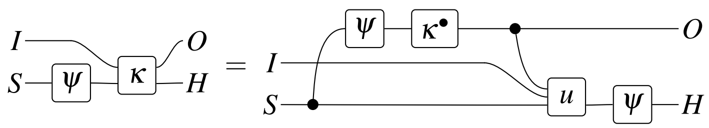
, (17)

or



, (18)

where u is an update map for α. By marginalising both sides (i.e. post-composing with $\mathrm{id}_O \otimes \mathrm{del}_H$) we have $\alpha^\bullet = \psi \mathbin{⨾} \kappa^\bullet$, so this equation becomes



, (19)

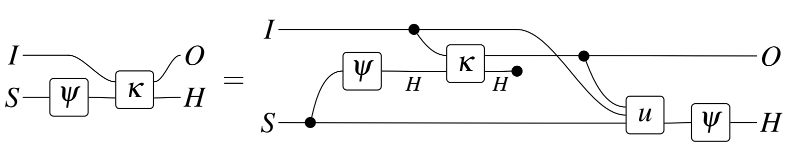
where u is $\psi \mathbin{⨾} \kappa^\bullet$-g.a.s. deterministic. This is a Bayesian filtering version of Jacobs' [13] definition of conjugate priors. It is not quite the same as the one in [23] because in that paper u is not assumed to be almost-surely deterministic, so a stronger equation is needed. However, it is conceptually the same.

The morphism ψ can be regarded as what the author and colleagues called an *interpretation map* in [23]. This means we think of the update map u as a physical machine whose job is to keep track of an epistemic model of κ. At each step it receives both the input that was given to κ and the output that κ emitted in response. The machine's physical state (S) then updates in a ($\psi \mathbin{⨾} \kappa^\bullet$-g.a.s.) deterministic way.

Equation (19) expresses the idea that when the machine receives a new piece of information in the form of an (i,o) pair it should update its beliefs in a consistent way. The left-hand side can be seen as the agent's current beliefs about the next output and the next value of the hidden state, as a function of the next input. The equation says that after receiving an input and output pair, its new beliefs about the current hidden state should equal a conditional of its prior beliefs, conditioned on i and o.

The adjoint map $\psi^\square\colon S \to PH$ can then be seen as mapping the unifilar machine's physical state to a probability measure over H that we think of as "the machine's beliefs about H," i.e. its current Bayesian prior. Since $\psi^\square$ underlies a morphism $(S,\alpha) \to B((H,\kappa))$ it means that α's updates have to be able to 'simulate' the idealised Bayesian filtering that $B((H,\kappa))$ performs. The map $\psi^\square$ can thus be seen as assigning a semantic meaning to the states of the unifilar machine.

We now state the corresponding result for Mealy machines: as for comb machines there are functors $\mathbf{Mealy}(I,O) \underset{F}{\overset{B}{\rightleftarrows}} \mathbf{UnifilarMealy}(I,O)$ such that F is left adjoint to B. The definitions and proofs are the same as for comb machines and unifilar comb machines, except that we don't need to care about the comb condition. These functors can be thought of in the same terms, with B mapping a dynamical model to a corresponding epistemic model. The Mealy machine version of eq. (19) is

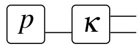


. (20)

An example with enormous practical importance in control theory is the Kalman filter. Although Kalman [16] originally derived it in terms of error minimisation it is well known that it can also be constructed as a Bayesian filter. (See [12] for a somewhat informal exposition, for example.)

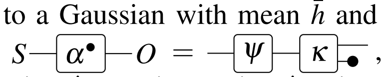
Here we give only the briefest sketch of how the Kalman filter can be formulated in our framework. We consider a version with measurement noise but no input signal. Unlike most treatments we allow the measurement noise and the process noise to be correlated.

Example 2.8 (Kalman filter)**.** Let $\mathcal{C} = \mathbf{BorelStoch}$ and let $I = 1, H = \mathbb{R}^n, O = \mathbb{R}^m$. Consider a comb machine (H,κ) where $\kappa(- \mid h)$ is normally distributed according to $\kappa(- \mid h) \sim \mathcal{N}(Ah,\Sigma)$, where A is an $(m+n)\times n$ matrix and the $(m+n)\times(m+n)$ covariance matrix Σ doesn't depend on h. Taking (H,κ) as a dynamical model of a process, we want to construct a unfilar comb machine that will act as an epistemic model.

To do this, we first note that if $p\colon 1 \to H$ is a normal distribution with mean $\bar{h}$ and covariance matrix Σ_p, then  is also Gaussian, with mean $\bar{s}$ and covariance $\Sigma' := A\Sigma_p A^T + \Sigma$. (See section 6 of [8], for example.) Writing Σ' in block form as $\Sigma' = \left(\begin{smallmatrix}\Sigma'_{OO} & \Sigma'_{OH}\\ \Sigma'_{HO} & \Sigma'_{HH}\end{smallmatrix}\right)$, this distribution $p \mathbin{⨟} \kappa$ has a conditional $c\colon O \to H$ given by

$$c(- \mid o) \sim \mathcal{N}(\Sigma'_{HO}\Sigma'^{-}_{OO}o\,,\, \Sigma'_{HH} - \Sigma'_{HO}\Sigma'^{-}_{OO}\Sigma'_{OH}), \tag{21}$$

where Σ'^{-}_{OO} is the Moore-Penrose pseudoinverse of Σ'_{OO}. (See example 11.8 of [8].)

Let us therefore define a unifilar machine (S,α) where S is the set of pairs $(\bar{h},\Sigma_p)$, where $\bar{h} \in \mathbb{R}^n$ and Σ_p is an $n \times n$ positive definite matrix. To define $\alpha\colon S \to O \times S$ we first define the map $\psi\colon S \to H$, which maps $(\bar{h},\Sigma_p)$ to a Gaussian with mean $\bar{h}$ and covariance Σ_p. We can define $\alpha\colon S \to O \times S$ by the readout function , and a deterministic update map $u\colon O \times S \to S$ that maps $((\bar{h},\Sigma_p),o)$ to $(\Sigma'_{HO}\Sigma'^{-}_{OO}o\,,\, \Sigma'_{HH} - \Sigma'_{HO}\Sigma'^{-}_{OO}\Sigma'_{OH})$, as in eq. (21).

By construction, the maps u, ψ, $\alpha^\bullet$ and κ obey eq. (19), and we can conclude that $\psi^\square$ is a map of unfilar machines from (S,α) to $B((H,\kappa))$.

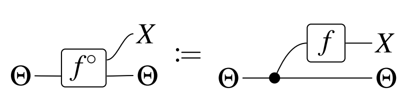
The update map u is a version of the Kalman filter. Its state space H parametrises Gaussian distributions via the map $\psi^\square$. The machine (H,κ) is such that for a Gaussian prior the posterior will also be a Gaussian, and therefore the deterministic map u only has to update the parameters upon receiving new data. We note a similarity between Kalman filtering and the category **Gauss** defined in [8], which we referred to in deriving it.

2.1 Bayesian Inference and Conjugate Priors

Up to now we have considered a version of Bayesian filtering in which the systems being modelled have the form of a comb machine. In this section we consider an important special case of this, in which the system being modelled simply emits independent and identically distributed outputs. This corresponds to the standard setting of Bayesian inference, where we receive independent samples from a known distribution with an unknown (but fixed) value for its parameters, and wish to use this data to make inferences about the parameters.

In this section we primarily consider machines whose input space is the terminal object in $\mathcal{C}$. In this case the distinction between comb machines and Mealy machines isn't relevant, and we refer to such machines as generators, defining $\mathbf{Generator}(X) = \mathbf{Mealy}(1,X) \cong \mathbf{UnifilarMachine}(1,X)$ and $\mathbf{UnifilarGenerator}(X) = \mathbf{UnifilarMealy}(1,X) \cong \mathbf{UnifilarComb}(1,X)$.

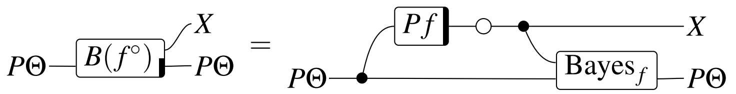
To model Bayesian inference in our setup we consider objects of $\mathbf{Generator}(X)$ represented by morphisms in $\mathcal{C}$ of the following special form:



(22)

In this setting we call X the *sample space* and Θ the *parameter space*, and we think of f as a statistical model, that is, a family of distributions over X parametrised by Θ.

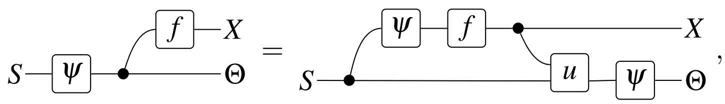
Applying the functor B we get



(23)

where we have called the conditional Bayes_f because that is what it does: it takes in a prior over the parameters together with some data $x \in X$, and returns the Bayesian posterior over the parameters, according to the model f. We give a precise statement and proof for this claim in appendix A.4.

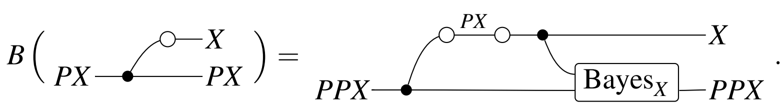
If we consider a map $\psi^\square$ into this machine from some other unifilar machine (S,α), we obtain exactly the notion of a conjugate prior. Its adjoint map of comb machines, $\psi\colon F((S,\alpha)) \to f^\circ$, obeys



(24)

which is the equation given in [13] as a definition of conjugate prior. The only minor difference is that here the update map is only defined up to $(\psi\,\mathbin{;}\,f)$-g.a.s. equality, instead of being a specified deterministic function. We think of $\psi\colon S \to \Theta$ as a statistical model and say that it is a conjugate prior for f. Its parameter space S is referred to as the space of hyperparameters. Obtaining this more abstract perspective on the definition from [13] was one of the main motivations of this work.

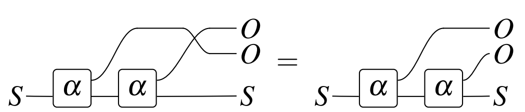
It is worth briefly mentioning the further special case in which f is the sampling map, although we will not make use of it.



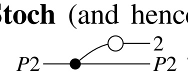
(25)

Here Bayes_X also performs Bayesian updating, corresponding to inference about an unknown distribution. It takes a distribution over distributions over X, representing a prior, along with a sample from the unknown distribution. Its output is the Bayesian posterior over distributions, conditioned on the sample.

We note that all the generators in this section obey the property of *exchangeability*, specifically the version of that concept defined in [15] in the context of de Finetti's theorem. That is, they are all machines (S,α) such that



(26)

In our context, one of the results of [15] is that in **Stoch** (and hence also in **BorelStoch**) the category **Generator**($\{0,1\}$) has a terminal object, given by  , which is part of their category-theoretic treatment of de Finetti's theorem. (A much more general version of de Finetti's theorem is proved for **BorelStoch** in [9], though in a less machine-like context.)

In the context of the machines in eq. (23) and eq. (25), exchangeability amounts to the idea that a Bayesian reasoner should reach the same posterior from the same data, regardless of the order in which the data are presented. (Except that here this is subject to the usual generalised-almost-surely condition.)

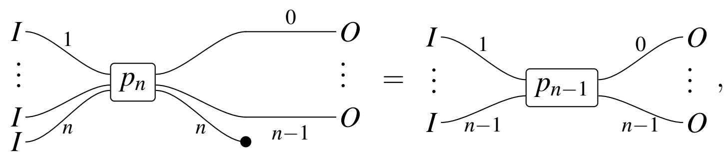
There is much more that can be said about exchangeability and its relationship to Bayesian inference within the framework of unifilar machines, but we will leave the topic here and return to the more general case of non-exchangeable machines in the next section.

3 Terminal objects as "objects of behaviours"

If **UnifilarComb**(I,O) has a terminal object then it can be seen as an "object of behaviours," in much the same manner as a final coalgebra. If such a terminal object exists we call it an *object of transducers*. The intuition is that if we can meaningfully talk about its elements then they can be thought of as stochastic maps from infinite sequences of inputs to infinite sequences of outputs, subject to the causality condition described above, that each output can only depend on inputs that were received prior to it.

To illustrate this idea we prove that transducer objects always exist in **Dist**, and their elements indeed have the form of stochastic maps between sequences. For this we will need the definition of a controlled stochastic process. This is a classical idea, but the category theoretic definition we give is similar to definition 9.12 of [6]. For further generalisations with a slightly different flavour, see section 7 of [8].

Definition 3.1 (controlled stochastic process)**.** In a Markov category $\mathcal{C}$, we define an *output-first controlled stochastic process* with input space I and output space O as a family of morphisms $p_n\colon I^{n-1} \to O^n$ for $n \geq 1$, subject to the condition that



(27)

where the labels on the wires represent the indexes of the inputs and outputs. (Note that the indices for the inputs start from 1 while the indices for the outputs start from 0, so that p_n has $n-1$ inputs and n outputs. We use this convention because we consider the first output to occur "at time 0," before the first input.) An *input-first controlled stochastic process* is defined similarly, but with the outputs indexed starting from 1 instead of 0, so that p_n has type $I^{n-1} \to O^{n-1}$.

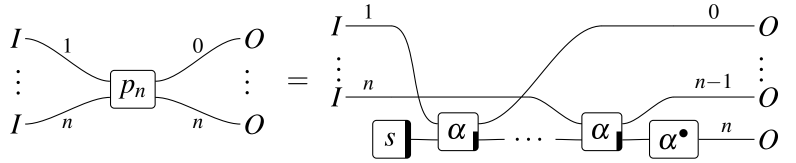
When we say "controlled stochastic process" without qualification, we mean an output-first controlled stochastic process. The condition says both that the family of distributions has to be consistent with each other, and that each output can only depend on inputs that were received prior to it.

Proposition 3.2 (**Dist** has transducer objects)**.** *In* **Dist***, the terminal object* (ω, T) *of* **UnifilarComb**(I,O) *exists and is as follows. T is the set of all output-first controlled stochastic processes (in* **Dist***). ω is composed of the following readout and update maps: the readout map sends a controlled stochastic process p to the distribution p_1, which is a distribution over O with no input. Given $i \in I$, $o \in O$ and a controlled stochastic process p, the update map sends (i,o,p) to a delta distribution concentrated on a new controlled stochastic process $p^{i,o}$ given by*

$$p^{i,o}_n(o_0,\ldots,o_n \mid i_1,\ldots,i_n) = \frac{1}{p_1(o)} p_{n+1}(o,o_0,\ldots,o_n \mid i,i_1,\ldots,i_n) \tag{28}$$

if $p_1(o) > 0$, and to some arbitrary distribution over controlled stochastic processes otherwise. (As such it is defined up to the appropriate generalised-almost-surely condition.)

Proof. Given a unifilar machine (S,α) and a state $s \in S$, one can show inductively that under any morphism of unifilar machines $(S,\alpha) \to (T,\omega)$, the state s must map to the controlled stochastic process given by



(29)

We give the details in appendix A.5 □

The update map of (T,ω) performs Bayesian conditioning: it returns a new map from input sequences to output sequences, formed by fixing the first input and conditioning on the first output.

As usual a similar result holds for Mealy machines: **UnifilarMealy**(I,O) has a terminal object in **Dist**, whose objects can be seen as input-first controlled stochastic processes. The proof is similar and doesn't involve the comb condition.

One advantage of formulating transducers internally in this way is that we can consider probability distributions over them. In particular, since (T,ω) is a terminal object it is equipped with an algebra of the monad $F \mathbin{⨟} B$ arising from the adjunction in theorem 2.7. This means that we can think of the unique map $B(F((T,\omega))) \to (T,\omega)$ as taking a probability distribution over transducers and returning a new transducer that represents its 'average' or 'expected' behaviour. This will work in any suitable Markov category, whenever a terminal object of **UnifilarComb**(I,O) exists.

On the other hand, in **BorelStoch** the category **UnifilarMealy**$(\mathbb{R},\{0,1\})$ does not have a terminal object. Consider those machines with trivial state spaces, whose output depends only on the current input. Specifying the behaviour of such a machine amounts to specifying a measurable map $\mathbb{R} \to [0,1]$. But there is no measurable space of such functions, so there is no measurable space that includes the behaviours of all such machines. However, we conjecture that **BorelStoch** has terminal objects for **UnifilarComb**(I,O) and **UnifilarMealy**(I,O) when I is a countable or finite set.

Acknowledgements

The author thanks Martin Biehl, Matteo Capucci and the anonymous reviewers for insightful comments on the manuscript, and Martin Biehl, Simon McGregor, Timorl, Matteo Capucci and Toby Smithe for discussions that stimulated the work. This paper was made possible through the support of Grant 62229 from the John Templeton Foundation. The opinions expressed in this publication are those of the author(s) and do not necessarily reflect the views of the John Templeton Foundation.

References

[1] Jiří Adámek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius & Alexandra Silva (2012): *A coalgebraic perspective on minimization and determinization*. In: *Foundations of Software Science and Computational Structures: 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24–April 1, 2012. Proceedings 15*, Springer, pp. 58–73, doi:10.1007/978-3-642-28729-9_4.

[2] Nix Barnett & James P. Crutchfield (2015): *Computational Mechanics of Input-Output Processes: Structured Transformations and the ϵ-Transducer*. *Journal of Statistical Physics* 161(2), pp. 404–451, doi:10.1007/s10955-015-1327-5.

[3] Martin Biehl & Nathaniel Virgo (2023): *Interpreting Systems as Solving POMDPs: A Step Towards a Formal Understanding of Agency*. In: *Active Inference. IWAI 2022. Communications in Computer and Information Science*, Springer, pp. 16–31, doi:10.1007/978-3-031-28719-0_2.

[4] Dylan Braithwaite, Jules Hedges & Toby St Clere Smithe (2023): *The Compositional Structure of Bayesian Inference*, doi:10.48550/ARXIV.2305.06112. Available at https://arxiv.org/abs/2305.06112.

[5] Kenta Cho & Bart Jacobs (2017): *Disintegration and Bayesian Inversion via String Diagrams*, doi:10.48550/ARXIV.1709.00322. Available at https://arxiv.org/abs/1709.00322.

[6] Elena Di Lavore, Giovanni de Felice & Mario Román (2022): *Coinductive Streams in Monoidal Categories*, doi:10.48550/ARXIV.2212.14494. Available at https://arxiv.org/abs/2212.14494.

[7] T Fritz, A Klinger, D. McNeely, A. Shah-Mohammed & Y. Wang (2023): *Hidden Markov Models and the Bayes Filter in Categoryical Probability*. Available at https://act2023.github.io/papers/paper73.pdf. Abstract presented at Applied Category Theory 2023.

[8] 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.

[9] Tobias Fritz, Tomáš Gonda & Paolo Perrone (2021): *De Finetti's Theorem in Categorical Probability*. Journal of Stochastic Analysis 2(4), doi:10.31390/josa.2.4.06.

[10] Tobias Fritz, Tomáš Gonda, Paolo Perrone & Eigil Fjeldgren Rischel (2020): *Representable Markov Categories and Comparison of Statistical Experiments in Categorical Probability (v2)*, doi:10.48550/ARXIV.2010.07416. Available at https://arxiv.org/abs/2010.07416v2. The material on strongly representable Markov categories appears in version 2 of the preprint but not in version 3.

[11] Tobias Fritz, Tomáš Gonda, Paolo Perrone & Eigil Fjeldgren Rischel (2023): *Representable Markov categories and comparison of statistical experiments in categorical probability*. Theoretical Computer Science 961, p. 113896, doi:10.1016/j.tcs.2023.113896.

[12] Ramakrishna Gurajala, Praveen B. Choppala, James Stephen Meka & Paul D. Teal (2021): *Derivation of the Kalman filter in a Bayesian filtering perspective*. In: 2nd International Conference on Range Technology (ICORT), pp. 1–5, doi:10.1109/ICORT52730.2021.9581918.

[13] B. Jacobs (2020): *A channel-based perspective on conjugate priors*. Mathematical Structures in Computer Science 30(1), pp. 44–61, doi:10.1017/s0960129519000082.

[14] Bart Jacobs, Aleks Kissinger & Fabio Zanasi (2019): *Causal Inference by String Diagram Surgery*. In Mikołaj Bojańczyk & Alex Simpson, editors: Foundations of Software Science and Computation Structures, Springer International Publishing, Cham, pp. 313–329, doi:10.1007/978-3-030-17127-8_18.

[15] Bart Jacobs & Sam Staton (2020): *De Finetti's Construction as a Categorical Limit*. In D. Petrişan & J. Rot, editors: Coalgebraic Methods in Computer Science, Springer International Publishing, pp. 90–111, doi:10.1007/978-3-030-57201-3_6.

[16] Rudolph Emil Kalman (1960): *A New Approach to Linear Filtering and Prediction Problems*. Transactions of the ASME–Journal of Basic Engineering 82(Series D), pp. 35–45, doi:10.1115/1.3662552.

[17] Kotaro Kamiya & John Welliaveetil (2021): *A category theory framework for Bayesian learning*, doi:10.48550/ARXIV.2111.14293. Available at https://arxiv.org/abs/2111.14293.

[18] David Jaz Myers (2022): *Categorical Systems Theory*. Unpublished book draft. Available at http://davidjaz.com/Papers/DynamicalBook.pdf.

[19] Alexandra Silva, Filippo Bonchi, Marcello Bonsangue & Jan Rutten (2013): *Generalizing determinization from automata to coalgebras*. Logical Methods in Computer Science Volume 9, Issue 1, doi:10.2168/lmcs-9(1:9)2013.

[20] Toby St. Clere Smithe (2020): *Bayesian Updates Compose Optically*, doi:10.48550/ARXIV.2006.01631. Available at https://arxiv.org/abs/2006.01631.

[21] David I. Spivak (2019): *Generalized Lens Categories via functors* $\mathscr{C}^{\mathrm{op}} \to \mathsf{Cat}$, doi:10.48550/ARXIV.1908.02202. Available at https://arxiv.org/abs/1908.02202.

[22] Nicholas F. Travers & James P. Crutchfield (2011): *Equivalence of History and Generator Epsilon-Machines*, doi:10.48550/ARXIV.1111.4500. Available at https://arxiv.org/abs/1111.4500.

[23] Nathaniel Virgo, Martin Biehl & Simon McGregor (2021): *Interpreting Dynamical Systems as Bayesian Reasoners*. In: International Workshops of ECML PKDD 2021, Springer, pp. 726–762, doi:10.1007/978-3-030-93736-2_52.

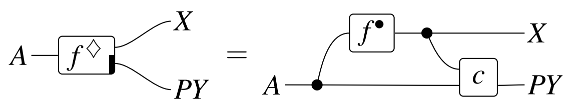
A Proof Details

A.1 Dist is strongly representable

For a morphism $f\colon A \to B$ in **Dist** we write $f(b \mid a)$ for the probability assigned to b by the morphism f when given a as an input. We have that for a given a there only finitely many elements of b for which $f(b \mid a) > 0$, and we have $\sum_{b\in B} f(b \mid a) = 1$.

For a set A the distribution object PA is the set of all finitely supported probability distributions over A. In other words it is the set of functions $A \to [0,1]$ that satisfy the properties above, i.e. functions that have a finite support and sum to 1. The sampling map $\mathrm{samp}_A\colon PA \to A$ is given by $\mathrm{samp}_A(a \mid p) = p(a)$.

We now consider an arbitrary morphism $f\colon A \to X \otimes Y$ and a morphism $f^{\diamondsuit}\colon A \to X \otimes PY$ such that $f^{\diamondsuit}$ is deterministic given X and such that eq. (5) holds. We can factor $f^{\diamondsuit}(x, p \mid a)$ as



, (30)

or $f^{\bullet}(x \mid a)c(p \mid x,a)$, for some conditional c.

For $f^{\diamondsuit}$ to be deterministic given X means that the conditional c must be $f^{\bullet}$-generalised-almost-surely deterministic (eq. (4)), which in **Dist** means that $c(- \mid x,a)$ is a delta distribution whenever $f^{\bullet}(x \mid a) > 0$. For this to be true we must have that that whenever a and x are such that $f^{\bullet}(x \mid a) > 0$ there exists a distribution $p_{x,y} \in PY$ such that

$$f^{\diamondsuit}(x, p \mid a) = \begin{cases} f^{\bullet}(x \mid a) & \text{if } p = p_{a,x} \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

In **Dist**, eq. (5) (the definition of $f^{\diamondsuit}$) amounts to

$$\begin{aligned} f(x,y \mid a) &= \sum_{p\in PY} f^{\diamondsuit}(x,p \mid a)\,\mathrm{samp}_Y(y \mid p) \\ &= \sum_{p\in PY} f^{\diamondsuit}(x,p \mid a)p(y) \\ &= f^{\bullet}(x \mid a)p_{x,a}(y). \end{aligned} \tag{32}$$

To show that **Dist** is strongly representable we have to show that $p_{x,a}$ is uniquely defined whenever $f^{\bullet}(x \mid a) > 0$. But this follows immediately because we have, from eq. (32),

$$p_{x,a}(y) = \frac{f(x,y \mid a)}{f^{\bullet}(x \mid a)}, \tag{33}$$

which completes the proof.

Explicitly, the only choices for $f^{\diamondsuit}$ are those of the form

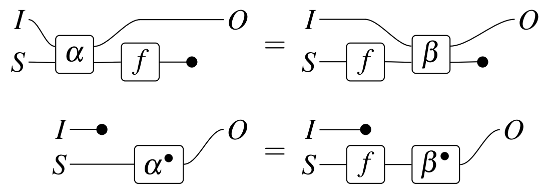
$$f^{\diamondsuit}(x,p \mid a) = \begin{cases} \left\{ \begin{array}{ll} f^{\bullet}(x \mid a) & \text{if } p = \frac{f(x,-|a)}{f^{\bullet}(x|a)} \\ 0 & \text{otherwise} \end{array} \right\} & \text{if } f^{\bullet}(x \mid a) > 0, \\ \\ \text{arbitrary} & \text{otherwise.} \end{cases} \tag{34}$$

The arbitrary values are subject to the constraint that $\sum_{x,p} f(x,p \mid a) = 1$, as always. They occur only outside the support of $f^{\bullet}(- \mid a)$, which in **Dist** means that all possible choices for $f^{\diamondsuit}$ are $f^{\bullet}$-g.a.s. equal, as required.

We note that the last step in the proof, eq. (33), would not be valid in example 3.23 of [10], in which the probabilities are not real-valued.

A.2 Comb machines morphisms commute with readout maps

We want to show that if $f\colon (S,\alpha) \to (T,\beta)$ is a morphism of comb machines, then $\alpha^\bullet = f \mathbin{⨾} \beta^\bullet$. By marginalising the definition of a morphism of comb machines and then substituing the definition of $\alpha^\bullet$ and $\beta^\bullet$ we have



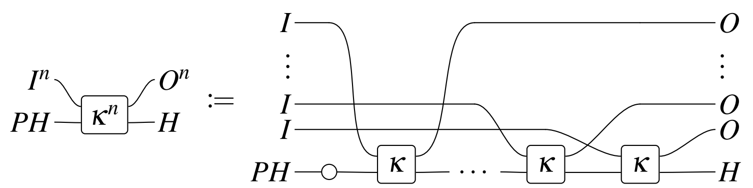
(35)

and the result follows.

A.3 Filtering on sequences

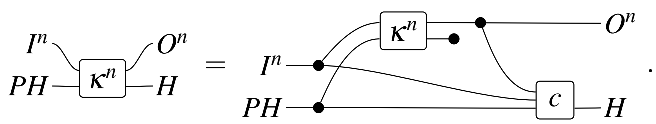
Our claim is that given a dynamical model in the form of a comb machine (H,κ), an initial distribution over H, a sequence of inputs (an element of I^n), and an observed sequence of outputs (an element of O^n), the Bayesian filter (i.e. an update map for $B((H,\kappa))$) allows us to recover the posterior distribution over H, given the observations.

To make this formal let us define



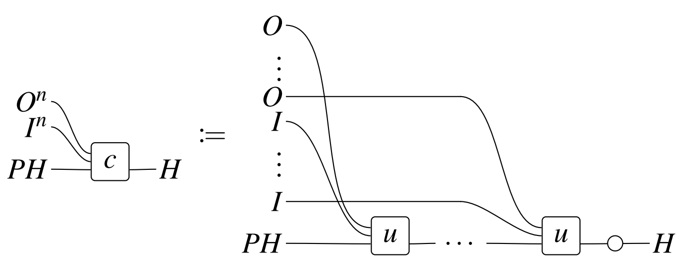
(36)

What we seek is a conditional of κ^n, that is $c\colon O^n \otimes I^n \otimes PH \to H$ such that



(37)

Then c, or rather $c^\square : O^n \times I^n \times PH \to PH$, is the map that takes the observed output sequence, the input sequence and returns the prior over H to the posterior over H. Our claim is that such a conditional c is given by composing n instances of u as follows,



(38)

where u is an update map for $B((H,\kappa))$.

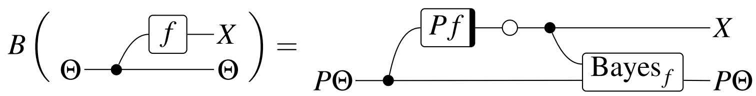
The proof of this is by induction and can be expressed as a lengthy but straightforward string diagram calculation; we omit it for reasons of space. A similar result holds in the Mealy machine case.

A proof of a similar statement was given previously by the author and colleagues in [23] (proposition 2 in appendix B.2), although with somewhat different definitions since that paper does not consider representable Markov categories.

It is worth remarking that the omitted proof uses the fact that u is $(P\kappa^\bullet \mathbin{⨟} \mathrm{samp})$-g.a.s. deterministic, since this fact is not directly used in the proofs of our other results. (The assumption is nevertheless needed, in order to apply the defining property of a strongly representable Markov category.)

A.4 Bayes_f does Bayes

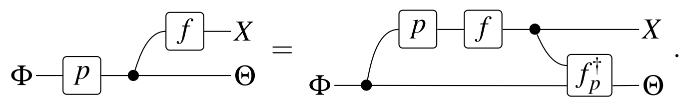
We want to show that the morphism Bayes_f in the expression



(39)

can be seen as performing a Bayesian update.

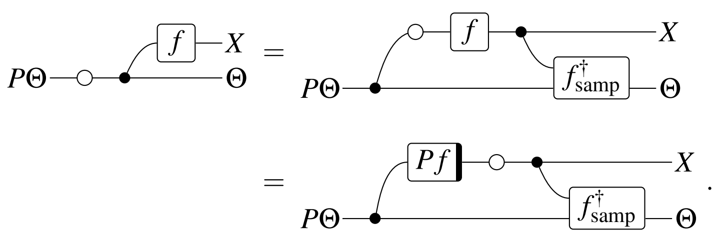
For this we recall the definition of a *Bayesian inverse* from [5], which we generalise slightly by allowing the prior to depend on a parameter. In a Markov category $\mathcal{C}$, given a morphism $f\colon \Theta \to X$ and a morphism $p\colon \Phi \to \Theta$ called the prior, we say that $f_p^\dagger\colon X \to \Theta$ is a *Bayesian inverse of* $f\colon \Theta \to X$ *with respect to* p if



(40)

If $\mathcal{C}$ has conditionals then such Bayesian inverse exists for any f and p. It is not necessarily unique, but like any conditional it is unique up to $(p \mathbin{⨟} f)$-g.a.s. equivalence.

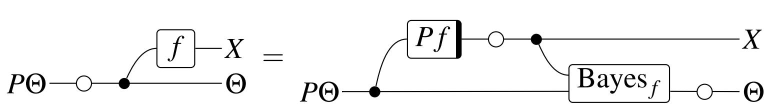
If $\mathcal{C}$ is representable we can take $\Phi = P\Theta$ and $p = \mathrm{samp}_\Theta$, yielding



(41)

The morphism $f^\dagger_{\mathrm{samp}}$ can be thought of as performing a Bayesian inversion of f with respect to *any* prior, since it takes an element of $P\Theta$ as an input.

If $\mathcal{C}$ is strongly representable, then from the definition of Bayes_f we have



(42)

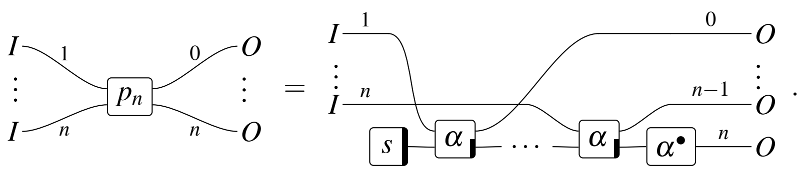
We conclude that up to $Pf \mathbin{⨟} \mathrm{samp}$-g.a.s. equivalence, Bayes_f is the same as $(f^\dagger_{\mathrm{samp}})^\square$, the deterministic version of $f^\dagger_{\mathrm{samp}}$. It takes as input a prior in $P\Theta$ and a data point from X, and gives as output the Bayesian posterior as an element of $P\Theta$.

A.5 Dist has transducers

We write $\omega^\bullet\colon T \to O$ for the readout map of the unifilar machine (T,ω) defined in proposition 3.2. We want to show that this unifilar machine is the terminal object of $\mathbf{UnifilarComb}(I,O)$.

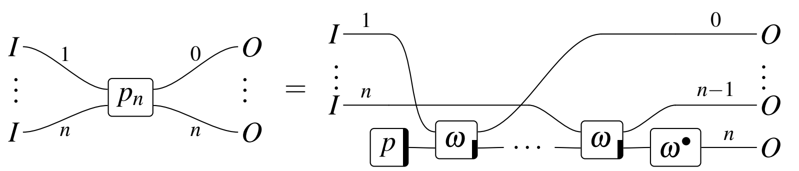
We note that the existence of the terminal object could be proven in a different way, by noting that comb machines in Dist can be formulated as coalgebras of a polynomial functor on **Set**. The existence of a terminal object (a.k.a. final coalgebra) is then a standard result. However, the proof below leads more directly to the probabilistic interpretation in terms of controlled stochastic processes.

As mentioned in the main text we will show that given a unifilar machine (S,α) and a state $s \in S$, under any morphism of unifilar machines $(S,\alpha) \to (T,\omega)$, the state s must map to the controlled stochastic process given by



(43)

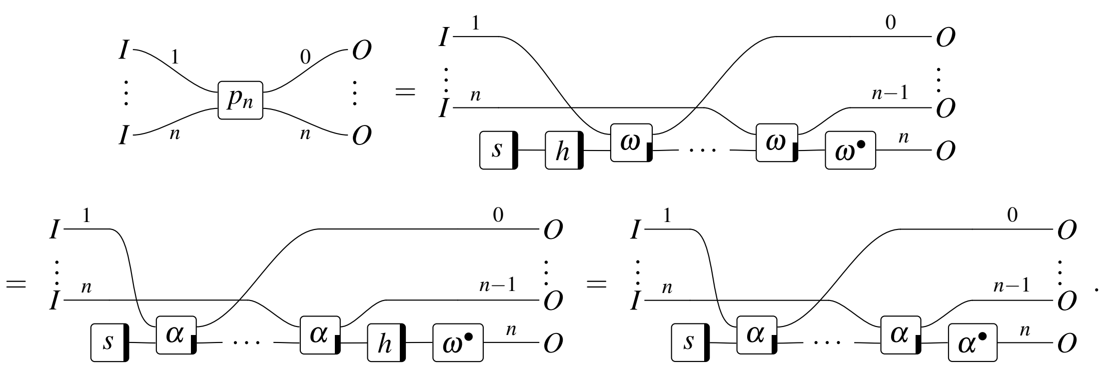
It is straightforward to show inductively that for $p \in T$ we have



(44)

where $\omega^\bullet$ is the readout map of (T,ω).

Now suppose we are given a unifilar machine (S,α), and write $\alpha^\bullet\colon S \to O$ for its readout map. Consider a map of unifilar machines $h\colon (S,\alpha) \to (T,\omega)$. Our goal is to show that such a map always exists and is uniquely defined. Let $s \in S$ be state of α and let $p = h(s)$ be the transducer that it maps to under h. We then calculate



(45)

The second equality is by induction, moving h to the right across the chain of n morphisms.

This leaves us with exactly one choice for $p = h(s)$ for each $s \in S$, and we conclude that the map $h\colon (S,\alpha) \to (T,\omega)$ is unique.