Limits and Colimits in a Category of Lenses

Emma Chollet
ETH Zürich
Zürich, Switzerland
emma.chollet@eawag.ch

Bryce Clarke
Macquarie University
Sydney, Australia
bryce.clarke1@hdr.mq.edu.au

Michael Johnson
Macquarie University
Sydney, Australia
mike@ics.mq.edu.au

Maurine Songa
University of KwaZulu-Natal
Durban, South Africa
maurine@aims.ac.za

Vincent Wang
University of Oxford
Oxford, UK
vincent.wang@cs.ox.ac.uk

Gioele Zardini
ETH Zürich
Zürich, Switzerland
gzardini@ethz.ch

Lenses are an important tool in applied category theory. While individual lenses have been widely used in applications, many of the mathematical properties of the corresponding categories of lenses have remained unknown. In this paper, we study the category of small categories and asymmetric delta lenses, and prove that it has several good exactness properties. These properties include the existence of certain limits and colimits, as well as so-called imported limits, such as imported products and imported pullbacks, which have arisen previously in applications. The category is also shown to be extensive, and it has an image factorisation system.

1 Introduction

Lenses, and their use for synchronising systems, have been an important tool in applied category theory dating back to even before the term "Applied Category Theory" was first used in its modern form. Lenses were introduced by Pierce and Schmitt in 2003 under that name [12], but under other names lenses were an important part of the database view updating work of the 1980s. The full axiomatic description of what are now called *very well behaved set-based lenses* first appeared in a study of storage management in the thesis of Oles [11]. Since that time many different flavours of lenses have been introduced, and a very wide variety of applications have been found.

The first lenses were *asymmetric lenses*, so called to emphasise that when they were used to maintain consistency between two systems, one of the systems had all the information required to reproduce the entire state of the other system (as in a database and its views). However, many real-world synchronisation problems are more symmetric in that each system has state that cannot be derived from the other. From the beginning of the study of such symmetric systems it was recognised that symmetric lenses could be built from asymmetric lenses, so the mathematical study of asymmetric lenses has remained central to the subject.

The set-based asymmetric lenses were soon seen to be a special case of a more general, and more useful, notion called *delta lenses* [5], which might also be described as *category-based lenses*. The original set-based lenses are the special case where the categories in question are codiscrete [7]. These asymmetric category-based lenses were seen to unify a wide range of lenses and their applications, and they are the subject of study in this paper.

Another distinction among lenses worthy of note has sometimes been described as the lawful versus the lawless lenses. It often happens in engineering that systems are designed with axioms or assertions or other rules of well-definedness in mind, but the major engineering job is to build the infrastructure which can support those systems, and that infrastructure may, or may not, enforce the axioms — it is

K. Kishida (Ed.): Fourth International Conference
on Applied Category Theory (ACT 2021).
EPTCS 372, 2022, pp. 164–177, doi:10.4204/EPTCS.372.12

quite common to leave the questions of validity with respect to axioms or assertions to the user. Thus we have the *lawless lenses*, those which have the lens operations, usually called Put and Get, but with few or no requirements about how those operations interact with each other or with data. In fact these lawless lenses have come to be seen as important in a range of applications of their own including economics, game theory and machine learning. Nevertheless, the *lawful lenses*, those that are required to satisfy the basic axioms originally proposed, axioms which are seen here to correspond to various types of functoriality and fibering, remain the principal object of mathematical study, and are the lenses analysed in this paper.

When we say *lens* in this paper we will mean *lawful category-based asymmetric lens*.

The urgency of the applications of lenses has meant over the years that much of the work has focused on individual lenses as needed. Of course it was recognised early that lenses compose, associatively and with identities, and so form a category called $\mathcal{L}\text{ens}$, whose objects are small categories and whose arrows are lenses. But that category has, until this paper, been little studied, and its properties were only hinted at in earlier work. One of those properties caught the attention of early workers, and is an important motivation for this paper.

We have already noted that symmetric lenses can be studied via asymmetric lenses: a *symmetric lens* is an equivalence class of spans of asymmetric lenses. So one might expect that the well-understood theory of spans in a category would apply, and would support the study of the (bi)category of symmetric lenses as $\mathsf{Span}(\mathcal{L}\text{ens})$. That theory depends on using pullbacks to compose spans, so the obvious first step was to construct pullbacks in $\mathcal{L}\text{ens}$. Attempts to do this seemed straightforward: one can calculate the pullback of the lenses' Get functors in $\mathcal{C}\text{at}$, and it is easy to find a canonical construction of Put operations on the resultant projections which satisfy all the required axioms. Thus one has a "pullback" in $\mathcal{L}\text{ens}$, but the quotation marks are there because it soon became apparent that most of the "pullbacks" were not pullbacks in $\mathcal{L}\text{ens}$ at all — they did not satisfy the required universal property with respect to lenses. Nevertheless, and somewhat surprisingly, these "pullbacks" did exhibit many of the properties of pullbacks and in fact did everything required to support the imagined theory of symmetric lenses [8]. In some sense one could "import" pullbacks from $\mathcal{C}\text{at}$ into $\mathcal{L}\text{ens}$ by adding canonical Put operations, and the imported-pullbacks would behave sufficiently like real pullbacks to develop the required theory.

In our view, it is time to seriously study the categorical properties of the category $\mathcal{L}\text{ens}$. This paper begins that study, exploring in $\mathcal{L}\text{ens}$ imported pullbacks and real pullbacks, imported products and real products, equalisers, coproducts, extensivity, and a surprisingly simple proper orthogonal factorisation system. Each of these notions has important practical applications, and understanding the categorical nature of $\mathcal{L}\text{ens}$, including various imported exactness properties, is an important step in advancing applied category theory using lenses.

Acknowledgements

This paper arose from the ACT2020 Adjoint School through research by the Maintainable Relations group. We are grateful to the organisers of the school for their support. We have benefited from valuable conversations with a number of colleagues in the School and in our home and other institutions. We particularly mention Chris Heunen, who asked a number of questions that are now answered by this paper. We also extend our gratitude to the anonymous referees for their helpful feedback on this paper.

Bryce Clarke is grateful for the support of the Australian Government Research Training Program Scholarship. The work of Michael Johnson is supported in part by the Australian Research Council. Gioele Zardini is supported by the Swiss National Science Foundation under NCCR Automation, grant agreement `51NF40_180545`, and he would like to thank Emilio Frazzoli for support.

2 Background

In this section, we recall the category $\mathcal{L}\mathrm{ens}$ of small categories and (delta) lenses [5], and establish notation for the rest of the paper. The only new result presented here is Lemma 2.6(ii).

Definition 2.1. Let A and B be categories. A (*delta*) *lens* $(f, \varphi)\colon A \rightleftharpoons B$ consists of a functor $f\colon A \to B$ together with a *lifting operation*,

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

which satisfies the following axioms:

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

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

(3) $\varphi(a, v \circ u) = \varphi(a', v) \circ \varphi(a,u)$

Remark. In the literature, the functor part of a lens is often called the Get, while the lifting operation is called the Put. The three axioms are also called Put-Get, Get-Put, and Put-Put, respectively. This terminology can be confusing and distracts from the mathematics, so will be avoided in this paper.

Example 2.2. A *split opfibration* is a lens whose chosen lifts $\varphi(a,u)$ are opcartesian.

Definition 2.3. Let $\mathcal{L}\mathrm{ens}$ denote the category whose objects are (small) categories and whose morphisms are lenses. Given a pair of lenses $(f, \varphi)\colon A \rightleftharpoons B$ and $(g, \gamma)\colon B \rightleftharpoons C$, their composite is given by the functor $g \circ f\colon A \to C$ together the lifting operation:

$$(a \in A, u\colon gfa \to c \in C) \quad \longmapsto \quad \varphi(a, \gamma(fa, u))$$

The identity lens on a category A consists of the identity functor $1_A\colon A \to A$ together with the trivial lifting operation given by projection $\pi(a, u\colon a \to a') = u$.

There is an identity-on-objects, forgetful functor $\mathcal{U}\colon \mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}$ which assigns a lens to its underlying functor. The functor $\mathcal{U}$ is neither *full*, as not every functor can be given a lifting operation, nor *faithful*, as a functor may have many possible lifting operations; however it is an *isofibration*. Despite $\mathcal{U}$ failing to be full or faithful, there is a large class of functors for which there does exist a unique lifting operation, called discrete opfibrations, that play a special role in the theory of lenses.

Definition 2.4. A functor $f\colon A \to B$ is a *discrete opfibration* if for all pairs $(a \in A, u\colon fa \to b \in B)$ there exists a unique morphism $w\colon a \to a'$ in A such that $fw = u$. A *cosieve* is an injective-on-objects discrete opfibration (equivalently, fully faithful discrete opfibration).

Discrete opfibrations are equivalent to lenses whose lifting operation is an isomorphism. Let $\mathcal{D}\mathrm{opf}$ denote the wide subcategory of $\mathcal{C}\mathrm{at}$ whose morphisms are discrete opfibrations. Discrete opfibrations are also stable under pullback along arbitrary functors. The following result, due to Clarke [2], establishes the importance of discrete opfibrations for understanding lenses.

Proposition 2.5. *Every lens* $(f, \varphi)\colon A \rightleftharpoons B$ *may be represented as a commutative diagram of functors,*

$$\begin{array}{ccc} & X & \\ {}^{\varphi}\swarrow & & \searrow^{\overline{\varphi}} \\ A & \xrightarrow[f]{} & B \end{array} \tag{1}$$

where φ *is a faithful, bijective-on-objects functor and* $\overline{\varphi}$ *is a discrete opfibration.*

Remark. As noted in [3], this result has a converse which implies that every lens $(f,\varphi)\colon A \rightleftharpoons B$ may be identified with an equivalence class of diagrams,

$$\begin{tikzcd} & X \arrow[dl, "\varphi"'] \arrow[dr, "\overline{\varphi}"] & \\ A \arrow[rr, "f"'] & & B \end{tikzcd} \qquad \simeq \qquad \begin{tikzcd} & Y \arrow[dl, "\gamma"'] \arrow[dr, "\overline{\gamma}"] & \\ A \arrow[rr, "f"'] & & B \end{tikzcd}$$

generated by isomorphisms $q\colon X \cong Y$ such that $\gamma \circ q = \varphi$ and $\overline{\gamma} \circ q = \overline{\varphi}$. In practice, we may always identify a lens with a chosen representative (1) of this equivalence class.

Proposition 2.5 is powerful as it allows us to prove results about lenses through manipulating their representation as diagrams in $\mathcal{C}\mathrm{at}$. For example, composition of lenses may be understood diagrammatically via pullback:

$$\begin{tikzcd} & & X \times_B Y \arrow[dl] \arrow[dr] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & & \\ & X \arrow[dl, "\varphi"'] \arrow[dr, "\overline{\varphi}"] & & Y \arrow[dl, "\gamma"'] \arrow[dr, "\overline{\gamma}"] & \\ A \arrow[rr, "f"'] & & B \arrow[rr, "g"'] & & C \end{tikzcd} \tag{2}$$

This technique is central to proving many of the results in this paper, including the following lemma.

Lemma 2.6. *Consider the following commutative diagram in* $\mathcal{C}\mathrm{at}$ *with* $g\colon B \to C$ *a discrete opfibration:*

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "g\circ f"'] & & B \arrow[dl, "g"] \\ & C & \end{tikzcd} \tag{3}$$

Then:

(i) If $g \circ f$ is a discrete opfibration, then f is a discrete opfibration;

(ii) If $g \circ f$ has a lens structure, then f has a unique lens structure such that (3) *commutes in* $\mathcal{L}\mathrm{ens}$.

Proof. The first statement is a well-known property of discrete opfibrations. To prove the second statement, suppose $g \circ f$ has a lens structure given by the following commutative diagram of functors:

$$\begin{tikzcd} & X \arrow[dl, "\varphi"'] \arrow[dr, "\overline{\varphi}"] & \\ A \arrow[rr, "g\circ f"'] & & C \end{tikzcd}$$

Now consider the commutative diagram of functors:

$$\begin{tikzcd} & X \arrow[dl, "\varphi"'] \arrow[dr, "f\circ\varphi"] & \\ A \arrow[rr, "f"'] & & B \end{tikzcd}$$

For this to be a lens structure on f, we need to show that $f \circ \varphi$ is a discrete opfibration. However this follows from the first statement, since g is a discrete opfibration and $g \circ (f \circ \varphi) = \overline{\varphi}$ is a discrete opfibration. Using lens composition as in (2), noting that discrete opfibrations are diagrams (1) where φ is an isomorphism, it is not difficult to show that this lens structure makes the diagram (3) commute, and that the lens structure on f such that this holds is unique. □

3 Illustrative examples of lenses

In this section, we present two basic examples illustrating how lenses may arise in certain applications. These examples are not central to the purpose of this paper, but they may provide some concrete reference points for the abstract theory developed in the following sections.

State-transition machines as lenses

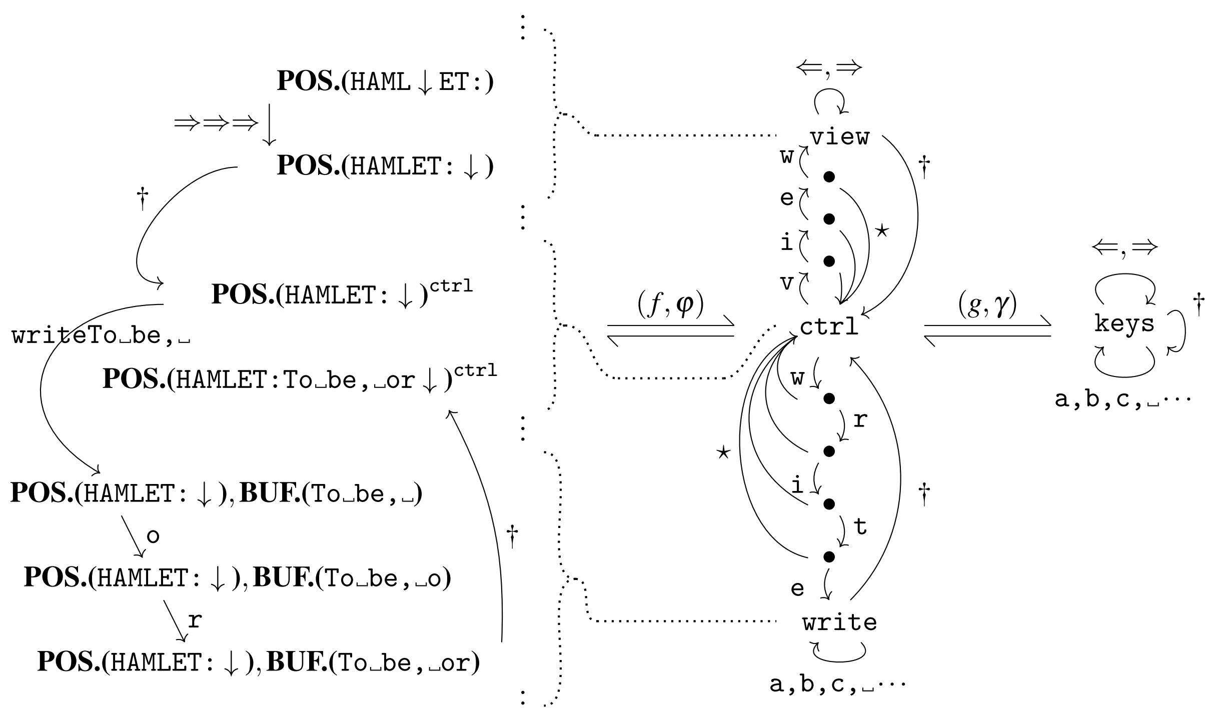
Let B be a free monoid considered as a one-object category, finitely generated by the set $\{\mathtt{b}_1, \mathtt{b}_2, \ldots, \mathtt{b}_N\}$ where we consider the labels $\mathtt{b}_i$ as *interface buttons* used to operate a machine.

A lens $(f, \varphi)\colon A \rightleftharpoons B$ can be understood as specifying a generalised state-transition machine, where the states are $\mathrm{Ob}(A)$, and the transitions are arrows of A labelled by their domains and elements of the monoid B. We examine this in more detail.

The underlying functor f maps arrows in A to strings of labels in B. The lift φ of the lens, given any object $a \in A$ and a transition label $\mathtt{b} \in B$, selects a morphism $\varphi(a, \mathtt{b})$ whose source is a.

The lifting operation φ of the lens takes an object of A, a state of the machine, and shows what state-transition will take place if button $\mathtt{b}_i$ is pressed when the machine is in that state.

In this example, the underlying functor f necessarily maps all objects of A to the single object of B, which suggests a natural generalisation. Indeed, the state-transition machine example extends to lenses with codomains of more than one object: the fibre of f over $b \in \mathrm{Ob}(B)$ consists of a *type* of states $f^{-1}(b) \subseteq \mathrm{Ob}(A)$, where the lens selects transitions out of $a \in f^{-1}(b)$ labelled by $B(b, -)$.



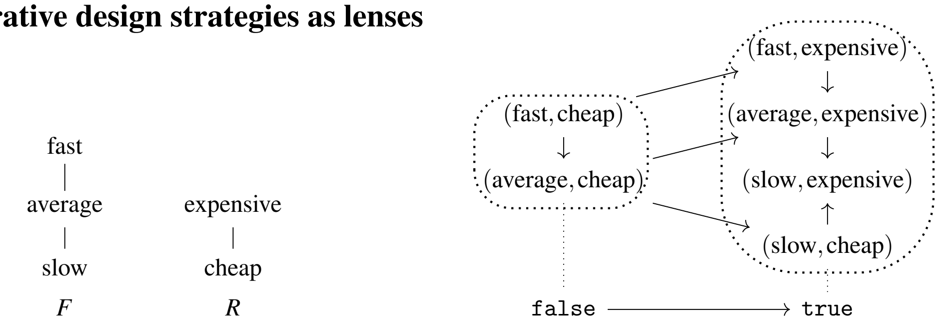
Example 3.1 ("typed" state-transition machines, and composition of lenses)**.** We sketch a rudimentary text-editor program operated by keystrokes from a keyboard. The *STATE* category where objects are internal states of the program might resemble the leftmost diagram above: objects are tuples of strings

with marked ($\downarrow$) cursor **positions** modelling text files, along with text **buffers** that hold onto strings of text to be inserted. We depict the path starting from the **POS.(**`HAML`$\downarrow$`ET:`**)** state in `view`-mode, and inputting the keyboard sequence $\Rightarrow\Rightarrow\Rightarrow$ †`writeTo␣be,␣or`†.

The program may have *modes of operation*, such that the same key on the keyboard has different functions depending on the current mode of operation. We depict the *MODE* category in the middle. In `view`-mode, arrow keys move the cursor's position through text. The special key † enters `control`-mode which keeps memories of cursor position intact, while awaiting strings `view` or `write` to switch to another mode; failed commands return to `control`-mode, notated by wildcard $\star$ arrows in the diagram. The `write`-mode allows alphabetic inputs to fill a temporary text buffer, the contents of which are appended to the main body of text upon returning to `control`-mode. We model the coordination between *STATE* and *MODE* as a lens, in fact a discrete opfibration, $(f,\varphi)\colon \mathit{STATE} \rightleftharpoons \mathit{MODE}$. The "typing" of states by modes arises from the fact that the fibre of f over `write` contains all states of the program accessible in `write`-mode, and similarly for the fibres of f above `ctrl` and `view`.

We model the *KEYBOARD* as a one-object category with generating endomorphisms of alphabetic keys `a,b,c,␣...`, arrow keys $\Leftarrow, \Rightarrow$ for navigation, and a command key †. The *MODE* category is a state-machine over *KEYBOARD*, so we coordinate the two with a lens $(g,\gamma)\colon \mathit{MODE} \rightleftharpoons \mathit{KEYBOARD}$. Altogether, we have a composition of lenses between categories $\mathit{STATE} \rightleftharpoons \mathit{MODE} \rightleftharpoons \mathit{KEYBOARD}$.

Collaborative design strategies as lenses



(a) Functionalities and resources.

(b) Fibre representation of a boolean profunctor.

The monotone theory of co-design presented in [1, 6] has found concrete applications in engineering, ranging from the design of intermodal mobility systems [14] to robotics and control [13, 15].

Let F be a poset representing *functionalities*, let R be a poset representing *costs* or *resources*, and let $\mathcal{B}\mathrm{ool}$ be the two element poset $\{\texttt{false} \to \texttt{true}\}$. A *boolean profunctor*, denoted by $F \nrightarrow R$, is a functor $F^{\mathrm{op}} \times R \to \mathcal{B}\mathrm{ool}$ which captures a relation between functionalities and requirements modelling feasibility, where decreasing demanded functionalities, or increasing resources, both increase feasibility.

Consider hiring an autonomous vehicle (AV): depending on how sophisticated the AV will be, the ride cost might change. Suppose F is the poset of performance grades of the AV, and R is the poset of ride costs (see (a) above). We define a boolean profunctor relating F and R following the rationale that the only cheap rides are slow rides, and to get average and fast rides one needs to pay more.

Objectwise, a boolean profunctor behaves as a judgement of whether each (F,R) pair is feasible, which is evident when we view the functor fibre-wise over $\mathcal{B}\mathrm{ool}$ (see (b) above). A lens structure on such a functor additionally provides, for each infeasible pair, a specified (reachable) feasible (F,R) pair. For instance, the pair $(\mathrm{average}, \mathrm{cheap})$ is infeasible. Possible ways to get feasible scenarios include accepting paying more (i.e. mapping to $(\mathrm{average}, \mathrm{expensive})$) or sacrificing performance (i.e. mapping to $(\mathrm{slow}, \mathrm{cheap})$). The lifting operation of a lens structure chooses one alternative.

Altogether, a lens in this setting models someone's design opinion: whether or not something is feasible, along with a *satisfaction strategy* that informs how to concretely compromise infeasible parameters, by either lowering demanded functionalities or increasing supplied resources.

4 Limits, colimits, and a factorisation system

In this section, we show that the category $\mathcal{L}\mathrm{ens}$ has a terminal object, an initial object, small coproducts, and equalisers. We also provide a characterisation of the monomorphisms and epimorphisms, and prove that $\mathcal{L}\mathrm{ens}$ has an (epi, mono)-factorisation system.

Proposition 4.1. *The category* $\mathcal{L}\mathrm{ens}$ *has a terminal object.*

Proof. The terminal object in $\mathcal{L}\mathrm{ens}$, as in $\mathcal{C}\mathrm{at}$, is the discrete category 1 with a single object. Given a category A, the unique lens $A \rightleftharpoons 1$ consists of the unique functor $! \colon A \to 1$ together with the trivial lifting operation. Following Proposition 2.5, this lens may be represented as the commutative diagram,

$$\begin{array}{ccc} & A_0 & \\ {}^{i}\swarrow & & \searrow^{!} \\ A & \xrightarrow[!]{} & 1 \end{array} \tag{4}$$

where $i \colon A_0 \to A$ is the inclusion of the discrete category A_0 of objects into A. □

Example 4.2 (The terminal interface)**.** The terminal object 1 in this setting is an interface with a single button (the identity) which does nothing. The lift of an identity is an identity, so pressing the button does not change the state of the machine. All machines are compatible with a 'do-nothing' interface.

Proposition 4.3. *The category* $\mathcal{L}\mathrm{ens}$ *has an initial object.*

Proof. The initial object in $\mathcal{L}\mathrm{ens}$, as in $\mathcal{C}\mathrm{at}$, is the empty category 0. Given a category A, the unique lens $0 \rightleftharpoons A$ consists of the unique functor $! \colon 0 \to A$ together with the trivial lifting operation.

$$\begin{array}{ccc} & 0 & \\ // & & \searrow^{!} \\ 0 & \xrightarrow[!]{} & A \end{array} \tag{5}$$

Following Proposition 2.5, this lens may be represented as the commutative diagram above. □

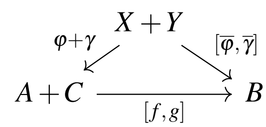
Example 4.4 (The initial machine)**.** The initial object 0 in this setting is the null machine with no internal states, which is compatible with any (unplugged) keyboard A.

Proposition 4.5. *The category* $\mathcal{L}\mathrm{ens}$ *has small coproducts.*

Proof. Given a pair of categories A and B, their coproduct $A+B$ in $\mathcal{L}\mathrm{ens}$ coincides with their coproduct in $\mathcal{C}\mathrm{at}$. The coproduct injections in $\mathcal{C}\mathrm{at}$ are discrete opfibrations, and therefore have a unique lens structure. To see that the universal property holds, consider a pair of lenses $(f,\varphi) \colon A \rightleftharpoons B$ and $(g,\gamma) \colon C \rightleftharpoons B$ represented as commutative diagrams following Proposition 2.5:

$$\begin{array}{ccc} & X & \\ {}^{\varphi}\swarrow & & \searrow^{\overline{\varphi}} \\ A & \xrightarrow[f]{} & B \end{array} \qquad\qquad \begin{array}{ccc} & Y & \\ {}^{\gamma}\swarrow & & \searrow^{\overline{\gamma}} \\ C & \xrightarrow[g]{} & B \end{array}$$

Since bijective-on-objects functors are closed under coproducts, and $\mathcal{D}\mathrm{opf}/B$ has coproducts, the unique lens $A+C \rightleftharpoons B$ is represented by the commutative diagram:



(6)

The above arguments extend to coproducts indexed by any set. □

Example 4.6 (Coproduct interfaces)**.** Consider A and C to be windowed programs that operate through a common interface B, a keyboard. The coproduct machine $A+C$ behaves as a window manager, that focuses on one window: functionally, the window manager forwards keystrokes from B to whichever of A or C is currently in focus.

Unlike the previous examples of limits and colimits, equalisers in $\mathcal{L}\mathrm{ens}$ are an example which does not coincide with the equaliser of the underlying functors in $\mathcal{C}\mathrm{at}$.

Proposition 4.7. *The category* $\mathcal{L}\mathrm{ens}$ *has equalisers.*

Proof (sketch). Consider a parallel pair of lenses $(f,\varphi)\colon A \rightleftharpoons B$ and $(g,\gamma)\colon A \rightleftharpoons B$, and construct the equaliser $j\colon E \to A$ of their underlying functors in $\mathcal{C}\mathrm{at}$. The equaliser of the parallel pair of lenses is the largest subobject $m\colon M \rightarrowtail E$ such that $j \circ m\colon M \to A$ is a discrete opfibration which forms a cone over the parallel pair in $\mathcal{L}\mathrm{ens}$. □

Example 4.8 (Equalising co-design strategies)**.** Consider a parallel pair of lenses $(f,\varphi)\colon F^{\mathrm{op}} \times R \rightleftharpoons \mathcal{B}\mathrm{ool}$ and $(g,\gamma)\colon F^{\mathrm{op}} \times R \rightleftharpoons \mathcal{B}\mathrm{ool}$ to model two experts' opinions on the design problem encoded by $F^{\mathrm{op}} \times R$. Their equaliser $E \rightleftharpoons F^{\mathrm{op}} \times R$ is an embedding of E into $F^{\mathrm{op}} \times R$, which selects all pairs in $F^{\mathrm{op}} \times R$ such that the feasibility judgements f and g agree, and moreover, such that the satisfaction strategies φ and γ concur. The equaliser always exists: in the worst case where there is total disagreement, $E = 0$.

Corollary 4.9. *In the category* $\mathcal{L}\mathrm{ens}$*, all idempotents split.*

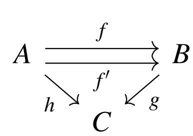
Proof. The splitting of an idempotent lens is given by the equaliser with the identity lens. □

Remark. Split idempotents are simple kinds of limits, but are interesting here for two reasons: they are also examples of coequalisers in $\mathcal{L}\mathrm{ens}$ (which are explored further in the paper by Di Meglio [4]) and they are also absolute (co)limits, meaning that they are examples of (co)equalisers which are preserved by any functor, in particular, by the forgetful functor $\mathcal{U}\colon \mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}$.

Both coproduct injections and equalisers are examples of monomorphisms in $\mathcal{L}\mathrm{ens}$. We now turn our attention to establishing sufficient conditions for a lens to be a monomorphism or an epimorphism.

Lemma 4.10. *If a lens is an injective-on-objects discrete opfibration, then it is a monomorphism.*

Proof. Every injective-on-objects discrete opfibration is also injective-on-morphisms, thus a monomorphism in $\mathcal{C}\mathrm{at}$. Consider the following diagram in $\mathcal{L}\mathrm{ens}$ (which omits the information of the lifting operation), consisting of a parallel pair of lenses f and f' which are equal to a lens h under post-composition by an injective-on-objects discrete opfibration g:



Since g is a monomorphism in $\mathcal{C}\text{at}$, the underlying functors of f and f' are equal. Furthermore, by Lemma 2.6, the lifting operations on f and f' are also equal. □

Proposition 4.11. *The functor* $\mathcal{U}\colon \mathcal{L}\text{ens} \to \mathcal{C}\text{at}$ *reflects monomorphisms.*

Proof. We need to show that if a lens $(f,\varphi)\colon A \rightleftharpoons B$ has an underlying functor f which is a monomorphism in $\mathcal{C}\text{at}$, then the lens is a monomorphism. Since such a lens is injective-on-objects, by Lemma 4.10 it suffices to show that it is also a discrete opfibration. Now for each pair $(a \in A, u\colon fa \to b \in B)$, there exists a unique morphism $\varphi(a,u)$ in A such that $f\varphi(a,u) = u$, since f is injective-on-morphisms. □

Lemma 4.12. *If a lens is surjective-on-objects, then it is an epimorphism.*

Proof. Consider a surjective-on-objects lens $(f,\varphi)\colon A \rightleftharpoons B$. Then (f,φ) must also be surjective-on-morphisms, since given any morphism $u\colon b \to b'$ in B, there exists an object a such that $fa = b$, and thus from the lifting operation a morphism $\varphi(a,u)\colon a \to a'$ in A such that $f\varphi(a,u) = u$. Therefore the underlying functor $f\colon A \to B$ is an epimorphism in $\mathcal{C}\text{at}$. Now consider a parallel pair of lenses $(g,\gamma)\colon B \rightleftharpoons C$ and $(g',\gamma')\colon B \rightleftharpoons C$ such that $g \circ f = g' \circ f$ and $\varphi(a,\gamma(fa,u)) = \varphi(a,\gamma'(fa,u))$ for all pairs $(a \in A, u\colon gfa \to c \in C)$. Then $g = g'$ since f is an epimorphism, and $\gamma(fa,u) = \gamma'(fa,u)$ since they are both equal to $f\varphi(a,\gamma(fa,u))$. □

Corollary 4.13. *The functor* $\mathcal{U}\colon \mathcal{L}\text{ens} \to \mathcal{C}\text{at}$ *reflects epimorphisms.*

Proof. This follows from Lemma 4.12, since every epimorphism in $\mathcal{C}\text{at}$ is surjective-on-objects. □

While Lemma 4.10 and Lemma 4.12 only provide sufficient conditions for monomorphisms and epimorphisms in $\mathcal{L}\text{ens}$, it is natural to wonder if they are also necessary conditions. This is indeed the case and is proved by Di Meglio [4]. Altogether, these results provide the following characterisation of monomorphisms and epimorphisms in $\mathcal{L}\text{ens}$.

Proposition 4.14. *A lens* $(f,\varphi)\colon A \rightleftharpoons B$ *is a monomorphism if and only if any of the following hold:*

1. (f,φ) *is an injective-on-objects discrete opfibration;*
2. (f,φ) *is a fully faithful discrete opfibration;*
3. f *is a monomorphism in* $\mathcal{C}\text{at}$.

Proposition 4.15. *A lens* $(f,\varphi)\colon A \rightleftharpoons B$ *is an epimorphism if and only if any of the following hold:*

1. f *is surjective-on-objects;*
2. f *is surjective-on-morphisms.*

It is surprising that unlike $\mathcal{C}\text{at}$, the epimorphisms in $\mathcal{L}\text{ens}$ admit a simple characterisation; epimorphisms in $\mathcal{L}\text{ens}$ are discussed further in [4]. Together, Proposition 4.14 and Proposition 4.15 have several consequences, including that $\mathcal{L}\text{ens}$ is a *balanced* category.

Corollary 4.16. *A lens is an isomorphism if and only if it is a monomorphism and an epimorphism.*

Proof. It is immediate that every bijective-on-objects (that is, both injective-on-objects and surjective-on-objects) discrete opfibration is an isomorphism, and conversely. □

In a recent paper by Johnson and Rosebrugh [9], it was noted that $\mathcal{L}\text{ens}$ admits a proper orthogonal factorisaton system. Using the above propositions this is actually an (epi, mono)-factorisation system, meaning that the left class is exactly the epimorphisms, and the right class is exactly the monomorphisms. We now provide a (new) proof of this result based on the following two known results.

Lemma A. *There is an orthogonal factorisation system on* $\mathcal{C}\mathrm{at}$ *which factors every functor into a surjective-on-objects functor followed by an injective-on-objects fully faithful functor.*

Lemma B. *There is an (epi, mono)-factorisation system on* $\mathcal{D}\mathrm{opf}$ *which factors every discrete opfibration into a surjective-on-objects discrete opfibration (epimorphism) followed by an injective-on-objects discrete opfibration (monomorphism).*

Note that the second lemma is a special case of the first, in the sense that the canonical inclusion functor $\mathcal{D}\mathrm{opf} \to \mathcal{C}\mathrm{at}$ preserves the factorisation system. We are now able to prove the following result.

Theorem 4.17. *The category* $\mathcal{L}\mathrm{ens}$ *has an orthogonal factorisation system which factors every lens into a surjective-on-objects lens (epimorphism) followed by a cosieve (monomorphism).*

Proof. Consider a lens $(f, \varphi)\colon A \rightleftharpoons B$ represented by the diagram (1). By Lemma B, we can factorise $\overline{\varphi}\colon X \to B$ into a surjective-on-objects discrete opfibration $j\colon X \to I$ followed by an injective-on-objects (fully faithful) discrete opfibration $k\colon I \to B$. By Lemma A, the orthogonality property induces a unique functor f' which is necessarily surjective-on-objects:

$$\begin{array}{ccc} X & \xrightarrow{\;j\;} & I \\ {\scriptstyle\varphi}\big\downarrow & \overset{f'}{\nearrow} & \big\downarrow{\scriptstyle k} \\ A & \xrightarrow[\;f\;]{} & B \end{array}$$

This provides the (epi, mono)-factorisation of the lens $(f, \varphi)\colon A \rightleftharpoons B$ as claimed.

To show this is an orthogonal factorisation system, consider the following diagram in $\mathcal{L}\mathrm{ens}$ where e is an epimorphism and m is a monomorphism:

$$\begin{array}{ccc} A & \xrightarrow{\;f\;} & C \\ {\scriptstyle e}\big\downarrow & & \big\downarrow{\scriptstyle m} \\ B & \xrightarrow[\;g\;]{} & D \end{array} \tag{7}$$

Considering the diagram (7) under the forgetful functor $\mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}$, by Lemma A there exists a unique functor $h\colon B \to C$ such that $h \circ e = f$ and $m \circ h = g$ in $\mathcal{C}\mathrm{at}$. Since m is a discrete opfibration, by Lemma 2.6 the functor h has a unique lens structure such that $m \circ h = g$ in $\mathcal{L}\mathrm{ens}$. Moreover, since m is a monomorphism in $\mathcal{L}\mathrm{ens}$, we also have that $h \circ e = f$ in $\mathcal{L}\mathrm{ens}$. This proves the claim of orthogonality. □

Remark. It is interesting to note that the forgetful functor $\mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}$ sends the (epi, mono)-factorisation in $\mathcal{L}\mathrm{ens}$ to both the orthogonal factorisation system on $\mathcal{C}\mathrm{at}$ stated in Lemma A, as well as the classical *image factorisation* of a functor.

Example 4.18 (BIOS / OS factorisation)**.** Recall that when interpreting lenses as state machines, the objects in the codomain of the lens can model *modes* or *types* of states in the domain. For a computer, such a codomain might look like the two-object category $\{\mathtt{BIOS} \to \mathtt{OS}\}$ with some additional endomorphisms. The arrow models the fact that the BIOS is encountered at startup, and if nothing is done to stay in the BIOS, there is a one-way transition into the OS where all everyday operations occur.

A software engineer who is only interested in the everyday operations is concerned only with the behaviour of the computer over the `OS` states. This leads to a factorisation of $\{\mathtt{EverydayOperation}\} \rightleftharpoons \{\mathtt{BIOS} \to \mathtt{OS}\}$ as the epimorphism of interest $\{\mathtt{EverydayOperation}\} \rightleftharpoons \{\mathtt{OS}\}$, followed by the embedding monomorphism $\{\mathtt{OS}\} \rightleftharpoons \{\mathtt{BIOS} \to \mathtt{OS}\}$.

5 Imported limits, distributivity, and extensivity

In this section, we introduce a notion of *imported limits*, and show that the category $\mathcal{L}\mathrm{ens}$ has imported products and imported pullbacks. While generally imported limits do not coincide with limits in $\mathcal{L}\mathrm{ens}$, we show that $\mathcal{L}\mathrm{ens}$ admits all products with discrete categories, and all pullbacks along discrete opfibrations. We also show that imported products and imported pullbacks in $\mathcal{L}\mathrm{ens}$ behave nicely with coproducts, proving that $\mathcal{L}\mathrm{ens}$ is a distributive and extensive category.

Definition 5.1. The *imported limit* of a diagram $\mathcal{D}\colon J \to \mathcal{L}\mathrm{ens}$ along the forgetful functor $\mathcal{U}\colon \mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}$ is a canonical cone $\Delta_{\mathcal{D}}$ over $\mathcal{D}$ such that $\mathcal{U} \circ \Delta_{\mathcal{D}}$ coincides with the limit of the diagram $\mathcal{U} \circ \mathcal{D}\colon J \to \mathcal{C}\mathrm{at}$.

Remark. The above definition is an attempt to describe the phenomenon where the projection functors from a limit in $\mathcal{C}\mathrm{at}$ (for example, products or pullbacks) have canonical lens structures, without explaining what is meant by *canonical*. A thorough investigation of this concept is planned for future work.

Every limit *created* by the forgetful functor $\mathcal{U}\colon \mathcal{L}\mathrm{ens} \to \mathcal{C}\mathrm{at}$ is an imported limit; for example, terminal objects and monomorphisms. The goal of this section is to consider two examples of imported limits which are *not necessarily* limits in $\mathcal{L}\mathrm{ens}$.

Imported products and distributivity

Possibly the simplest example of an imported limit in $\mathcal{L}\mathrm{ens}$, which is not a limit in general, is the imported product. In the literature, this has previously be called the *constant complement lens* [10].

Proposition 5.2. *The category* $\mathcal{L}\mathrm{ens}$ *has all imported products along the forgetful functor to* $\mathcal{C}\mathrm{at}$*.*

Proof. Given a pair of categories A and B, we need to show that the projection functors (for example, $\pi_0\colon A \times B \to A$) have a canonical lens structure. Using Proposition 2.5, the lens structure on the projection functor may be represented by the following diagram in $\mathcal{C}\mathrm{at}$,

$$\begin{array}{ccc} & A \times B_0 & \\ {\scriptstyle 1\times i}\swarrow & & \searrow{\scriptstyle \pi_0} \\ A \times B & \xrightarrow[\pi_0]{} & A \end{array} \tag{8}$$

where $i\colon B_0 \to B$ is the inclusion of the discrete category B_0 of objects into B. More explicitly, the lifting operation on π_0 is given by $\varphi((a,b), u\colon a \to a') = (u, 1_b)$. The above argument extends to imported products indexed by any set. □

Remark. In general, the imported product of a pair of categories is *not* the cartesian product in $\mathcal{L}\mathrm{ens}$, as the corresponding universal property does not hold. For example, given the imported product $A \times A$, there does not exist (in general) a unique lens $A \rightleftharpoons A \times A$ such that the composite with the projections yields identity lenses, since a lifting operation $\varphi(a \in A, (u\colon a \to x, v\colon a \to y) \in A \times A)$ is not well-defined unless $u = v$.

Despite the above remark, there are instances where the imported product in $\mathcal{L}\mathrm{ens}$ does coincide with the cartesian product in $\mathcal{L}\mathrm{ens}$.

Proposition 5.3. *The imported product $A \times B$ in* $\mathcal{L}\mathrm{ens}$ *corresponds with the cartesian product in* $\mathcal{L}\mathrm{ens}$ *if A or B is a discrete category.*

Proof. Consider the imported product $A \times B_0$ where B_0 is a discrete category. Then the projection lens $A \times B_0 \rightleftharpoons A$ defined in (8) is a discrete opfibration. Thus given any pair of lenses $(f, \varphi)\colon C \rightleftharpoons A$ and $(g, \gamma)\colon C \rightleftharpoons B_0$, the canonical functor $\langle f, g\rangle\colon C \to A \times B_0$ has a unique lens structure which commutes with the projection $A \times B_0 \rightleftharpoons A$ by Lemma 2.6. This unique lens structure also commutes with the other projection $A \times B_0 \rightleftharpoons B_0$. Therefore, $A \times B_0$ has the universal property of the product in $\mathcal{L}\text{ens}$. □

To show that $\mathcal{L}\text{ens}$ is distributive, we first need the following corollary of Proposition 5.2.

Corollary 5.4. *The category* $\mathcal{L}\text{ens}$ *has a semi-cartesian symmetric monoidal structure given by imported product, and the forgetful functor* $\mathcal{U}\colon \mathcal{L}\text{ens} \to \mathcal{C}\text{at}$ *is strong monoidal.*

Proposition 5.5. *The category* $\mathcal{L}\text{ens}$ *is a distributive monoidal category with respect to the imported product monoidal structure. In other words, imported products distribute over coproducts.*

Proof. We need to show that for all categories A, B, and C, the canonical lens,

$$[1 \times \iota_B, 1 \times \iota_C]\colon (A \times B) + (A \times C) \rightleftharpoons A \times (B + C)$$

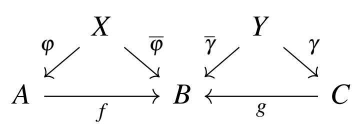
is an isomorphism, where $\iota_B\colon B \rightleftharpoons B + C$ and $\iota_C\colon C \rightleftharpoons B + C$ are the coproduct injections. Since $\mathcal{C}\text{at}$ is a distributive cartesian monoidal category, and the forgetful functor $\mathcal{U}\colon \mathcal{L}\text{ens} \to \mathcal{C}\text{at}$ is a strong monoidal isofibration by Corollary 5.4, the result follows immediately. □

Imported pullbacks and extensivity

We now turn our attention to imported pullbacks, one of the primary motivations for this paper.

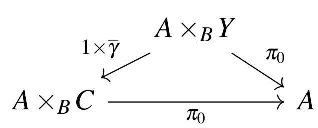
Proposition 5.6. *The category* $\mathcal{L}\text{ens}$ *has all imported pullbacks along the forgetful functor to* $\mathcal{C}\text{at}$.

Proof. Given a cospan of lenses represented as commutative diagrams,



(9)

we need to show that the pullback projection functors (for example, $\pi_0\colon A \times_B C \to A$) have a canonical lens structure such that $f \circ \pi_0 = g \circ \pi_1$ in $\mathcal{L}\text{ens}$. Following Proposition 2.5, the lens structure on the projection functor may be represented by the following diagram in $\mathcal{C}\text{at}$,



(10)

where $A \times_B Y$ is the pullback of f along $\overline{\gamma}$. More explicitly, the lifting operation on π_0 is given by:

$$((a, c) \in A \times_B C, u\colon a \to a' \in A) \qquad \longmapsto \qquad (u, \gamma(c, u))$$

Moreover the projection lenses defined above make the appropriate square in $\mathcal{L}\text{ens}$ commute. □

Example 5.7 (Pullbacks as independent components of a state machine)**.** Consider two state machines A and C over the same interface B, as lenses $A \rightleftharpoons B$ and $C \rightleftharpoons B$. The imported pullback lens $A \times_B C \rightleftharpoons B$ models a state-machine where the states are pairs $(a \in A, c \in C)$; it can be viewed as a state machine with two independent components A and C, which concurrently update according to inputs from interface B.

There is a close relationship between imported products and imported pullbacks.

Proposition 5.8. *Imported pullbacks over the terminal category correspond to imported products.*

We also have the following result, which generalises Corollary 5.4.

Corollary 5.9. *For each category B, the category* $\mathcal{L}\text{ens}/B$ *has a semi-cartesian monoidal structure given by imported pullback, and the forgetful functor* $\mathcal{U}/B\colon \mathcal{L}\text{ens}/B \to \mathcal{C}\text{at}/B$ *is strong monoidal.*

As with imported products, it is again natural to ask when the imported pullback in $\mathcal{L}$ens coincides with the categorical pullback in $\mathcal{L}$ens, leading to the following result which generalises Proposition 5.3.

Proposition 5.10. *The imported pullback* $A \times_B C$ *of the cospan* (9) *in* $\mathcal{L}$ens *corresponds with the categorical pullback in* $\mathcal{L}$ens *if* $f\colon A \to B$ *or* $g\colon C \to B$ *is a discrete opfibration.*

Proof. Suppose $g\colon C \to B$ in the cospan (9) is a discrete opfibration. Since discrete opfibrations are stable under pullback, the pullback projection (10) is a discrete opfibration. Then using Lemma 2.6, it is straightforward to show using an analogous argument to the proof of Proposition 5.3 that $A \times_B C$ has the universal property of the pullback in $\mathcal{L}$ens. □

Remark. It is natural to wonder if all pullbacks in $\mathcal{L}$ens are of the kind described in Proposition 5.10. There are examples where pullbacks exist along lenses which are not discrete opfibrations; however the details are outside the scope of this paper.

We are now able to prove the main theorem of this section.

Theorem 5.11. *The category* $\mathcal{L}$ens *is extensive.*

Proof. By Proposition 4.5, the category $\mathcal{L}$ens has finite coproducts. By Proposition 5.10, the category $\mathcal{L}$ens has pullbacks along discrete opfibrations, hence pullbacks along coproduct injections. Moreover, given any commutative diagram in $\mathcal{L}$ens of the form,

$$\begin{array}{ccccc} X & \longrightarrow & Z & \longleftarrow & Y \\ \downarrow & & \downarrow & & \downarrow \\ A & \underset{\iota_A}{\longrightarrow} & A+B & \underset{\iota_B}{\longleftarrow} & B \end{array} \tag{11}$$

the statement that the two squares are pullbacks if and only if the top row is a coproduct diagram follows directly, since $\mathcal{C}$at is extensive and the functor $\mathcal{U}\colon \mathcal{L}\text{ens} \to \mathcal{C}\text{at}$ is an identity-on-objects isofibration. □

6 Conclusion

This paper has begun the study of the category $\mathcal{L}$ens whose morphisms are lenses between small categories. We have presented results about limits, about some imported limits, and about coproducts, along with aspects of their interaction including extensivity. The work has continued apace with important findings by Di Meglio [4] who studies further colimits in $\mathcal{L}$ens.

The results have been surprising because the category of lenses, which is practically important but seemed rather ad hoc, turns out to have many aspects which are simpler than $\mathcal{C}$at, and some aspects which are surprisingly like the category of sets. In many respects imported limits interact well with one another, and with real limits and colimits.

So far we have only studied one category of lenses, but there are many more, including (2-)categories whose morphisms are symmetric lenses, split opfibrations, and discrete opfibrations. Future work aims to explore these categories and their interactions with $\mathcal{L}$ens, and to further clarify the role played by identity-on-objects isofibrations and limits and colimits imported along them.

References

[1] Andrea Censi (2016): *A Mathematical Theory of Co-Design*. Available at `https://arxiv.org/abs/1512.08055v7`.

[2] Bryce Clarke (2020): *Internal lenses as functors and cofunctors*. In: *Applied Category Theory 2019*, 323, pp. 183–195, doi:10.4204/EPTCS.323.13.

[3] 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`.

[4] Matthew Di Meglio (2021): *Coequalisers under the lens*. To appear in Applied Category Theory 2021.

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

[6] Brendan Fong & David I. Spivak (2019): *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press, Cambridge, doi:10.1017/9781108668804.

[7] Michael Johnson & Robert Rosebrugh (2016): *Unifying Set-Based, Delta-Based and Edit-Based Lenses*. In: *Proceedings of the 5th International Workshop on Bidirectional Transformations*, *CEUR Workshop Proceedings* 1571, pp. 1–13. Available at `http://ceur-ws.org/Vol-1571/paper_13.pdf`.

[8] Michael Johnson & Robert Rosebrugh (2017): *Symmetric delta lenses and spans of asymmetric delta lenses*. *Journal of Object Technology* 16(1), doi:10.5381/jot.2017.16.1.a2.

[9] Michael Johnson & Robert Rosebrugh (2021): *The more legs the merrier: A new composition for symmetric (multi-)lenses*. In: *Applied Category Theory 2020*, 333, pp. 92–107, doi:10.4204/EPTCS.333.7.

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

[11] Frank J. Oles (1982): *A category-theoretic approach to the semantics of programming languages*. Ph.D. thesis, Syracuse University. Available at `https://www.cs.cmu.edu/afs/cs.cmu.edu/project/fox-19/member/jcr/www/FrankOlesThesis.pdf`.

[12] Benjamin C. Pierce & Alan Schmitt (2003): *Lenses and view update translation*. Preprint. Available at `https://www.cis.upenn.edu/~bcpierce/papers/dblenses.pdf`.

[13] Gioele Zardini, Andrea Censi & Emilio Frazzoli (2021): *Co-Design of Autonomous Systems: From Hardware Selection to Control Synthesis*. In: *2021 20th European Control Conference (ECC)*, IEEE, doi:10.23919/ECC54610.2021.9654960.

[14] Gioele Zardini, Nicolas Lanzetti, Mauro Salazar, Andrea Censi, Emilio Frazzoli & Marco Pavone (2020): *On the Co-Design of AV-Enabled Mobility Systems*. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, doi:10.1109/ITSC45102.2020.9294499.

[15] Gioele Zardini, Dejan Milojevic, Andrea Censi & Emilio Frazzoli (2021): *Co-Design of Embodied Intelligence: A Structured Approach*. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, doi:10.1109/IROS51168.2021.9636513.