Internal lenses as functors and cofunctors

Bryce Clarke*

Centre of Australian Category Theory
Macquarie University, Australia

`bryce.clarke1@hdr.mq.edu.au`

Lenses may be characterised as objects in the category of algebras over a monad, however they are often understood instead as morphisms, which propagate updates between systems. Working internally to a category with pullbacks, we define lenses as simultaneously functors and cofunctors between categories. We show that lenses may be canonically represented as a particular commuting triangle of functors, and unify the classical state-based lenses with both c-lenses and d-lenses in this framework. This new treatment of lenses leads to considerable simplifications that are important in applications, including a clear interpretation of lens composition.

1 Introduction

Lenses form a mathematical structure that aims to capture the fundamental aspects of certain synchronisations between pairs of systems. The central goal of such synchronisation is to coherently propagate updates in one system to updates in another, and vice versa. The precise nature of the synchronisation process depends closely on the type of system being studied, and thus many different kinds of lenses have been defined to characterise various applications and examples.

Although a relatively recent subject for detailed abstract study, lenses are an impressive example of applied category theory, playing major roles in database view updating, in Haskell programs of many kinds, and in diverse examples of Systems Interoperations, Data Sharing, and Model-Driven Engineering. Thus, further clarifying the category-theoretic status and systematising the use of lenses, as this paper aims to do, is an important part of applied category theory.

Lenses were originally introduced [8] to provide a solution to the view-update problem [3]. In treatments of the view-update problem systems are generally modelled as a set of states, where it is possible to update from one state of the system to any other, and the only information retained about this update are its initial and final states. Thus a system may be understood as a *codiscrete category* on its set of states A with set of updates $A \times A$ given by a pair of initial and final states.

Lenses have long been recognised to be some kind of morphism between systems. An obvious notion of morphism between systems is simply a function $f \colon A \to B$ between their sets of states. Since systems may be modelled as codiscrete categories, there is also an induced function $f \times f \colon A \times A \to B \times B$ between the sets of updates of these systems. The map $f \colon A \to B$ is called the Get function and provides the first component of a lens between the systems A and B, often called the *source* and *view*.

The second component of a lens is called the Put function $p \colon A \times B \to A$ whose role is less obvious. The set $A \times B$ may be interpreted as the set of *anchored view updates* via the induced function $f \times 1_B \colon A \times B \to B \times B$ which produces a view update whose initial state is given by the Get function. The induced function $\langle \pi_0, p \rangle \colon A \times B \to A \times A$ may be regarded as the Put function, propagating every anchored view

---

update to a source update, illustrated in the diagram below.

$$
\begin{array}{ccc}
A & a \dashrightarrow p(a,b) \\
f\downarrow & \vdots \qquad\quad \vdots \\
B & fa \longrightarrow b
\end{array}
$$

Frequently the Get and Put functions of a lens are required to satisfy three additional axioms, called the *lens laws*, which ensure the synchronisation of updates between systems is well-behaved.

$$
\begin{array}{ccc}
A \times B \xrightarrow{\;p\;} A & A \xrightarrow{\langle 1_A, f\rangle} A \times B & A \times B \times B \xrightarrow{p \times 1_B} A \times B \\
\quad\pi_1 \searrow \;\; \downarrow f & \quad 1_A \searrow \;\; \downarrow p & \pi_{0,2}\downarrow \qquad\qquad \downarrow p \\
\qquad\quad B & \qquad\quad A & A \times B \xrightarrow[\;p\;]{} A
\end{array}
$$

In order from left to right: the Put-Get law ensures that the systems A and B are indeed synchronised under the Get and Put functions; the Get-Put law ensures that anchored view updates which are identities are preserved by the Put function; the Put-Put law ensures that composite anchored view updates are preserved under the Put function.

In summary, a *state-based lens* [8], denoted $(f,p)\colon A \rightleftharpoons B$, consists of a Get function $f\colon A \to B$ and a Put function $p\colon A \times B \to A$ satisfying the lens laws. Early mathematical work [14] characterised state-based lenses as algebras for a well-known monad,

$$
\begin{array}{ccc}
\mathbf{Set}\big/_B & \longrightarrow & \mathbf{Set}\big/_B \\
f\colon A \to B & \longmapsto & \pi_1\colon A \times B \to B
\end{array}
$$

which may be generalised to any category with finite products. It was later shown that lenses are also coalgebras for a comonad [9] and may be defined inside any cartesian closed category. While these works took the first steps towards internalisation of lenses, they characterised lenses as objects in the category of Eilenberg-Moore (co)algebras, rather than morphisms between sets, and did not account for composition of lenses.

A significant shortcoming of state-based lenses in many applications is they only describe synchronisation between systems as a set of states, or codiscrete categories, ignoring the information on how states are updated. This motivated the independent development of both *c-lenses* [15] and *d-lenses* [7] between systems modelled as arbitrary categories. Making use of comma categories instead of products, c-lenses were defined as algebras for a classical KZ-monad [18], and may be also understood as split Grothendieck opfibrations. In contrast d-lenses were shown [12] to be more general, as split opfibrations *without* the usual universal property, and could only be characterised as algebras for a *semi-monad* satisfying an additional axiom.

Later work [13] showed that the category of state-based lenses (as morphisms) is a full subcategory of the category of d-lenses (which also contains a subcategory of c-lenses). Despite this unification of category-based lenses, composition was still defined in an ad hoc fashion, and there was no mathematical explanation as to why lenses characterised as algebras should be understood as morphisms.

Summary of Paper

The contribution of this paper may be summarised as follows:

- Generalise the theory of lenses to be internal to any category $\mathscr{E}$ with pullbacks.

- Define an *internal lens* as an internal functor and an internal cofunctor, which provide the appropriate notion of Get and Put, respectively.

- Characterise internal lenses as diagrams of internal functors, using the span representation of an internal cofunctor.

- Show there is a well-defined category $\mathsf{Lens}(\mathscr{E})$ whose objects are internal categories and whose morphisms are internal lenses.

- Demonstrate state-based lenses, c-lenses, and d-lenses as examples of internal lenses.

2   Background

This section provides a brief review of the relevant internal category theory required for the paper, most of which can be found in standard references such as [4, 16, 17]. Throughout we work internal to a category $\mathscr{E}$ with pullbacks, with the main examples being $\mathscr{E} = \mathbf{Set}, \mathbf{Cat}$.

The idea is that a system may be defined as an internal category with an *object of states* and an *object of updates*. An internal functor will later be interpreted as the Get component of an internal lens, while internal discrete opfibrations will also be central in defining the Put component of an internal lens. Codiscrete categories and arrow categories are presented as examples and will later be used to define internal versions of state-based lenses and c-lenses.

Definition 1. An *internal category A* consists of an *object of objects* A_0 and an *object of morphisms* A_1 together with a span,

$$
\begin{array}{ccc}
 & A_1 & \\
{\scriptstyle d_1}\swarrow & & \searrow{\scriptstyle d_0} \\
A_0 & & A_0
\end{array}
\tag{1}
$$

where $d_1 \colon A_1 \to A_0$ is the *domain map* and $d_0 \colon A_1 \to A_0$ is the *codomain map*, and the pullbacks,

$$
\begin{array}{ccccc}
 & & A_2 & & \\
 & {\scriptstyle d_2}\swarrow & {\scriptstyle \vee} & \searrow{\scriptstyle d_0} & \\
A_1 & & & & A_1 \\
 & {\scriptstyle d_0}\searrow & & \swarrow{\scriptstyle d_1} & \\
 & & A_0 & &
\end{array}
\qquad
\begin{array}{ccccc}
 & & A_3 & & \\
 & {\scriptstyle d_3}\swarrow & {\scriptstyle \vee} & \searrow{\scriptstyle d_0} & \\
A_2 & & & & A_2 \\
 & {\scriptstyle d_0}\searrow & & \swarrow{\scriptstyle d_2} & \\
 & & A_1 & &
\end{array}
\tag{2}
$$

where A_2 is the *object of composable pairs* and A_3 is the *object of composable triples*, as well as an *identity map* $i_0 \colon A_0 \to A_1$ and *composition map* $d_1 \colon A_2 \to A_1$ satisfying the following commutative diagrams:

$$
\begin{array}{cc}
A_0 \xrightarrow{\ i_0\ } A_1 \\
{\scriptstyle i_0}\downarrow \ {\scriptstyle 1_{A_0}} \ \downarrow{\scriptstyle d_1} \\
A_1 \xrightarrow[\ d_0\ ]{} A_0
\end{array}
\quad
\begin{array}{ccc}
A_1 \xleftarrow{\ d_2\ } A_2 \xrightarrow{\ d_0\ } A_1 \\
{\scriptstyle d_1}\downarrow \quad \downarrow{\scriptstyle d_1} \quad \downarrow{\scriptstyle d_0} \\
A_0 \xleftarrow[\ d_1\ ]{} A_1 \xrightarrow[\ d_0\ ]{} A_0
\end{array}
\quad
\begin{array}{cc}
A_1 \xrightarrow{\ i_0\ } A_2 \\
{\scriptstyle i_1}\downarrow \ {\scriptstyle 1_{A_1}} \ \downarrow{\scriptstyle d_1} \\
A_2 \xrightarrow[\ d_1\ ]{} A_1
\end{array}
\quad
\begin{array}{cc}
A_3 \xrightarrow{\ d_1\ } A_2 \\
{\scriptstyle d_2}\downarrow \quad \downarrow{\scriptstyle d_1} \\
A_2 \xrightarrow[\ d_1\ ]{} A_1
\end{array}
\tag{3}
$$

The morphisms $i_0, i_1 \colon A_1 \to A_2$ and $d_1, d_2 \colon A_3 \to A_2$ appearing in (3) are defined using the universal property of the pullback A_2.
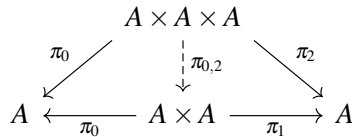
Example 2. A *small category* is an internal category in **Set**. Thus a small category consists of a *set* of objects and a *set* of morphisms, together with *functions* specifying the domain, codomain, identity, and composition.

Example 3. A (small) *double category* is an internal category in **Cat**, the category of small categories and functors. Thus a double category consists of a *category* of objects and a *category* of morphisms, together with *functors* specifying the domain, codomain, identity, and composition.
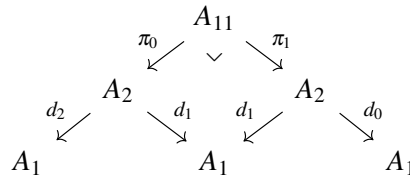
Example 4. Assume $\mathscr{E}$ has finite limits. A *codiscrete category* on an object $A \in \mathscr{E}$ is an internal category whose object of objects is A and whose object of morphisms is the product $A \times A$, with domain and codomain maps given by the left and right projections:

$$
\begin{array}{ccc}
 & A \times A & \\
\pi_0 \swarrow & & \searrow \pi_1 \\
A & & A
\end{array}
$$
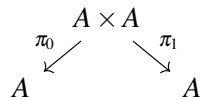
The identity map is given by the diagonal $\langle 1_A, 1_A \rangle \colon A \to A \times A$, the object of composable pairs is given by the product $A \times A \times A$, and the composition map is given by the following universal morphism:

$$
\begin{array}{ccccc}
 & & A \times A \times A & & \\
\pi_0 \swarrow & & \downarrow \pi_{0,2} & & \searrow \pi_2 \\
A & \xleftarrow{\ \pi_0\ } & A \times A & \xrightarrow{\ \pi_1\ } & A
\end{array}
$$

Example 5. Let A be an internal category. The *arrow category* ΦA has an object of objects A_1 and an object of morphisms $A_{11} := A_2 \times_{A_1} A_2$ defined by the pullback,

$$
\begin{array}{ccccccc}
 & & & A_{11} & & & \\
 & & \pi_0 \swarrow & \vee & \searrow \pi_1 & & \\
 & A_2 & & & & A_2 & \\
 d_2 \swarrow & & \searrow d_1 & & d_1 \swarrow & & \searrow d_0 \\
A_1 & & & A_1 & & & A_1
\end{array}
$$

with domain map $d_2\pi_0 \colon A_{11} \to A_1$ and codomain map $d_0\pi_1 \colon A_{11} \to A_1$. The pullback A_{11} may be understood as the *object of commutative squares* in A. The identity and composition maps require tedious notation to define precisely, however we note they are induced from the diagrams (3).

Definition 6. Let A and B be internal categories. An *internal functor* $f \colon A \to B$ consists of morphisms,

$$
f_0 \colon A_0 \longrightarrow B_0 \qquad\qquad f_1 \colon A_1 \longrightarrow B_1
$$

satisfying the following commutative diagrams:

$$
\begin{array}{ccccccccccccc}
A_0 & \xleftarrow{d_1} & A_1 & \xrightarrow{d_0} & A_0 & \quad & A_0 & \xrightarrow{i_0} & A_1 & \quad & A_2 & \xrightarrow{d_1} & A_1 \\
{\scriptstyle f_0}\downarrow & & {\scriptstyle f_1}\downarrow & & \downarrow{\scriptstyle f_0} & & {\scriptstyle f_0}\downarrow & & \downarrow{\scriptstyle f_1} & & {\scriptstyle f_2}\downarrow & & \downarrow{\scriptstyle f_1} \\
B_0 & \xleftarrow{d_1} & B_1 & \xrightarrow{d_0} & B_0 & & B_0 & \xrightarrow{i_0} & B_1 & & B_2 & \xrightarrow{d_1} & B_1
\end{array} \tag{4}
$$

The morphism $f_2 \colon A_2 \to B_2$ appearing in (4) is defined using the universal property of the pullback B_2.

Remark. Given an internal category A, the *identity functor* consists of a pair of morphisms:

$$1_{A_0}: A_0 \longrightarrow A_0 \qquad\qquad 1_{A_1}: A_1 \longrightarrow A_1$$

Given internal functors $f: A \to B$ and $g: B \to C$, their *composite functor* $g \circ f: A \to C$ consists of a pair of morphisms:

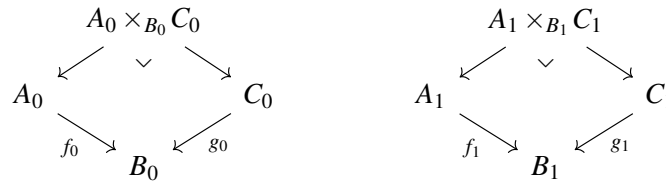$$g_0 f_0: A_0 \longrightarrow C_0 \qquad\qquad g_1 f_1: A_1 \longrightarrow C_1$$

Composition of internal functors is both unital and associative, as it is induced by composition of morphisms in $\mathscr{E}$.

Definition 7. Let $\mathsf{Cat}(\mathscr{E})$ be the category whose objects are internal categories and whose morphisms are internal functors.

Example 8. The category of sets and functions **Set** has pullbacks, thus we obtain the familiar example **Cat** = $\mathsf{Cat}(\mathbf{Set})$ of small categories and functors between them.

Example 9. The category **Cat** has pullbacks, so we obtain the category **Dbl** = $\mathsf{Cat}(\mathbf{Cat})$ of double categories and double functors between them.

Remark. The category $\mathsf{Cat}(\mathscr{E})$ has all pullbacks. Given internal functors $f: A \to B$ and $g: C \to B$, their pullback is the category $A \times_B C$ constructed from the pullbacks,

$$
\begin{array}{ccc}
& A_0 \times_{B_0} C_0 & \\
\swarrow \quad \vee \quad \searrow & \\
A_0 \qquad\qquad & & C_0 \\
_{f_0}\searrow \quad\quad \swarrow_{g_0} & \\
& B_0 &
\end{array}
\qquad\qquad
\begin{array}{ccc}
& A_1 \times_{B_1} C_1 & \\
\swarrow \quad \vee \quad \searrow & \\
A_1 \qquad\qquad & & C_1 \\
_{f_1}\searrow \quad\quad \swarrow_{g_1} & \\
& B_1 &
\end{array}
$$

which define the object of objects and object of morphisms, respectively. The rest of the structure is defined using the universal property of the pullback. Therefore *internal double categories* may be defined as categories internal to $\mathsf{Cat}(\mathscr{E})$.

Example 10. An *internal discrete opfibration* is an internal functor $f: A \to B$ such that the following diagram is a pullback:

$$
\begin{array}{ccc}
& A_1 & \\
_{d_1}\swarrow \quad & & \quad \searrow_{f_1} \\
A_0 & & B_1 \\
_{f_0}\searrow \quad & & \quad \swarrow_{d_1} \\
& B_0 &
\end{array}
$$
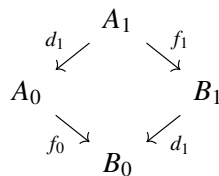
Note the identity functor is a discrete opfibration, and the composite of discrete opfibrations is a discrete opfibration, by the Pullback Pasting Lemma.

Definition 11. Let $\mathsf{DOpf}(\mathscr{E})$ be the category whose objects are internal categories and whose morphisms are discrete opfibrations.

3 Internal cofunctors

This section introduces the notion of an internal cofunctor and proves a useful representation of internal cofunctors as certain spans of internal functors. Since their introduction [1, 10] there has been almost

no work on cofunctors, apart from the recent reference [2]. To avoid confusion, we explicitly note that a cofunctor is *not* a contravariant functor.

The idea of a cofunctor is to generalise discrete opfibrations, providing a way to lift certain morphisms while preserving identities and composition. Cofunctors are dual to functors in the sense that they *lift* morphisms in the opposite direction to the object assignment, while functors *push-forward* morphisms in the same direction. In the context of synchronisation, a cofunctor will later be interpreted as the Put component of an internal lens which lifts anchored view updates in the pullback $\Lambda_1 := A_0 \times_{B_0} B_1$ to source updates in A_1.

Definition 12. Let A and B be internal categories. An *internal cofunctor* $\varphi \colon B \nrightarrow A$ consists of morphisms,

$$\varphi_0 \colon A_0 \longrightarrow B_0 \qquad\qquad \varphi_1 \colon \Lambda_1 \longrightarrow A_1 \qquad\qquad p_0 \colon \Lambda_1 \longrightarrow A_0$$

together with the pullbacks,

$$ (5) $$

such that the following diagrams commute:

$$ (6) $$

Remark. The pullback projections in (5) will play different roles which prompt different notational conventions. The projection $d_1 \colon \Lambda_1 \to A_0$ should be understood as the domain map for an internal category with object of morphisms Λ_1 which will be defined in Proposition 17. The projection $\overline{\varphi}_1 \colon \Lambda_1 \to B_1$ should be understood as morphism assignment for a discrete opfibration $\overline{\varphi}$ which will be defined in Theorem 18. The projections d_2 and $\overline{\varphi}_2$ for Λ_2 may be understood similarly.

Notation. The commutative diagrams (6) include morphisms defined using the universal property of the pullback via the diagrams below:

$$ (7) $$

Remark. Strictly speaking, the morphism $p_0 \colon \Lambda_1 \to A_0$ is not required for the definition of a cofunctor. Instead the two commutative diagrams in (6) which contain it may be replaced with the commutative

diagram:

$$
\begin{array}{ccc}
\Lambda_1 \xrightarrow{\ \varphi_1\ } A_1 \xrightarrow{\ d_0\ } A_0 \\
\Big\downarrow{\overline{\varphi}_1} \qquad\qquad \Big\downarrow{\varphi_0} \\
B_1 \xrightarrow{\qquad d_0 \qquad} B_0
\end{array}
\tag{8}
$$

Example 13. An internal cofunctor with $\varphi_1 \colon \Lambda_1 \cong A_1$ is a discrete opfibration.

Example 14. An internal cofunctor between monoids, as categories with one object, is a monoid homomorphism.

Example 15. An internal cofunctor with $\varphi_0 = 1_{A_0}$ is an identity-on-objects functor.

Remark. Given an internal category A, the *identity cofunctor* consists of morphisms:

$$
1_{A_0} \colon A_0 \longrightarrow A_0 \qquad 1_{A_1} \colon A_1 \longrightarrow A_1 \qquad d_0 \colon A_1 \longrightarrow A_0
$$

Given internal cofunctors $\varphi \colon B \nrightarrow A$ and $\gamma \colon C \nrightarrow B$, consisting of triples $(\varphi_0, \varphi_1, p_0)$ and $(\gamma_0, \gamma_1, q_0)$ respectively, their *composite cofunctor* $\varphi \circ \gamma \colon C \to A$ consists of the morphism,

$$
\gamma_0 \varphi_0 \colon A_0 \longrightarrow C_0
$$

together with the pullback $A_0 \times_{C_0} C_1$ and the morphisms,

$$
\varphi_1 \langle \pi_0, \gamma_1(\varphi_0 \times 1_{C_1}) \rangle \colon A_0 \times_{C_0} C_1 \longrightarrow A_1 \qquad p_0 \langle \pi_0, \gamma_1(\varphi_0 \times 1_{C_1}) \rangle \colon A_0 \times_{C_0} C_1 \longrightarrow A_0
\tag{9}
$$

where the universal morphisms are defined via the following commutative diagram:

$$\tag{10}$$

Composition of cofunctors is both unital and associative, however we omit the diagram-chasing required for the proof.

Definition 16. Let $\mathsf{Cof}(\mathcal{E})$ be the category whose objects are internal categories and whose morphisms are internal cofunctors.

Proposition 17. *If $\varphi \colon B \nrightarrow A$ is an internal cofunctor, then there exists an internal category Λ with object of objects A_0 and object of morphisms Λ_1, together with domain map $d_1 \colon \Lambda_1 \to A_0$, codomain map $p_0 \colon \Lambda_1 \to A_0$, identity map $i_0 \colon A_0 \to \Lambda_1$, and composition map $d_1 \colon \Lambda_2 \to \Lambda_1$.*

Proof. We give a partial proof and show the first pair of diagrams in (3) are satisfied. Using the relevant diagrams from Definition 1 and Definition 12 we have the following commutative diagram:

$$
\begin{array}{c}
\text{(commutative diagram with } A_0, A_1, \Lambda_1, A_0 \text{ and maps } 1_{A_0}, i_0, \varphi_1, d_0, d_1, p_0)
\end{array}
$$

This shows that the identity map $i\colon A_0 \to \Lambda_1$ is well-defined.

To show that Λ_2 is well-defined as the the pullback of the domain and codomain maps (left-most square below) we use the Pullback Pasting Lemma, noting that the outer rectangles below are equal:

$$
\begin{array}{ccc}
\Lambda_2 \xrightarrow{p_0} \Lambda_1 \xrightarrow{\overline{\varphi}_1} B_1 & & \Lambda_2 \xrightarrow{\overline{\varphi}_2} B_2 \xrightarrow{d_0} B_1 \\
\downarrow{\scriptstyle d_2} \quad \downarrow{\scriptstyle d_1} \quad\lrcorner\quad \downarrow{\scriptstyle d_1} & = & \downarrow{\scriptstyle d_2} \quad\lrcorner\quad \downarrow{\scriptstyle d_2} \quad\lrcorner\quad \downarrow{\scriptstyle d_1} \\
\Lambda_1 \xrightarrow{p_0} A_0 \xrightarrow{\varphi_0} B_0 & & \Lambda_1 \xrightarrow{\overline{\varphi}_1} B_1 \xrightarrow{d_0} B_0
\end{array}
$$

Again using the relevant diagrams from Definition 1 and Definition 12 we have the following commutative diagram:

$$
\begin{array}{c}
\text{(commutative diagram with } \Lambda_1, \Lambda_2, \Lambda_1, A_2, A_1, A_1, A_0, A_0, \Lambda_1, A_0 \text{ and maps } d_2, p_0, \varphi_2, d_0, \varphi_1, d_1, d_0, p_0, \varphi_1, d_1, d_0, p_0)
\end{array}
$$

This shows that the composition map $d_1\colon \Lambda_2 \to \Lambda_1$ is well-defined. $\qquad\square$

Remark. Proposition 17 may be understood as showing that a cofunctor induces a category whose objects are source states and whose morphisms are anchored view updates. The internal category Λ is shown in Theorem 18 to mediate between the source and the view, and reduces the complexity of Definition 12 to a simple statement concerning internal categories and functors.

Theorem 18. *If $\varphi\colon B \nrightarrow A$ is an internal cofunctor, then there is an internal discrete opfibration $\overline{\varphi}\colon \Lambda \to B$ consisting of the morphisms,*

$$\varphi_0\colon A_0 \longrightarrow B_0 \qquad\qquad \overline{\varphi}_1\colon \Lambda_1 \longrightarrow B_1$$

and an identity-on-objects internal functor $\varphi\colon \Lambda \to A$ consisting of morphisms:

$$1_{A_0}\colon A_0 \longrightarrow A_0 \qquad\qquad \varphi_1\colon \Lambda_1 \longrightarrow A_1$$

Thus every internal cofunctor $\varphi\colon B \nrightarrow A$ may be represented as a span of internal functors:

$$
\begin{array}{ccc}
 & \Lambda & \\
\overline{\varphi}\swarrow & & \searrow\varphi \\
B & & A
\end{array}
$$

Proof. To show that $\overline{\varphi}: \Lambda \to B$ is a well-defined internal discrete opfibration, we note from (5), (6), and (7) that the following diagrams commute:

$$
\begin{array}{ccccc}
A_0 & \xleftarrow{d_1} \Lambda_1 \xrightarrow{p_0} & A_0 & \quad & A_0 \xrightarrow{i_0} \Lambda_1 \quad\quad \Lambda_2 \xrightarrow{d_1} \Lambda_1 \\
\varphi_0 \downarrow & \quad \overline{\varphi}_1 \downarrow & \downarrow \varphi_0 & \quad & \varphi_0 \downarrow \quad\quad \downarrow \overline{\varphi}_1 \quad\quad \overline{\varphi}_2 \downarrow \quad\quad \downarrow \overline{\varphi}_1 \\
B_0 & \xleftarrow{d_1} B_1 \xrightarrow{d_0} & B_0 & \quad & B_0 \xrightarrow{i_0} B_1 \quad\quad B_2 \xrightarrow{d_1} B_1
\end{array}
$$

To show that $\varphi: \Lambda \to A$ is a well-defined identity-on-objects internal functor, we again note from (5), (6), and (7) that the following diagrams commute:

$$
\begin{array}{ccccc}
A_0 & \xleftarrow{d_1} \Lambda_1 \xrightarrow{p_0} & A_0 & \quad & A_0 \xrightarrow{i_0} \Lambda_1 \quad\quad \Lambda_2 \xrightarrow{d_1} \Lambda_1 \\
1_{A_0} \downarrow & \quad \varphi_1 \downarrow & \downarrow 1_{A_0} & \quad & 1_{A_0} \downarrow \quad\quad \downarrow \varphi_1 \quad\quad \varphi_2 \downarrow \quad\quad \downarrow \varphi_1 \\
A_0 & \xleftarrow{d_1} A_1 \xrightarrow{d_0} & A_0 & \quad & A_0 \xrightarrow{i_0} A_1 \quad\quad A_2 \xrightarrow{d_1} A_1
\end{array}
$$

Thus every internal cofunctor may be represented as a span of internal functors, with left-leg an internal discrete opfibration, and right-leg an identity-on-objects internal functor. $\square$

4 Internal Lenses

In this section we define an internal lens to consist of an internal Get functor and an internal Put cofunctor satisfying a simple axiom akin to the Put-Get law. An immediate corollary of Theorem 18 is that every internal lens may be understood as a particular commuting triangle (13) of internal functors. We also construct a category whose objects are internal categories and whose morphisms are internal lenses. The section concludes with a unification of discrete opfibrations, state-based lenses, c-lenses, and d-lenses in this internal framework, based upon results in [5].

Definition 19. An *internal lens* $(f, \varphi): A \rightleftharpoons B$ consists of an internal functor $f: A \to B$ comprised of morphisms,

$$f_0: A_0 \longrightarrow B_0 \qquad\qquad f_1: A_1 \longrightarrow B_1$$

and an internal cofunctor $\varphi: B \nrightarrow A$ comprised of morphisms,

$$\varphi_0: A_0 \longrightarrow B_0 \qquad\qquad \varphi_1: \Lambda_1 \longrightarrow A_1 \qquad\qquad p_0: \Lambda_1 \longrightarrow A_0$$

such that $\varphi_0 = f_0$ and the following diagram commutes:

$$
\begin{array}{ccc}
 & \Lambda_1 & \\
\varphi_1 \swarrow & & \searrow 1_{\Lambda_1} \\
A_1 & \xrightarrow[\langle d_1, f_1 \rangle]{} & \Lambda_1
\end{array}
\tag{11}
$$

Remark. Alternatively, the commutative diagram (11) for an internal lens may be replaced with the requirement that the following diagram commutes:

$$
\begin{array}{ccc}
 & \Lambda_1 & \\
\varphi_1 \swarrow & & \searrow \overline{\varphi}_1 \\
A_1 & \xrightarrow[f_1]{} & B_1
\end{array}
\tag{12}
$$

In either case, this axiom for an internal lens ensures that the functor and cofunctor parts interact as expected. Explicitly it states that *lifting* a morphism by the cofunctor then *pushing-forward* by the functor should return the original morphism.

Corollary 20. *Every internal lens* $(f, \varphi)\colon A \rightleftharpoons B$ *may be represented as a commuting triangle of internal functors,*

$$
\begin{array}{ccc}
 & \Lambda & \\
\varphi \swarrow & & \searrow \overline{\varphi} \\
A & \xrightarrow{\quad f \quad} & B
\end{array}
\tag{13}
$$

where $\overline{\varphi}\colon \Lambda \to B$ *is an internal discrete opfibration, and* $\varphi\colon \Lambda \to A$ *is an identity-on-objects internal functor.*

Corollary 21. *Given a pair of internal lenses* $(f, \varphi)\colon A \rightleftharpoons B$ *and* $(g, \gamma)\colon B \rightleftharpoons C$, *their composite internal lens may be computed via the composition of the respective functor and cofunctor parts, and has a simple representation using the pullback of internal functors:*

$$
\begin{array}{ccccc}
 & & \Lambda \times_B \Omega & & \\
 & \swarrow & \vee & \searrow & \\
 & \Lambda & & \Omega & \\
\varphi \swarrow & & \overline{\varphi} \searrow \quad \gamma \swarrow & & \searrow \overline{\gamma} \\
A & \xrightarrow{\quad f \quad} & B & \xrightarrow{\quad g \quad} & C
\end{array}
\tag{14}
$$

Definition 22. Let $\mathsf{Lens}(\mathscr{E})$ be the category whose objects are internal categories and whose morphisms are internal lenses. Composition of internal lenses is determined by composition of the corresponding functor and cofunctor parts.

Example 23. Every discrete opfibration is both an internal functor and an internal cofunctor, hence also an internal lens. Therefore $\mathsf{DOpf}(\mathscr{E})$ is a wide subcategory of $\mathsf{Lens}(\mathscr{E})$.

Example 24. If $\mathscr{E} = \mathbf{Set}$, then the category $\mathsf{Lens}(\mathbf{Set})$ is the category of d-lenses [7]. The Get of a d-lens $A \rightleftharpoons B$ is given by a functor $f\colon A \to B$, while the Put of a d-lens is given by a cofunctor $\varphi\colon B \nrightarrow A$.

In particular, the function $\varphi_1\colon \Lambda_1 \to A_1$ takes each pair $(a, u\colon fa \to b) \in \Lambda_1$ to a morphism $\varphi(a, u)\colon a \to p(a, u) \in A$, as illustrated in the diagram below.

$$
\begin{array}{ccc}
A & & a \xrightarrow{\ \varphi(a,u)\ } p(a,u) \\
f \Big\downarrow \Big\uparrow \varphi & & \vdots \qquad\qquad \vdots \\
B & & fa \xrightarrow{\quad u \quad} b
\end{array}
\tag{15}
$$

The Put-Get law is satisfied by (11), which corresponds in the above diagram to the morphism $\varphi(a, u)$ being a genuine lift of $u\colon fa \to b$ with respect to the functor acting on morphisms. The Get-Put and Put-Put laws are satisfied as $\varphi\colon \Lambda \to A$ is a functor, which respects identities and composition by definition.

Example 25. Every state-based lens (see [8]) consisting of Get function $f\colon A \to B$ and Put function $p\colon A \times B \to A$ induces a lens in $\mathsf{Lens}(\mathbf{Set})$.

Let $\widehat{A}$ and $\widehat{B}$ be the small codiscrete categories induced by the sets A and B, respectively, and let $f\colon \widehat{A} \to \widehat{B}$ be the canonical functor,

$$
\begin{array}{ccccc}
A & \xleftarrow{\ \pi_0\ } & A \times A & \xrightarrow{\ \pi_1\ } & A \\
f \downarrow & & \downarrow f \times f & & \downarrow f \\
B & \xleftarrow{\ \pi_0\ } & B \times B & \xrightarrow{\ \pi_1\ } & B
\end{array}
$$

induced by the Get function. Let Λ be the category with domain and codomain maps described by the span:

$$
\begin{array}{ccc}
 & A \times B & \\
\pi_0 \swarrow & & \searrow p \\
A & & A
\end{array}
$$

The category Λ is well-defined by the lens laws. The functor $\overline{\varphi} \colon \Lambda \to \widehat{B}$ is induced using the Put-Get law,

$$
\begin{array}{ccc}
A & \xleftarrow{\pi_0} A \times B \xrightarrow{\;p\;} & A \\
f\downarrow & \quad \llcorner \quad \downarrow f \times 1_B & \downarrow f \\
B & \xleftarrow{\pi_0} B \times B \xrightarrow{\pi_1} & B
\end{array}
$$

while the functor $\varphi \colon \Lambda \to \widehat{A}$ is induced for free:

$$
\begin{array}{ccc}
A & \xleftarrow{\pi_0} A \times B \xrightarrow{\;p\;} & A \\
1_A\downarrow & \downarrow \langle \pi_0, p\rangle & \downarrow 1_A \\
A & \xleftarrow{\pi_0} A \times A \xrightarrow{\pi_1} & A
\end{array}
$$

This example may be instantiated internal to any category $\mathscr{E}$ with finite limits.

Example 26. Given a pair of state-based lenses $(f,p) \colon A \rightleftharpoons B$ and $(g,q) \colon B \rightleftharpoons C$, their composite is a lens whose Get function is given by $gf \colon A \to C$ and whose Put function may be computed from the formula (9):

$$
p\langle \pi_0, q(f \times 1_C)\rangle \colon A \times C \longrightarrow A
$$

Example 27. Every c-lens (also known as a split opfibration, see [15]) consisting of a Get functor $f \colon A \to B$ and Put functor $p \colon f \downarrow B \to A$ induces a lens in $\mathsf{Lens}(\mathbf{Cat})$.

Let $\mathbb{B}$ be the double category of squares, whose category of objects is B and whose category of morphisms is the arrow category ΦB, together with domain and codomain functors $l, r \colon \Phi B \to B$ given by,

$$
\begin{array}{ccc}
B_1 & \xleftarrow{d_2\pi_0} B_{11} \xrightarrow{d_0\pi_1} & B_1 \\
d_1\downarrow & \downarrow d_2\pi_1 & \downarrow d_1 \\
B_0 & \xleftarrow{d_1} B_1 \xrightarrow{d_0} & B_0
\end{array}
\qquad
\begin{array}{ccc}
B_1 & \xleftarrow{d_2\pi_0} B_{11} \xrightarrow{d_0\pi_1} & B_1 \\
d_0\downarrow & \downarrow d_0\pi_0 & \downarrow d_0 \\
B_0 & \xleftarrow{d_1} B_1 \xrightarrow{d_0} & B_0
\end{array}
$$

using the same notation from the diagram in Example 5; define $\mathbb{A}$ similarly. Construct the functor $\Phi f \colon \Phi A \to \Phi B$ between the arrow categories,

$$
\begin{array}{ccc}
A_1 & \xleftarrow{d_2\pi_0} A_{11} \xrightarrow{d_0\pi_1} & A_1 \\
f_1\downarrow & \downarrow f_2 \times f_2 & \downarrow f_1 \\
B_1 & \xleftarrow{d_2\pi_0} B_{11} \xrightarrow{d_0\pi_1} & B_1
\end{array}
$$

induced by the Get functor, which forms a canonical double functor $f \colon \mathbb{A} \to \mathbb{B}$.

Let $\mathbb{A}$ be the double category with domain and codomain functors described by the span:

$$
\begin{array}{ccc}
 & f \downarrow B & \\
{}^{l}\swarrow & & \searrow{}^{p} \\
A & & A
\end{array}
$$

Note that the comma category $f \downarrow B$ may defined as the pullback,

$$
\begin{array}{ccc}
 & \overset{r}{\frown} & \\
f \downarrow B \longrightarrow & \Phi B \xrightarrow{\ r\ } & B \\
{}^{l}\downarrow \quad {}_{\lrcorner} & \downarrow{}^{l} & \\
A \xrightarrow{\ f\ } & B &
\end{array}
$$

where $l\colon f \downarrow B \to A$ and $r\colon f \downarrow B \to B$ are the usual comma category projections. The double category $\mathbb{A}$ is well-defined by the c-lens laws, and we may show with further reasoning that there exist unique double functors $\varphi\colon \mathbb{A} \to \mathbb{A}$ and $\overline{\varphi}\colon \mathbb{A} \to \mathbb{B}$.

5 Conclusion and Future Work

In this paper it was shown that lenses may be defined internal to any category $\mathscr{E}$ with pullbacks, providing a significantly generalised yet minimal framework to understand the notion of synchronisation between systems. It was demonstrated that the enigmatic Put of a lens may be understood as a cofunctor, which has a simple description as a span of a discrete opfibration and an identity-on-objects functor. The surprising characterisation of a lens as a functor/cofunctor pair both promotes the prevailing attitude of lenses as morphisms between categories, and yields a straightforward definition for composition in the category $\mathsf{Lens}(\mathscr{E})$, which fits within a diagram of forgetful functors.

$$
\begin{array}{ccccc}
 & & & \mathsf{Cof}(\mathscr{E})^{\mathsf{op}} & \\
 & & \nearrow & & \searrow \\
\mathsf{DOpf}(\mathscr{E}) \longrightarrow & \mathsf{Lens}(\mathscr{E}) & & & \mathscr{E} \\
 & & \searrow & & \nearrow \\
 & & & \mathsf{Cat}(\mathscr{E}) &
\end{array}
$$

The success of internal lenses in unifying the known examples of state-based lenses, c-lenses, and d-lenses promotes the effectiveness of this perspective for use in applications such programming, databases, and Model-Driven Engineering, and also anticipates many future mathematical developments. Current work in progress indicates that $\mathsf{Lens}(\mathscr{E})$ may be enhanced to a 2-category through incorporating natural transformations between lenses, while consideration of spans in $\mathsf{Lens}(\mathscr{E})$ leads towards a clarified understanding of symmetric lenses; both ideas which have been shown to be important in applications and the literature [6, 11]. In future work we will investigate examples of lenses internal to a diverse range of categories, as well as taking steps towards a theory of lenses between enriched categories.

Acknowledgements

References

[1] Marcelo Aguiar (1997): *Internal Categories and Quantum Groups*. Ph.D. thesis, Cornell University. Available at `http://pi.math.cornell.edu/~maguiar/thesis2.pdf`.

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

[3] F. Bancilhon & N. Spyratos (1981): *Update Semantics of Relational Views*. ACM Transactions on Database Systems 6(4), pp. 557–575, doi:10.1145/319628.319634.

[4] Francis Borceux (1994): *Handbook of Categorical Algebra*. Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, doi:10.1017/CBO9780511525858.

[5] Bryce Clarke (2018): *Characterising Asymmetric Lenses using Internal Categories*. Master's thesis, Macquarie University. Available at `http://hdl.handle.net/1959.14/1268984`.

[6] Zinovy Diskin (2017): *Compositionality of Update Propagation: Laxed PutPut*. In: *Proceedings of the 6th International Workshop on Bidirectional Transformations*, CEUR Workshop Proceedings 1827, pp. 74–89. Available at `http://ceur-ws.org/Vol-1827/paper12.pdf`.

[7] Zinovy Diskin, Yingfei Xiong & Krzysztof Czarnecki (2011): *From State- to Delta-Based Bidirectional Model Transformations: the Asymmetric Case*. Journal of Object Technology 10, pp. 6:1–25, doi:10.5381/jot.2011.10.1.a6.

[8] 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.

[9] Jeremy Gibbons & Michael Johnson (2012): *Relating Algebraic and Coalgebraic Descriptions of Lenses*. In: *Proceedings of the 1st International Workshop on Bidirectional Transformations*, Electronic Communications of the EASST 49, doi:10.14279/tuj.eceasst.49.726.

[10] 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), pp. 471–488, doi:10.1017/S0305004100071760.

[11] Martin Hofmann, Benjamin Pierce & Daniel Wagner (2011): *Symmetric Lenses*. SIGPLAN Not. 46(1), doi:10.1145/1925844.1926428.

[12] Michael Johnson & Robert Rosebrugh (2013): *Delta Lenses and Opfibrations*. Electronic Communications of the EASST 57, doi:10.14279/tuj.eceasst.57.875.

[13] 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`.

[14] Michael Johnson, Robert Rosebrugh & Richard Wood (2010): *Algebras and Update Strategies*. Journal of Universal Computer Science 16(5), pp. 729–748, doi:10.3217/jucs-016-05-0729.

[15] Michael Johnson, Robert Rosebrugh & Richard Wood (2012): *Lenses, Fibrations and Universal Translations*. Mathematical Structures in Computer Science 22(1), pp. 25–42, doi:10.1017/S0960129511000442.

[16] Peter T Johnstone (2002): *Sketches of an Elephant: A Topos Theory Compendium*. Oxford University Press.

[17] Saunders Mac Lane (1998): *Categories for the Working Mathematician*, 2nd edition. Graduate Texts in Mathematics 5, Springer-Verlag, New York, doi:10.1007/978-1-4757-4721-8.

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