Characterising Asymmetric Lenses using Internal Categories

By

Bryce Clarke
BSc(Hons)

A thesis submitted to Macquarie University
for the degree of Master of Research
Centre of Australian Category Theory (CoACT)
Department of Computing
12 December 2018

MACQUARIE
University
SYDNEY · AUSTRALIA

Typeset in $\LaTeX\,2_\varepsilon$.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Bryce Clarke

Acknowledgements

First and foremost, I would like to express my deep gratitude to my supervisor, Michael Johnson, for nurturing my growth as a mathematician over the course of my MRes, for allowing me the freedom to explore the rich universe of category theory, and for always sharing in the excitement of new ideas.

I would also like to thank: Daniel Lin, Yuki Maehara, Adrian Miranda, and Branko Nikolic, for the interesting discussion and friendly conversation throughout our regular Friday reading group; everyone who participated in the Australian Category Seminar each Wednesday, for sharing their expertise and knowledge; and all my friends and colleagues in E6D, especially Ori Livson and Chakaveh Saedi, for making every other day of the week something to look forward to.

Finally, I would like to thank my parents, for their continued love and support as I take on new challenges.

Abstract

Asymmetric lenses were originally defined in Computer Science as a solution to the view update problem, and are mathematically well understood as a generalisation of split opfibrations. In this thesis, we utilise internal category theory to unify three kinds of asymmetric lens — set-based, c-lenses, and d-lenses — through the construction of an internal category of view updates produced using the well-known lens laws. We show that this category forms the head of a span of internal functors, which induces a commutative triangle with the Get of a lens. The composition of these commuting triangles is used to characterise the three categories **Lens**, **Clens**, and **Dlens**.

Contents

1 Introduction

Lenses were developed as a mathematical way to describe solutions to the view update problem [1] which arises when utilising bidirectional transformations in Computer Science. The definition of a *lens* [2] consists of a pair of functions,

$$g \colon S \longrightarrow V \qquad\qquad p \colon S \times V \longrightarrow S$$

called the Get and Put, respectively, such that for all so-called *states* $s \in S$ and $v, u \in V$, the following axioms, frequently called the *lens laws*, are satisfied:

$$g(p(s,v)) = v \tag{Put-Get}$$

$$p(s, g(s)) = s \tag{Get-Put}$$

$$p(p(s,v),u) = p(s,u) \tag{Put-Put}$$

The notion of a lens captures the central purpose of a bidirectional transformation to maintain consistency, via the Put function p, between the *source* S and *view* V, related via the Get function g. It was later shown in [3] that lenses arise both as algebras for a monad and as coalgebras for a comonad.

The primary assumption underlying a lens is that there exists exactly one way to transition between states, in both the source and view. In 2012, a categorial generalisation was provided to relax this assumption and allow the possibility of many transitions between states. A *c-lens* [4] consists of a pair of functors,

$$G \colon \mathbf{S} \longrightarrow \mathbf{V} \qquad\qquad P \colon (G \downarrow \mathbf{V}) \longrightarrow \mathbf{S}$$

called the Get and Put, respectively, where $(G \downarrow \mathbf{V})$ is the comma category whose objects are pairs $(S, \alpha \colon GS \to V)$ called *view updates* and whose morphisms are pairs $\langle f, g\rangle \colon (S, \alpha) \to (S', \alpha')$, consisting of an arrow of $\mathbf{S}$ and an arrow of $\mathbf{V}$, such that $g \circ \alpha = \alpha' \circ Gf$. Analogously to lenses, a c-lens is also required to satisfy three lens laws:

$$GP(S,\alpha) = V \qquad\qquad GP\langle f, g\rangle = g \colon V \to V' \tag{Put-Get}$$

$$P(S, 1_{GS}) = S \qquad\qquad P\langle f, Gf\rangle = f \colon S \to S' \tag{Get-Put}$$

$$P(P(S,\alpha),\beta) = P(S, \beta\alpha) \qquad\qquad P\langle P\langle f, g\rangle, h\rangle = P\langle f, h\rangle \tag{Put-Put}$$

The definition of a c-lens is equivalent to that of a split Grothendieck opfibration, and c-lenses also arise as algebras for a well-known KZ-monad [5]. Furthermore the splitting functor,

$$K \colon (G \downarrow \mathbf{V}) \longrightarrow \Phi\mathbf{S}$$

induced by a c-lens, which assigns to each view update its opcartesian lift in the arrow category $\Phi\mathbf{S}$, is a left-adjoint right-inverse functor, ensuring a least-change solution to the view update problem.

While c-lenses are the correct generalisation of lenses mathematically, least-change updates may be difficult to establish in practical contexts. In 2011, prior to the introduction of c-lenses to the bidirectional transformation community, [6] presented another category-based generalisation of set-based lenses. A *d-lens* consists of a Get functor $G \colon \mathbf{S} \to \mathbf{V}$ together with an unnamed function[1],

$$k \colon \operatorname{obj}(G \downarrow \mathbf{V}) \to \operatorname{obj}(\Phi\mathbf{S}) \qquad (S, \alpha) \mapsto k(S, \alpha) \colon S \to p(S, \alpha)$$

which defines the Put function $p \colon \operatorname{obj}(G \downarrow \mathbf{V}) \to \operatorname{obj}(\mathbf{S})$ by taking the codomain of the output, and is required to satisfy three lens laws:

$$Gk(S, \alpha) = \alpha \colon GS \to V \tag{Put-Get}$$
$$k(S, 1_{GS}) = 1_S \colon S \to S \tag{Get-Put}$$
$$k(p(S, \alpha), \beta) \circ k(S, \alpha) = k(S, \beta\alpha) \tag{Put-Put}$$

In the paper [7] it was shown that every c-lens induces a d-lens, simply by taking the underlying object assignment of the splitting functor, and thus a d-lens may be seen as a c-lens without a universal property.

In many ways, the mathematical foundation of d-lenses is unsatisfying. While both lenses and c-lenses arise naturally as algebras for a monad, d-lenses only fit awkwardly into this mould as algebras for a semi-monad [7]. The Put for a d-lens is the byproduct of another function which lifts view updates (S, α) to source updates $k(S, \alpha)$, while in lenses and c-lenses the Put takes view updates to source states $p(s, v)$ or $P(S, \alpha)$, and the corresponding source updates are canonically induced. While there has been work done characterising lenses as d-lenses between codiscrete categories [8], and characterising c-lenses as d-lenses with a universal property on the lifts [7], as well as a growing amount of research into symmetric counterparts [9–11], there has been little success in finding a cohesive mathematical framework for lenses, c-lenses, and d-lenses.

Plan of the thesis

The goal of this thesis is to unify lenses, c-lenses, and d-lenses within a common mathematical framework using internal category theory. We now outline the structure of the thesis and highlight the key contributions.

- ◇ In Chapter 2 we present the background material on internal categories, focusing on **Set** and **Cat**, and fix notational conventions.

[1] In the literature, k is called the Put while the codomain remains unnamed; we reverse this convention to align with the notation and terminology used for lenses and c-lenses.

- ◇ In Chapter 3 we develop the internal theory of *set-based lenses* (often just *lenses* when the context is clear) working internal to **Set**. In Theorem 2 we prove that every lens induces a small category $\mathbf{\Lambda} = (S, S \times V)$ using the Get-Put and Put-Put laws. In Corollary 3 we obtain the result that every lens is equivalent to a commuting triangle of small categories and functors. These results allow us to provide a novel description for composition in the category **Lens** whose objects are sets and whose morphisms are lenses.

- ◇ In Chapter 4 we develop the internal theory of *c-lenses* working internal to **Cat**. In Theorem 4 we prove that every c-lens induces a double category $\mathbb{\Lambda} = (\mathbf{S}, (G \downarrow \mathbf{V}))$ which is moreover shown to be equivalent to a commuting triangle of double categories and double functors in Corollary 6. These results allow us to provide a novel description for composition in the category **Clens** whose objects are small categories and whose morphisms are c-lenses.

- ◇ In Chapter 5 we develop the internal theory of *d-lenses* working internal to **Set**. In Theorem 7 we prove that every d-lens induces a small category $\mathbf{\Lambda} = (S_0, S_0 \times_{V_0} V_1)$ which is moreover shown to be equivalent to a commuting triangle of small categories and functors in Corollary 8. These results allow us to provide a novel description for composition in the category **Dlens** whose objects are small categories and whose morphisms are d-lenses.

- ◇ In the Conclusion we summarise our results, discuss the remarkable similarities and subtle differences between the three kinds of lens, and explore some ideas towards future work in area of lenses and internal category theory.

As illustrated above, we have strived to find and present parallel treatments of the three kinds of lenses, allowing us to present each chapter with a similar structure and consistent notation.

Notation

We now provide some context for notation and terminology common in the Computer Science community for the mathematical audience.

Throughout we use symbol Λ to denote a lens as a whole; for example as both a quadruple $\Lambda = (S, V, g, p)$, and as a morphism $\Lambda\colon S \rightleftharpoons V$. Both of these notations are common in the bidirectional transformation community, and aim to represent a lens as consisting of a *source* and a *view*, together with operations, Get and Put, which work in opposite directions (justifying the notation $\rightleftharpoons$ for a lens). Later we use the notation Λ_1 for the so-called *object of view updates*, thus unifying the apparently different domains for the Put for each set-based lenses, c-lenses, and d-lenses.

We exclusively use the symbols g or G for the Get of a lens. The Put of a lens is commonly denoted by p or P in the literature, while in this thesis it often has an attached subscript, for example p_1 or P_1, indicating it is also the codomain map of an internal category; both of these notations are interchangeable.

Finally we remark that, as seen above, each kind of lens must satisfy some basic axioms, called the Put-Get, Get-Put, and Put-Put laws. These names, introduced by the computer scientists, are intended to indicate the results of computing a Put followed by a Get (Put-Get), a Get followed by a Put (at least in the set-based case where the name Get-Put was first introduced), and the Put done successively (Put-Put).

2
Background

This chapter provides an account of the relevant internal category theory required for the following chapters. There is nothing original presented, and experienced category theorists may prefer to briefly skim this chapter simply to see our notational conventions.

We establish the notational convention of reusing labels for similar morphisms, allowing them instead to be distinguished diagrammatically by their domain and codomain; for example, the domain, codomain, identity and composition maps will frequently use the same labels (l, r, i, c). Other common conventions include using $\Phi\mathbf{C}$ for the arrow category of $\mathbf{C}$, and using $(G \downarrow \mathbf{V})$ for the comma category $(G \downarrow 1_{\mathbf{V}})$.

The first section on small categories will be pertinent for both set-based lenses and d-lenses, while the section on codiscrete categories is used exclusively for set-based lenses. The remaining sections will form the core material needed to discuss c-lenses.

2.1 Small categories

Let **Set** be the category whose objects are sets and whose morphisms are functions. A considerable amount of category theory can be studied internal to **Set**, where the *collections* of objects and morphisms of a category are actually *sets*, and the set of composable morphisms is constructed via pullback. This notion is made rigorous with the definition of a small category and functors between them.

Definition 1. A *small category* $\mathbf{C}$, or a *category internal to* **Set**, consists of a set of objects C_0 and a set of morphisms C_1 together with functions

$$\begin{tikzcd} C_0 \arrow[r, "i"] & C_1 \arrow[l, bend right, "r_1"'] \arrow[l, bend left, "l_1"] & C_2 \arrow[l, "c"'] \arrow[l, bend right, "r_2"'] \arrow[l, bend left, "l_2"] & C_3 \arrow[l, shift right, "r_3"'] \arrow[l, shift left, "l_3"] \end{tikzcd}$$

where

- The function i assigns each object to its identity morphism;
- The functions l_1 and r_1 assign a morphism to its domain and codomain objects, respectively;
- The set C_2 is the set of composable pairs of morphisms, together with projection functions l_2 and r_2 onto the first and second component morphisms, respectively;
- The function c assigns a pair of morphisms in C_2 to their composite morphism;
- The set C_3 is the set of composable triples of morphisms, together with projection functions l_3 and r_3 onto the first and second pair of composable morphisms, respectively;
- The sets C_2 and C_3 are defined by the pullback diagram below:

$$\begin{tikzcd} & & C_3 \arrow[dl, "l_3"'] \arrow[dr, "r_3"] \arrow[dd, phantom, "\vee"] & & \\ & C_2 \arrow[dl, "l_2"'] \arrow[dr, "r_2"] \arrow[dd, phantom, "\vee"] & & C_2 \arrow[dl, "l_2"'] \arrow[dr, "r_2"] \arrow[dd, phantom, "\vee"] & \\ C_1 \arrow[dr, "r_1"'] & & C_1 \arrow[dl, "l_1"] \arrow[dr, "r_1"'] & & C_1 \arrow[dl, "l_1"] \\ & C_0 & & C_0 & \end{tikzcd}$$

These functions are required to satisfy the following commutative diagrams

$$\begin{tikzcd} C_0 \arrow[r, "i"] \arrow[d, "i"'] & C_1 \arrow[d, "r_1"] \\ C_1 \arrow[r, "l_1"'] \arrow[ur, phantom] & C_0 \end{tikzcd} \quad \begin{tikzcd} C_0 \arrow[dr, "1"] & \\ & C_0 \end{tikzcd} \qquad \begin{tikzcd} C_1 \arrow[d, "l_1"'] & C_2 \arrow[l, "l_2"'] \arrow[r, "r_2"] \arrow[d, "c"] & C_1 \arrow[d, "r_1"] \\ C_0 & C_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] & C_0 \end{tikzcd} \qquad \begin{tikzcd} C_1 \arrow[r, "\langle 1, i\, r_1 \rangle"] \arrow[d, "\langle i\, l_1, 1 \rangle"'] \arrow[dr, "1"] & C_2 \arrow[d, "c"] \\ C_2 \arrow[r, "c"'] & C_1 \end{tikzcd} \qquad \begin{tikzcd} C_3 \arrow[r, "1 \times c"] \arrow[d, "c \times 1"'] & C_2 \arrow[d, "c"] \\ C_2 \arrow[r, "c"'] & C_1 \end{tikzcd}$$

which respectively determine:

- The domain and codomain of identity morphisms;
- The domain and codomain of composite morphisms;
- The left and right composition with an identity morphism;
- The associativity of composition.

Definition 2. Let **C** and **D** be small categories. A *functor* $F\colon \mathbf{C} \to \mathbf{D}$ internal to **Set** consists of a pair of functions

$$f_0\colon C_0 \longrightarrow D_0 \qquad f_1\colon C_1 \longrightarrow D_1$$

satisfying the following commutative diagrams:

$$\begin{tikzcd} C_0 \arrow[d, "f_0"'] & C_1 \arrow[l, "l_1"'] \arrow[r, "r_1"] \arrow[d, "f_1"] & C_0 \arrow[d, "f_0"] \\ D_0 & D_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] & D_0 \end{tikzcd} \qquad \begin{tikzcd} C_0 \arrow[r, "i"] \arrow[d, "f_0"'] & C_1 \arrow[d, "f_1"] & C_2 \arrow[l, "c"'] \arrow[d, "f_1 \times f_1"] \\ D_0 \arrow[r, "i"'] & D_1 & D_2 \arrow[l, "c"] \end{tikzcd}$$

The *identity functor* $1_{\mathbf{C}}\colon \mathbf{C} \to \mathbf{C}$ for a small category **C** consists of the functions:

$$1_{C_0}\colon C_0 \to C_0 \qquad 1_{C_1}\colon C_1 \to C_1$$

Let **Cat** be the category whose objects are small categories and whose morphisms are functors. The 1-category **Cat** may be promoted to a 2-category with the definition of natural transformations between functors, together with the notions of whiskered, vertical, and horizontal composition.

Definition 3. Let $F, G\colon \mathbf{C} \to \mathbf{D}$ be functors between small categories. A *natural transformation* $\phi\colon F \Rightarrow G$ internal to **Set** consists of a function

$$\phi\colon C_0 \longrightarrow D_1$$

satisfying the following commutative diagrams:

$$\begin{tikzcd} & C_0 \arrow[dl, "f_0"'] \arrow[d, "\phi"] \arrow[dr, "g_0"] & \\ D_0 & D_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] & D_0 \end{tikzcd} \qquad \begin{tikzcd} C_1 \arrow[r, "\langle f_1, \phi r_1 \rangle"] \arrow[d, "\langle \phi l_1, g_1 \rangle"'] & D_2 \arrow[d, "c"] \\ D_2 \arrow[r, "c"'] & D_1 \end{tikzcd}$$

Definition 4. Given a diagram of functors and a natural transformation,

$$\begin{tikzcd} \mathbf{A} \arrow[r, "F"] & \mathbf{B} \arrow[r, bend left, "G"] \arrow[r, bend right, "H"'] \arrow[r, phantom, "\Downarrow \phi"] & \mathbf{C} \arrow[r, "K"] & \mathbf{D} \end{tikzcd}$$

their *whiskered composite* natural transformation $K\phi F\colon KGF \Rightarrow KHF$ is defined as the composite function:

$$A_0 \xrightarrow{f_0} B_0 \xrightarrow{\phi} C_1 \xrightarrow{k_1} D_1$$

Definition 5. Given a diagram of functors and natural transformations,

$$\begin{tikzcd} \mathbf{C} \arrow[r, bend left=60, "F"] \arrow[r, "G" description] \arrow[r, bend right=60, "H"'] \arrow[r, phantom, shift left=3, "\Downarrow \phi"] \arrow[r, phantom, shift right=3, "\Downarrow \psi"] & \mathbf{D} \end{tikzcd}$$

their *vertical composite* natural transformation $\psi \bullet \phi\colon F \Rightarrow H$ is defined as the composite function:

$$C_0 \xrightarrow{\langle \phi, \psi \rangle} D_2 \xrightarrow{c} D_1$$

Definition 6. Given a diagram of functors and natural transformations

$$\begin{tikzcd} \mathbf{A} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow\phi"] & \mathbf{B} \arrow[r, bend left, "H"] \arrow[r, bend right, "K"'] \arrow[r, phantom, "\Downarrow\psi"] & \mathbf{C} \end{tikzcd}$$

their *horizontal composite* natural transformation $\psi * \phi \colon HF \Rightarrow KG$ is defined using whiskering and vertical composition as the diagonal of the commutative square:

$$\begin{tikzcd} A_0 \arrow[r, "\langle h_1\phi, \psi g_0\rangle"] \arrow[d, "\langle \psi f_0, k_1\phi\rangle"'] \arrow[dr, dashed, "\psi*\phi"] & C_2 \arrow[d, "c"] \\ C_2 \arrow[r, "c"'] & C_1 \end{tikzcd}$$

The *vertical identity natural transformation* $1_F \colon F \Rightarrow F$ for a functor $F \colon \mathbf{C} \to \mathbf{D}$ is defined by the diagonal of the commutative square,

$$\begin{tikzcd} C_0 \arrow[r, "i"] \arrow[d, "f_0"'] \arrow[dr, dashed, "1_F"] & C_1 \arrow[d, "f_1"] \\ D_0 \arrow[r, "i"'] & D_1 \end{tikzcd}$$

while the *horizontal identity natural transformation* $1_{1_\mathbf{C}} \colon 1_\mathbf{C} \Rightarrow 1_\mathbf{C}$ for a small category $\mathbf{C}$ is defined by the identity map $i \colon C_0 \to C_1$.

It can be shown that whiskered, vertical, and horizontal composites all interact nicely to yield a 2-category **Cat** of small categories, functors, and natural transformations. We conclude with the notion of adjunction, arguably the most important concept within the 2-category **Cat**.

Definition 7. An *adjunction* consists of a pair of functors

$$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$$

where F is the *left adjoint* and G is the *right adjoint*, together with natural transformations

$$\eta \colon 1_\mathbf{C} \Rightarrow GF \qquad \varepsilon \colon FG \Rightarrow 1_\mathbf{D}$$

called the *unit* and *counit*, respectively, satisfying the triangle identities:

$$\begin{tikzcd} & D_2 \arrow[dr, "c"] & \\ C_0 \arrow[ur, "\langle f_1\eta, \varepsilon f_0\rangle"] \arrow[rr, "1_F"'] & & D_1 \end{tikzcd} \qquad \begin{tikzcd} & C_2 \arrow[dr, "c"] & \\ D_0 \arrow[ur, "\langle \eta g_0, g_1\varepsilon\rangle"] \arrow[rr, "1_G"'] & & C_1 \end{tikzcd}$$

Alternatively, the triangle identities state that the following diagrams compose to give the identity natural transformations $1_F \colon F \Rightarrow F$ and $1_G \colon G \Rightarrow G$, respectively.

$$\begin{tikzcd} & \mathbf{D} \arrow[rr, "1_\mathbf{D}"] \arrow[dr, "G"] \arrow[r, phantom, "\Downarrow\varepsilon"] & & \mathbf{D} \\ \mathbf{C} \arrow[ur, "F"] \arrow[rr, "1_\mathbf{C}"'] \arrow[r, phantom, "\Downarrow\eta"] & & \mathbf{C} \arrow[ur, "F"'] & \end{tikzcd} \qquad \begin{tikzcd} \mathbf{D} \arrow[rr, "1_\mathbf{D}"] \arrow[dr, "G"'] \arrow[r, phantom, "\Downarrow\varepsilon"] & & \mathbf{D} \arrow[dr, "G"] & \\ & \mathbf{C} \arrow[ur, "F"] \arrow[rr, "1_\mathbf{C}"'] \arrow[r, phantom, "\Downarrow\eta"] & & \mathbf{C} \end{tikzcd}$$

2.2 Codiscrete categories

Given a set C, the *codiscrete category* $\mathrm{cd}(C)$ has set of objects C and set of morphisms given by the cartesian product,

$$\begin{tikzcd} C & C \times C \arrow[l, "l_1"'] \arrow[r, "r_1"] & C \end{tikzcd}$$

where the left and right projections are the domain and codomain maps, respectively. The identity map, often called the *diagonal*, is induced by the universal property of the product:

$$\begin{tikzcd} & C \arrow[dl, "1_C"'] \arrow[d, dashed, "\Delta"] \arrow[dr, "1_C"] & \\ C & C \times C \arrow[l, "l_1"] \arrow[r, "r_1"'] & C \end{tikzcd}$$

The set of composable morphisms is given by the product $C \times C \times C$ constructed via the pullback,

$$\begin{tikzcd} & & C \times C \times C \arrow[dl, "l_2"'] \arrow[dr, "r_2"] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & & \\ & C \times C \arrow[dl, "l_1"'] \arrow[dr, "r_1"] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & & C \times C \arrow[dl, "l_1"'] \arrow[dr, "r_1"] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & \\ C \arrow[dr, "!"'] & & C \arrow[dl, "!"] \arrow[dr, "!"'] & & C \arrow[dl, "!"] \\ & \{*\} & & \{*\} & \end{tikzcd}$$

while the composition map also induced by the universal property of the product via the following diagram:

$$\begin{tikzcd} C \times C \arrow[d, "l_1"'] & C \times C \times C \arrow[l, "l_2"'] \arrow[r, "r_2"] \arrow[d, dashed, "\langle l_1 l_2, r_1 r_2 \rangle"] & C \times C \arrow[d, "r_1"] \\ C & C \times C \arrow[l, "l_1"] \arrow[r, "r_1"'] & C \end{tikzcd}$$

That composition is unitial and associative follows directly from the universal property of the product, and we omit the details.

Given a function $f \colon C \to D$ we can induce a functor $\mathrm{cd}(f) \colon \mathrm{cd}(C) \to \mathrm{cd}(D)$ between the corresponding codiscrete categories induced via the following diagram:

$$\begin{tikzcd} C \arrow[d, "f"'] & C \times C \arrow[l, "l_1"'] \arrow[r, "r_1"] \arrow[d, dashed, "f \times f"] & C \arrow[d, "f"] \\ D & D \times D \arrow[l, "l_1"] \arrow[r, "r_1"'] & D \end{tikzcd}$$

We omit the details showing that the following diagrams commute:

$$\begin{tikzcd} C \arrow[r, "\Delta"] \arrow[d, "f"'] & C \times C \arrow[d, "f \times f"] & C \times C \times C \arrow[l, "\langle l_1 l_2, r_1 r_2 \rangle"'] \arrow[d, "f \times f \times f"] \\ D \arrow[r, "\Delta"'] & D \times D & D \times D \times D \arrow[l, "\langle l_1 l_2, r_1 r_2 \rangle"] \end{tikzcd}$$

Here we prefer the more suggestive shorthand $f \times f \times f = (f \times f) \times (f \times f)$.

2.3 Comma categories and arrow categories

Comma categories are a versatile construction in the 2-category **Cat** of small categories, functors, and natural transformations. Here we provide the abstract definition of a comma category together with its universal properties.

Definition 8. Given a cospan of categories and functors,

$$\mathbf{A} \xrightarrow{F} \mathbf{C} \xleftarrow{G} \mathbf{B}$$

their *comma category* $(F \downarrow G)$ is the category together with projection functors and a natural transformation,

$$\begin{tikzcd}
& (F \downarrow G) \arrow[dl, "L"'] \arrow[dr, "R"] & \\
\mathbf{A} \arrow[dr, "F"'] \arrow[rr, Rightarrow, "\alpha", shorten=20pt] & & \mathbf{B} \arrow[dl, "G"] \\
& \mathbf{C} &
\end{tikzcd}$$

satisfying the following universal properties:

- ◇ Given functors $L' \colon \mathbf{D} \to \mathbf{A}$ and $R' \colon \mathbf{D} \to \mathbf{B}$ and a natural transformation $\alpha' \colon FL' \Rightarrow GR'$, there exists a unique functor $H \colon \mathbf{D} \to (F \downarrow G)$ such that the following diagrams are equal:

$$\begin{tikzcd}
& \mathbf{D} \arrow[dl, "L'"'] \arrow[dr, "R'"] & \\
\mathbf{A} \arrow[dr, "F"'] \arrow[rr, Rightarrow, "\alpha'", shorten=20pt] & & \mathbf{B} \arrow[dl, "G"] \\
& \mathbf{C} &
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
& \mathbf{D} \arrow[d, dashed, "H"] \arrow[ddl, bend right, "L'"'] \arrow[ddr, bend left, "R'"] & \\
& (F \downarrow G) \arrow[dl, "L"'] \arrow[dr, "R"] & \\
\mathbf{A} \arrow[dr, "F"'] \arrow[rr, Rightarrow, "\alpha", shorten=20pt] & & \mathbf{B} \arrow[dl, "G"] \\
& \mathbf{C} &
\end{tikzcd}$$

- ◇ Given functors $H, H' \colon \mathbf{D} \to (F \downarrow G)$ and natural transformations $\beta \colon LH \Rightarrow LH'$ and $\gamma \colon RH \Rightarrow RH'$ such that the following whiskered composites are equal,

$$\begin{tikzcd}
& \mathbf{D} \arrow[dl, "H"'] \arrow[dd, "H'"] & \\
(F \downarrow G) \arrow[dd, "L"'] \arrow[dr, Rightarrow, "\beta", shorten=10pt] & & \\
& (F \downarrow G) \arrow[dl, "L"'] \arrow[dr, "R"] & \\
\mathbf{A} \arrow[dr, "F"'] \arrow[rr, Rightarrow, "\alpha", shorten=20pt] & & \mathbf{B} \arrow[dl, "G"] \\
& \mathbf{C} &
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
& \mathbf{D} \arrow[dr, "H'"] \arrow[dd, "H"'] & \\
& & (F \downarrow G) \arrow[dd, "R"] \\
& (F \downarrow G) \arrow[dl, "L"'] \arrow[dr, "R"] \arrow[ur, Rightarrow, "\gamma", shorten=10pt] & \\
\mathbf{A} \arrow[dr, "F"'] \arrow[rr, Rightarrow, "\alpha", shorten=20pt] & & \mathbf{B} \arrow[dl, "G"] \\
& \mathbf{C} &
\end{tikzcd}$$

  there exists a unique natural transformation $\theta \colon H \Rightarrow H'$ such that

$$\beta = L\theta \qquad \text{and} \qquad \gamma = R\theta.$$

Given a small category $\mathbf{C}$, we wish to construct a category whose objects are the morphisms of $\mathbf{C}$. The instance of the comma category for the identity cospan,

$$\mathbf{C} \xrightarrow{1_{\mathbf{C}}} \mathbf{C} \xleftarrow{1_{\mathbf{C}}} \mathbf{C}$$

is known as the *arrow category* for $\mathbf{C}$ and is constructed formally from the comma square:

$$\begin{tikzcd}
 & (\mathbf{C} \downarrow \mathbf{C}) \arrow[dl, "L"'] \arrow[dr, "R"] & \\
\mathbf{C} \arrow[dr, "1_{\mathbf{C}}"'] \arrow[rr, phantom, "\xRightarrow{\phi}"] & & \mathbf{C} \arrow[dl, "1_{\mathbf{C}}"] \\
 & \mathbf{C} &
\end{tikzcd}$$

We prefer to use the shorter notation $\Phi\mathbf{C} \coloneqq (\mathbf{C} \downarrow \mathbf{C})$ for the arrow category.

Alternatively, given a small category $\mathbf{C}$, the *arrow category* $\Phi\mathbf{C}$ has set of objects C_1 and set of morphisms given by the commutative squares, constructed via the pullback with domain and codomain maps depicted:

$$\begin{tikzcd}
 & & C_2 \times_{C_1} C_2 \arrow[dl, "\lambda"'] \arrow[dr, "\rho"] \arrow[d, phantom, "\lrcorner" rotate=-45] & & \\
 & C_2 \arrow[dl, "l_2"'] \arrow[dr, "c"'] & {} & C_2 \arrow[dl, "c"] \arrow[dr, "r_2"] & \\
C_1 & & C_1 & & C_1
\end{tikzcd}$$

Given a functor $F\colon \mathbf{C} \to \mathbf{D}$ between small categories, we can induce a functor between the corresponding arrow categories using the universal property of the comma category:

$$\begin{tikzcd}
 & \Phi\mathbf{C} \arrow[d, dashed, "\Phi F"] \arrow[ddl, bend right, "FL"'] \arrow[ddr, bend left, "FR"] & \\
 & \Phi\mathbf{D} \arrow[dl, "L"'] \arrow[dr, "R"] & \\
\mathbf{D} \arrow[dr, "1_{\mathbf{D}}"'] \arrow[rr, phantom, "\xRightarrow{\phi}"] & & \mathbf{D} \arrow[dl, "1_{\mathbf{D}}"] \\
 & \mathbf{D} &
\end{tikzcd}$$

We may also define the comma category via the following pullback diagram with the arrow category:

$$\begin{tikzcd}
 & & (F \downarrow G) \arrow[dl] \arrow[dr] \arrow[ddll, bend right, "L"'] \arrow[ddrr, bend left, "R"] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & & \\
 & (F \downarrow \mathbf{C}) \arrow[dl] \arrow[dr] \arrow[d, phantom, "\lrcorner" rotate=-45, very near start] & & (\mathbf{C} \downarrow G) \arrow[dl] \arrow[dr] \arrow[d, phantom, "\lrcorner" rotate=-45, very near start] & \\
\mathbf{A} \arrow[dr, "F"'] & {} & \Phi\mathbf{C} \arrow[dl, "L"'] \arrow[dr, "R"] & {} & \mathbf{B} \arrow[dl, "G"] \\
 & \mathbf{C} \arrow[dr, "1_{\mathbf{C}}"'] \arrow[rr, phantom, "\xRightarrow{\phi}"] & & \mathbf{C} \arrow[dl, "1_{\mathbf{C}}"] & \\
 & & \mathbf{C} & &
\end{tikzcd}$$

2.4 Double categories

The category **Cat** of small categories and functors is known to have pullbacks. It is possible categorify the definition of a small category, simply by replacing every instance of *set* with *category*, and every instance of *function* with *functor*.

Definition 9. A *double category* $\mathbb{D}$, or a *category internal to* **Cat**, consists of a category of objects $\mathbf{D}_0$ and a category of morphisms $\mathbf{D}_1$ together with functors

$$\begin{tikzcd}
\mathbf{D}_0 \arrow[r, "\eta"] & \mathbf{D}_1 \arrow[l, bend right, "R_1"'] \arrow[l, bend left, "L_1"] & \mathbf{D}_2 \arrow[l, "\mu"'] \arrow[l, bend right, "R_2"'] \arrow[l, bend left, "L_2"] & \mathbf{D}_3 \arrow[l, shift right, "R_3"'] \arrow[l, shift left, "L_3"]
\end{tikzcd}$$

where $\mathbf{D}_2$ and $\mathbf{D}_3$ are pullbacks, and the functors satisfy the relevant commutative diagrams.

A more effective characterisation of a double category emphasises the two different types of morphism present. For a double category $\mathbb{D}$, let D_{00} and D_{01} be the sets of objects and morphisms of the category $\mathbf{D}_0$, and likewise D_{10} and D_{11} for the sets of objects and morphisms of the category $\mathbf{D}_1$.

Definition 10. A *double category* $\mathbb{D}$ consists of sets D_{00}, D_{01}, D_{10}, and D_{11} such that

- There is a small category (D_{00}, D_{01}) of objects and *vertical morphisms*, together with vertical composition defined by the category $\mathbf{D}_0$.
- There is a small category (D_{00}, D_{10}) of objects and *horizontal morphisms*, together with horizontal composition defined by the internal structure to **Cat**.
- There is a small category (D_{10}, D_{11}) of horizontal morphisms and 2-*cells*, together with vertical composition of 2-cells defined by the category $\mathbf{D}_1$.

$$\begin{tikzcd}
X \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\alpha"] & Y \arrow[d] \\
X' \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\gamma"] & Y' \arrow[d] \\
X'' \arrow[r] & Y''
\end{tikzcd}
=
\begin{tikzcd}
X \arrow[r] \arrow[dd] \arrow[dd, phantom, shift left=5ex, "\Downarrow\gamma\bullet\alpha"] & Y \arrow[dd] \\
& \\
X'' \arrow[r] & Y''
\end{tikzcd}$$

- There is a small category (D_{01}, D_{11}) of vertical morphisms and 2-cells, together with horizontal composition of 2-cells defined by the internal structure to **Cat**.

$$\begin{tikzcd}
X \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\alpha"] & Y \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\beta"] & Z \arrow[d] \\
X' \arrow[r] & Y' \arrow[r] & Z'
\end{tikzcd}
=
\begin{tikzcd}
X \arrow[rr] \arrow[d] \arrow[drr, phantom, "\Downarrow\beta*\alpha"] & & Z \arrow[d] \\
X' \arrow[rr] & & Z'
\end{tikzcd}$$

satisfying the following *interchange law* between vertical and horizontal composition of 2-cells:

$$\begin{tikzcd}
X \arrow[rr] \arrow[d] \arrow[drr, phantom, "\Downarrow\beta*\alpha"] & & Z \arrow[d] \\
X' \arrow[rr] \arrow[d] \arrow[drr, phantom, "\Downarrow\delta*\gamma"] & & Z' \arrow[d] \\
X'' \arrow[rr] & & Z''
\end{tikzcd}
=
\begin{tikzcd}
X \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\alpha"] & Y \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\beta"] & Z \arrow[d] \\
X' \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\gamma"] & Y' \arrow[r] \arrow[d] \arrow[dr, phantom, "\Downarrow\delta"] & Z' \arrow[d] \\
X'' \arrow[r] & Y'' \arrow[r] & Z''
\end{tikzcd}
=
\begin{tikzcd}
X \arrow[r] \arrow[dd] \arrow[dd, phantom, shift left=5ex, "\Downarrow\gamma\bullet\alpha"] & Y \arrow[r] \arrow[dd] \arrow[dd, phantom, shift left=5ex, "\Downarrow\delta\bullet\beta"] & Z \arrow[dd] \\
& & \\
X'' \arrow[r] & Y'' \arrow[r] & Z''
\end{tikzcd}$$

Definition 11. Let $\mathbb{C}$ and $\mathbb{D}$ be double categories. A *double functor* $\mathcal{F}\colon \mathbb{C} \to \mathbb{D}$, or simply a functor internal to **Cat**, consists of a pair of functors

$$F_0\colon \mathbf{C}_0 \longrightarrow \mathbf{D}_0 \qquad F_1\colon \mathbf{C}_1 \longrightarrow \mathbf{D}_1$$

satisfying the following commutative diagrams:

$$\begin{tikzcd}
\mathbf{C}_0 \arrow[d, "F_0"'] & \mathbf{C}_1 \arrow[l, "L_1"'] \arrow[r, "R_1"] \arrow[d, "F_1"] & \mathbf{C}_0 \arrow[d, "F_0"] \\
\mathbf{D}_0 & \mathbf{D}_1 \arrow[l, "L_1"] \arrow[r, "R_1"'] & \mathbf{D}_0
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{C}_0 \arrow[r, "\eta"] \arrow[d, "F_0"'] & \mathbf{C}_1 \arrow[d, "F_1"] & \mathbf{C}_2 \arrow[l, "\mu"'] \arrow[d, "F_1 \times F_1"] \\
\mathbf{D}_0 \arrow[r, "\eta"'] & \mathbf{D}_1 & \mathbf{D}_2 \arrow[l, "\mu"]
\end{tikzcd}$$

The *identity functor* $1_{\mathbb{D}}\colon \mathbb{D} \to \mathbb{D}$ for a double category $\mathbb{D}$ consists of the functions:

$$1\colon \mathbf{D}_0 \to \mathbf{D}_0 \qquad 1\colon \mathbf{D}_1 \to \mathbf{D}_1$$

Let **Dbl** be the category whose objects are double categories and whose morphisms are double functors. When working with double categories, we will often find it more convenient and intuitive to use explicit squares of objects, morphisms, and 2-cells rather than keeping within a completely internal framework.

2.5 Double category of squares

Given a category $\mathbf{D}$, the *double category of squares* $\mathrm{sq}(\mathbf{D})$ has category of objects $\mathbf{D}$ and category of morphisms given by the arrow category,

$$\mathbf{D} \xleftarrow{L_1} \Phi\mathbf{D} \xrightarrow{R_1} \mathbf{D}$$

where the left and right projections are the domain and codomain maps, respectively.

The identity map is induced by the universal property of the comma category:

$$\begin{tikzcd}
& \mathbf{D} \arrow[d, dashed, "\Delta"] \arrow[ddl, bend right, "1_{\mathbf{D}}"'] \arrow[ddr, bend left, "1_{\mathbf{D}}"] & \\
& \Phi\mathbf{D} \arrow[dl, "L_1"'] \arrow[dr, "R_1"] & \\
\mathbf{D} \arrow[rr, phantom, "\xRightarrow{\phi}"] \arrow[dr, "1_{\mathbf{D}}"'] & & \mathbf{D} \arrow[dl, "1_{\mathbf{D}}"] \\
& \mathbf{D} &
\end{tikzcd}$$

Recall the identity map Δ provides the horizontal identity, while the internal structure of the arrow category $\Phi\mathbf{D}$ provides the vertical identity, depicted by the following squares, respectively:

$$\begin{tikzcd}
X \arrow[r, "1_X"] \arrow[d, "f"'] & X \arrow[d, "f"] \\
X' \arrow[r, "1_{X'}"'] & X'
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "\phi"] \arrow[d, "1_X"'] & Y \arrow[d, "1_Y"] \\
X \arrow[r, "\phi"'] & Y
\end{tikzcd}$$

The category of composable morphisms is given by the pullback

$$\begin{tikzcd}
 & & \Phi\mathbf{D} \times_{\mathbf{D}} \Phi\mathbf{D} \arrow[dl, "L_2"'] \arrow[dr, "R_2"] \arrow[dd, phantom, very near start, "\lrcorner" rotate=-45] & & \\
 & \Phi\mathbf{D} \arrow[dl, "L_1"'] \arrow[dr, "R_1"] & & \Phi\mathbf{D} \arrow[dl, "L_1"'] \arrow[dr, "R_1"] & \\
\mathbf{D} \arrow[dr, "1_{\mathbf{D}}"'] \arrow[rr, phantom, "\overset{\phi}{\Longrightarrow}"] & & \mathbf{D} \arrow[dl, "1_{\mathbf{D}}"] \arrow[dr, "1_{\mathbf{D}}"'] \arrow[rr, phantom, "\overset{\phi}{\Longrightarrow}"] & & \mathbf{D} \arrow[dl, "1_{\mathbf{D}}"] \\
 & \mathbf{D} & & \mathbf{D} &
\end{tikzcd}$$

while the composition map also induced by the universal property of the comma category via the following diagram:

$$\begin{tikzcd}
 & \Phi\mathbf{D} \times_{\mathbf{D}} \Phi\mathbf{D} \arrow[d, dashed, "\mu"] \arrow[ddl, bend right, "L_1 L_2"'] \arrow[ddr, bend left, "R_1 R_2"] & \\
 & \Phi\mathbf{D} \arrow[dl, "L_1"'] \arrow[dr, "R_1"] & \\
\mathbf{D} \arrow[dr, "1_{\mathbf{D}}"'] \arrow[rr, phantom, "\overset{\phi}{\Longrightarrow}"] & & \mathbf{D} \arrow[dl, "1_{\mathbf{D}}"] \\
 & \mathbf{D} &
\end{tikzcd}$$

Again recall that the composition map μ defines the horizontal composition,

$$\begin{tikzcd}
X \arrow[r, "\phi"'] \arrow[d, "f"'] \arrow[rr, bend left, "\psi\phi"] & Y \arrow[r, "\psi"'] \arrow[d, "g"] & Z \arrow[d, "h"] \\
X' \arrow[r, "\phi'"] \arrow[rr, bend right, "\psi'\phi'"'] & Y' \arrow[r, "\psi'"] & Z'
\end{tikzcd}$$

while the internal structure of the arrow category $\Phi\mathbf{D}$ defines the vertical composition:

$$\begin{tikzcd}
X \arrow[r, "\phi"] \arrow[d, "f"] \arrow[dd, bend right, "hf"'] & Y \arrow[d, "g"'] \arrow[dd, bend left, "kg"] \\
X' \arrow[r, "\phi'"] \arrow[d, "h"] & Y' \arrow[d, "k"'] \\
X'' \arrow[r, "\phi''"'] & Y''
\end{tikzcd}$$

Given a functor $F\colon \mathbf{C} \to \mathbf{D}$ we can induce a double functor $\mathrm{sq}(F)\colon\ \mathrm{sq}(\mathbf{C}) \to \mathrm{sq}(\mathbf{D})$ between the corresponding double categories of squares:

$$\begin{tikzcd}
\mathbf{C} \arrow[d, "F"'] & \Phi\mathbf{C} \arrow[l, "L_2"'] \arrow[r, "R_2"] \arrow[d, dashed, "\Phi F"] & \mathbf{C} \arrow[d, "F"] \\
\mathbf{D} & \Phi\mathbf{D} \arrow[l, "L_1"] \arrow[r, "R_1"'] & \mathbf{D}
\end{tikzcd}$$

We omit the details showing that the following diagrams commute:

$$\begin{tikzcd}
\mathbf{C} \arrow[r, "\Delta"] \arrow[d, "F"'] & \Phi\mathbf{C} \arrow[d, "\Phi F"] & \Phi\mathbf{C} \times_{\mathbf{C}} \Phi\mathbf{C} \arrow[l, "\mu"'] \arrow[d, "\Phi F \times \Phi F"] \\
\mathbf{D} \arrow[r, "\Delta"'] & \Phi\mathbf{D} & \Phi\mathbf{D} \times_{\mathbf{D}} \Phi\mathbf{D} \arrow[l, "\mu"]
\end{tikzcd}$$

Given that a double category is internal to the 2-category **Cat**, we are also able to remark upon adjunctions between functors which provide the structure. In particular, for the double category of squares, the domain, identity, and codomain maps form an adjoint triple, a fact which will be useful later. A reference for this result may be found in [5].

Lemma 1. *The diagonal is left-adjoint right-inverse to the left projection $\Delta \dashv L_1$ with counit $\iota_{L_1} \colon \Delta L_1 \Rightarrow 1_{\Phi\mathbf{D}}$ defined by the whiskered natural transformations:*

$$L_1\iota_L = 1_{L_1} \colon L_1 \Rightarrow L_1 \qquad R_1\iota_L = \phi \colon L_1 \Rightarrow R_1.$$

Dually, the diagonal is right-adjoint right-inverse to the right projection $R_1 \dashv \Delta$ with unit $\iota_{R_1} \colon 1_{\Phi\mathbf{D}} \Rightarrow \Delta R_1$ defined by the whiskered natural transformations:

$$L_1\iota_{R_1} = \phi \colon L_1 \Rightarrow R_1 \qquad R_1\iota_{R_1} = 1_{R_1} \colon R_1 \Rightarrow R_1.$$

Given an object $\phi \colon X \to Y$, the component of the counit ι_L at ϕ and the component of the unit ι_R at ϕ, are given by the the following commutative squares, respectively:

$$\begin{tikzcd}
X \arrow[r, "1_X"] \arrow[d, "1_X"'] & X \arrow[d, "\phi"] \\
X \arrow[r, "\phi"'] & Y
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
X \arrow[r, "\phi"] \arrow[d, "\phi"'] & Y \arrow[d, "1_Y"] \\
Y \arrow[r, "1_Y"'] & Y
\end{tikzcd}$$

3

Internal characterisation of set-based lenses

Consider a pair of sets S and V called the *source* and *view*, respectively, whose elements are called *states*. The morphisms of the corresponding codiscrete categories are often called *updates*.

Definition 12. A *(set-based) lens* $\Lambda \colon S \rightleftharpoons V$ is a quadruple $\Lambda = (S, V, g, p)$ consisting of a function,

$$g \colon S \longrightarrow V \tag{Get}$$

together with a function,

$$p \colon S \times V \longrightarrow S \tag{Put}$$

satisfying the following commutative diagrams called the *lens laws*:

$$\begin{tikzcd} & S \times V \arrow[dl, "p"'] \arrow[dr, "r_1"] & \\ S \arrow[rr, "g"'] & & V \end{tikzcd} \qquad \begin{tikzcd} & S \times V \arrow[dr, "p"] & \\ S \arrow[ur, "\langle 1_S, g\rangle"] \arrow[rr, "1_S"'] & & S \end{tikzcd} \qquad \begin{tikzcd} S \times V \times V \arrow[r, "p \times 1_V"] \arrow[d, "l_1 \times 1_V"'] & S \times V \arrow[d, "p"] \\ S \times V \arrow[r, "p"'] & S \end{tikzcd}$$

In order, these diagrams are known as Put-Get, Get-Put, and Put-Put.

The definition of a lens was first stated in [2] and was later stated in the above diagrammatic form as algebras for a monad in [12].

The goal of this chapter is to recast the definition of a lens within the context of internal category theory. Working internal to **Set**, we characterise a lens as a small category $\mathbf{\Lambda} = (S, S \times V)$, and show this category takes part in a commutative triangle of functors. We show these commutative triangles compose, and conclude with a definition of the category **Lens** whose objects are sets and whose morphisms are lenses.

3.1 The Get function

The forward direction of a lens $\Lambda\colon S \rightleftharpoons V$ is simply a function $g\colon S \to V$ called the Get function. The Get function induces a canonical functor $\mathrm{cd}(g)\colon \mathrm{cd}(S) \to \mathrm{cd}(V)$ between codiscrete categories which, by virtue of being a functor, consists of a pair of functions,

$$g\colon S \longrightarrow V \qquad\qquad g \times g\colon S \times S \longrightarrow V \times V \tag{Get}$$

satisfying the following commutative diagrams:

$$\begin{tikzcd}
S \arrow[d, "g"'] & S \times S \arrow[l, "l_1"'] \arrow[r, "r_1"] \arrow[d, "g\times g"] & S \arrow[d, "g"] \\
V & V \times V \arrow[l, "l_1"] \arrow[r, "r_1"'] & V
\end{tikzcd}
\qquad
\begin{tikzcd}
S \arrow[r, "\Delta"] \arrow[d, "g"'] & S \times S \arrow[d, "g\times g"] & S \times S \times S \arrow[l, "\langle l_1 l_2, r_1 r_2\rangle"'] \arrow[d, "g\times g\times g"] \\
V \arrow[r, "\Delta"'] & V \times V & V \times V \times V \arrow[l, "\langle l_1 l_2, r_1 r_2\rangle"]
\end{tikzcd}$$

The *set of view updates* for the Get function is given by the left-hand pullback,

$$\begin{tikzcd}
S \times V \arrow[r, dashed, "g\times 1_V"] \arrow[d, "l_1"'] \arrow[rr, bend left, "r_1"] \arrow[dr, phantom, "\lrcorner", very near start] & V \times V \arrow[r, "r_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \arrow[d, "!"] \\
S \arrow[r, "g"'] & V \arrow[r, "!"'] & \{*\}
\end{tikzcd} \tag{3.1}$$

which is induced from the right-hand and outer pullbacks over the singleton set by the *pullback pasting lemma*; of course, these pullbacks are really just cartesian products.

Furthermore we can use the universal property of $S \times V$ to induce a canonical factorisation of the function $g \times g\colon S \times S \to V \times V$ depicted in the following diagram:

$$\begin{tikzcd}
S \times S \arrow[r, dashed, "1_S\times g"'] \arrow[d, "l_1"'] \arrow[rr, bend left, "g\times g"] & S \times V \arrow[r, "g\times 1_V"'] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \times V \arrow[d, "l_1"] \\
S \arrow[r, "1_S"'] & S \arrow[r, "g"'] & V
\end{tikzcd}$$

The universal property of the set of view updates may be used to induce two other canonical functions, which will be important in constructing a small category $\mathbf{\Lambda}$ whose set of morphisms is $S \times V$, and whose sets of objects is S.

Firstly, we define a candidate for the identity map via the following diagram:

$$\begin{tikzcd}
 & S \arrow[dl, "1_S"'] \arrow[d, dashed, "\langle 1_S, g\rangle"] \arrow[dr, "g"] & \\
S & S \times V \arrow[l, "l_1"] \arrow[r, "r_1"'] & V
\end{tikzcd} \tag{3.2}$$

Next we extend the diagram (3.1) to include an additional pullback:

$$
\begin{tikzcd}
S \times V \times V \arrow[r, "g\times 1_V \times 1_V"] \arrow[d, "l_2"'] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[rrr, bend left, "r_2"] & V \times V \times V \arrow[r, "r_2"] \arrow[d, "l_2"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \times V \arrow[r, "r_1"] \arrow[d, "l_1"] \arrow[dr, phantom, "\lrcorner", very near start] & V \arrow[d, "!"] \\
S \times V \arrow[r, "g\times 1_V"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \times V \arrow[r, "r_1"] \arrow[d, "l_1"] \arrow[dr, phantom, "\lrcorner", very near start] & V \arrow[r, "!"] \arrow[d, "!"] & \{*\} \\
S \arrow[r, "g"'] & V \arrow[r, "!"'] & V &
\end{tikzcd}
\tag{3.3}
$$

We define a candidate for the composition map via the following diagram:

$$
\begin{tikzcd}
S \times V \arrow[d, "l_1"'] & S \times V \times V \arrow[l, "l_2"'] \arrow[r, "r_2"] \arrow[d, dashed, "l_1\times 1_V"] & V \arrow[d, "1_V"] \\
S & S \times V \arrow[l, "l_1"] \arrow[r, "r_1"'] & V
\end{tikzcd}
\tag{3.4}
$$

Altogether the diagrams (3.1), (3.2), (3.4) suggest the definition of a small category $\mathbf{\Lambda}$ whose set of objects is $\Lambda_0 \coloneqq S$ and whose set of morphisms is $\Lambda_1 \coloneqq S \times V$. We already have suitable candidates for the domain map $l_1 \colon S \times V \to S$, the identity map $\langle 1_S, g\rangle \colon S \to S \times V$, and the composition map $l_1 \times 1_V \colon S \times V \times V \to S \times V$. A suitable candidate for the codomain map will be obtained from the Put function in the next section.

3.2 The Put function

The backwards direction of a lens $\Lambda \colon S \rightleftharpoons V$ is given by a function,

$$p \colon S \times V \longrightarrow S \tag{Put}$$

called the Put function, satisfying the following commutative diagrams:

$$
\begin{tikzcd}
& S \times V \arrow[dl, "p"'] \arrow[dr, "r_1"] & \\
S \arrow[rr, "g"'] & & V
\end{tikzcd}
\qquad
\begin{tikzcd}
& S \times V \arrow[dr, "p"] & \\
S \arrow[ur, "\langle 1_S, g\rangle"] \arrow[rr, "1_S"'] & & S
\end{tikzcd}
\qquad
\begin{tikzcd}
S \times V \times V \arrow[r, "p\times 1_V"] \arrow[d, "l_1\times 1_V"'] & S \times V \arrow[d, "p"] \\
S \times V \arrow[r, "p"'] & S
\end{tikzcd}
\tag{3.5}
$$

Recall the above diagrams are known as the *lens laws* from Definition 12 and the function $p \times 1_V$ present in the Put-Put law may be defined using the universal property of the product via the diagram,

$$
\begin{tikzcd}
S \times V \times V \arrow[r, dashed, "p\times 1_V"] \arrow[d, "l_2"'] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[rr, bend left, "r_2"] & S \times V \arrow[r, "r_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \arrow[d, "!"] \\
S \times V \arrow[r, "p"'] & S \arrow[r, "!"'] & \{*\}
\end{tikzcd}
\tag{3.6}
$$

where the right-hand square and the outer rectangle are pullbacks over the singleton which induce the left-hand pullback square by the *pullback pasting lemma*.

Theorem 2. *If the quadruple $\Lambda = (S, V, g, p)$ forms a lens, then the pair $\mathbf{\Lambda} = (\Lambda_0, \Lambda_1) = (S, S \times V)$ forms a small category.*

Proof. Given a lens $\Lambda\colon S \rightleftharpoons V$ consider the functions,

$$\begin{tikzcd}
\Lambda_0 \arrow[r, "\langle 1_S, g\rangle" description] & \Lambda_1 \arrow[l, bend right, "p"'] \arrow[l, bend left, "l_1"] & \Lambda_2 \arrow[l, "l_1 \times 1_V" description] \arrow[l, bend right, "p \times 1_V"'] \arrow[l, bend left, "l_2"]
\end{tikzcd}$$

denoting the candidates for the domain, codomain, identity, composition, and projection maps for a prospective small category $\mathbf{\Lambda}$ with set of objects $\Lambda_0 = S$, set of morphisms $\Lambda_1 = S \times V$, and set of composable morphisms $\Lambda_2 = S \times V \times V$.

Given the function $\langle 1_S, g\rangle\colon S \to S \times V$ defined in (3.2) we have,

$$\begin{tikzcd}
 & S \arrow[dl, "1_S"'] \arrow[d, "\langle 1_S, g\rangle"] \arrow[dr, "1_S"] & \\
S & S \times V \arrow[l, "l_1"] \arrow[r, "p"'] & S
\end{tikzcd}$$

where the left-hand triangle commutes by construction and the right-hand triangle commutes by the Get-Put law. Therefore the function $\langle 1_S, g\rangle\colon S \to S \times V$ satisfies the diagrams for the identity map of a small category.

Next note from (3.6) that $\Lambda_2 = S \times V \times V$ is the pullback of the candidates for the domain and codomain maps, and therefore is well-defined as the set of composable morphisms.

Given the function $l_1 \times 1_V\colon S \times V \times V \to S \times V$ defined in (3.4) we have,

$$\begin{tikzcd}
S \times V \arrow[d, "l_1"'] & S \times V \times V \arrow[l, "l_2"'] \arrow[r, "p \times 1_V"] \arrow[d, "l_1 \times 1_V"] & S \times V \arrow[d, "p"] \\
\Lambda_0 & S \times V \arrow[l, "l_1"] \arrow[r, "p"'] & S
\end{tikzcd}$$

where the left-hand square commutes by construction and the right-hand square commutes by the Put-Put law. Therefore the function $l_1 \times 1_V\colon S \times V \times V \to S \times V$ satisfies the diagrams for the composition map of a small category.

In order to show the right-unitality law holds, we construct the universal function into the set of composable morphisms:

$$\begin{tikzcd}
 & S \times V \arrow[ddl, bend right, "1_{S \times V}"'] \arrow[d, dashed] \arrow[r, "p"] & S \arrow[dd, "\langle 1_S, g\rangle"] \\
 & S \times V \times V \arrow[dl, "l_2"'] \arrow[dr, "p \times 1_V"] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & \\
S \times V \arrow[dr, "p"'] & & S \times V \arrow[dl, "l_1"] \\
 & S &
\end{tikzcd}$$

This map corresponds to the universal function $\langle 1, ir_1\rangle\colon C_1 \to C_2$ in Definition 1. Composing with the composition map $l_1 \times 1_V\colon S \times V \times V \to S \times V$ and using the

Put-Get law we have the following diagram:

$$\begin{tikzcd}
 & S \times V \arrow[r, "p"] \arrow[dl, "1_{S\times V}"'] \arrow[d, dashed] & S \arrow[d, "\langle 1_S, g\rangle"] \\
S \times V \arrow[d, "l_1"'] & S \times V \times V \arrow[l, "l_2"] \arrow[r, "p\times 1_V"] \arrow[d, dashed, "l_1 \times 1_V"] & S \times V \arrow[d, "r_1 = gp"] \\
S & S \times V \arrow[l, "l_1"] \arrow[r, "r_1 = gp"'] & V
\end{tikzcd}$$

Now using the Put-Get law we have $r_1 \circ \langle 1_S, p\rangle \circ p = r_1 \colon S \times V \to V$, and by the universal property of the product, the composite of the dashed functions is the identity $1_{S\times V} \colon S \times V \to S \times V$. Therefore the right-unitality law holds.

The proof of the left-unitality law and associativity follow in a similar way: first construct the relevant universal function, compose and simplify using the lens laws, then invoke the universal property of the product. We omit the routine verification of these details. □

Definition 13. Given a lens $\Lambda \colon S \rightleftharpoons V$, its *category of view updates* $\mathbf{\Lambda}$ is given by the following sets and functions:

$$\begin{tikzcd}
\Lambda_0 \arrow[r, "\langle 1_S, g\rangle" description] & \Lambda_1 \arrow[l, bend right, "p"'] \arrow[l, bend left, "l_1"] & \Lambda_2 \arrow[l, "l_1 \times 1_V" description] \arrow[l, bend right, "p\times 1_V"'] \arrow[l, bend left, "l_2"]
\end{tikzcd}$$

3.3 A lens as a commuting triangle of functors

Consider a lens $\Lambda \colon S \rightleftharpoons V$ together with the corresponding view update category $\mathbf{\Lambda} = (S, S \times V)$. We now introduce two canonical functors to the codiscrete categories induced by the source and view, which form a commuting triangle with the functor induced by the Get function.

Using the Get-Put and Put-Put laws, there exists a functor $K \colon \mathbf{\Lambda} \to \mathrm{cd}(S)$ consisting of a pair of functions,

$$1_S \colon S \longrightarrow S \qquad\qquad \langle l_1, p\rangle \colon S \times V \longrightarrow S \times S$$

satisfying the following commutative diagrams:

$$\begin{tikzcd}
S \arrow[d, "1_S"'] & S \times V \arrow[l, "l_1"'] \arrow[r, "p"] \arrow[d, "\langle l_1, p\rangle"] & S \arrow[d, "1_S"] \\
S & S \times S \arrow[l, "l_1"] \arrow[r, "r_1"'] & S
\end{tikzcd}
\qquad
\begin{tikzcd}
S \arrow[r, "\langle 1_S, g\rangle"] \arrow[d, "1_S"'] & S \times V \arrow[d, "\langle l_1, p\rangle"] & S \times V \times V \arrow[l, "l_1 \times 1_V"'] \arrow[d, "\langle l_1, p\rangle \times p"] \\
S \arrow[r, "\Delta"'] & S \times S & S \times S \times S \arrow[l, "\langle l_1 l_2, r_1 r_2\rangle"]
\end{tikzcd}$$

Using the Put-Get law, there exists a functor $Q \colon \mathbf{\Lambda} \to \mathrm{cd}(V)$ consisting of a pair of functions,

$$g \colon S \longrightarrow V \qquad\qquad g \times 1_V \colon S \times V \longrightarrow V \times V$$

satisfying the following commutative diagrams:

$$\begin{tikzcd}
S \arrow[d, "g"'] & S \times V \arrow[l, "l_1"'] \arrow[r, "p"] \arrow[d, "g\times 1_V"'] \arrow[dr, dotted, "r_1"] & S \arrow[d, "g"] \\
V & V \times V \arrow[l, "l_1"] \arrow[r, "r_1"'] & V
\end{tikzcd}
\qquad
\begin{tikzcd}
S \arrow[r, "\langle 1_S, g\rangle"] \arrow[d, "g"'] & S \times V \arrow[d, "g\times 1_V"] & S \times V \times V \arrow[l, "l_1 \times 1_V"'] \arrow[d, "g \times 1_V \times 1_V"] \\
V \arrow[r, "\Delta"'] & V \times V & V \times V \times V \arrow[l, "\langle l_1 l_2, r_1 r_2\rangle"]
\end{tikzcd}$$

Furthermore, from the Put-Get law we can see that $\mathrm{cd}(g)K = Q\colon \mathbf{\Lambda} \to \mathbf{V}$, which may be expanded into the following commutative triangle of small categories and functors:

$$\begin{tikzcd}
& & S \times V \arrow[d, shift right, "p"'] \arrow[d, shift left, "l_1"] \arrow[dddll, "\langle l_1, p\rangle"'] \arrow[dddrr, "g \times 1_V"] & & \\
& & S \arrow[dl, "1_S"'] \arrow[dr, "g"] & & \\
& S \arrow[rr, "g"'] & & V & \\
S \times S \arrow[ur, shift left, "l_1"] \arrow[ur, shift right, "r_1"'] \arrow[rrrr, "g \times g"'] & & & & V \times V \arrow[ul, shift right, "r_1"'] \arrow[ul, shift left, "l_1"]
\end{tikzcd} \tag{3.7}$$

Corollary 3. *The quadruple $\Lambda = (S, V, g, p)$ forms a lens if and only if the diagram* (3.7) *forms a commuting triangle of small categories and functors.*

It is interesting that the construction of a lens which is based entirely of sets and functions can be naturally stated as a commuting diagram in **Cat**. One of the primary benefits of this formulation is the natural definition for composition of lenses as morphisms between sets.

3.4 The category Lens

We wish to construct a category **Lens** whose objects are sets and whose morphisms are set-based lenses. While the definition is well-known, it is difficult to find a reference in the literature for the composition of lenses except for the recent preprint [13]. Here we motivate the composition of lenses from first principles, and show it coincides with the pullback of their representation as a commuting triangle of functors.

Given a pair of lenses $\Lambda\colon S \rightleftharpoons V$ and $\Lambda'\colon V \rightleftharpoons U$, we define the Get function of the composite lens $\Lambda' \circ \Lambda\colon S \rightleftharpoons U$ to be the composite function $g' \circ g\colon S \to U$. Mirroring the construction in (3.1), the set of view updates for the composite Get function is given the pullback:

$$\begin{tikzcd}
S \times U \arrow[r, dashed, "g'g \times 1_V"] \arrow[d, "l_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] \arrow[rr, bend left, "r_1"] & U \times U \arrow[r, "r_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & U \arrow[d, "!"] \\
S \arrow[r, "g'g"'] & U \arrow[r, "!"'] & \{*\}
\end{tikzcd} \tag{3.8}$$

To define the Put function for the composite lenses, we first consider the decomposition of the pullback (3.8) as the following:

$$\begin{tikzcd}
S \times U \arrow[r, dashed, "g \times 1_U"] \arrow[d, "l_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] \arrow[rr, bend left, "g'g \times 1_U"] & V \times U \arrow[r, "g' \times 1_U"] \arrow[d, "l_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & U \times U \arrow[d, "l_1"] \\
S \arrow[r, "g"'] & V \arrow[r, "g'"'] & U
\end{tikzcd}$$

However this pullback may be decomposed even further to yield the following commutative diagram:

$$\begin{tikzcd}
S \times U \arrow[r, dashed, "g\times 1_U"] \arrow[d, dashed, "\langle l_1, p'(g\times 1_U)\rangle"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \times U \arrow[r, "g'\times 1_U"] \arrow[d, "\langle l_1, p'\rangle"] \arrow[dd, bend right, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & U \times U \arrow[dd, "l_1"] \\
S \times V \arrow[r, "g\times 1_V"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \times V \arrow[d, "l_1"] & \\
S \arrow[r, "g"'] & V \arrow[r, "g'"'] & U
\end{tikzcd}$$

Thus we may define the Put function for the composite lens to be the function:

$$\begin{tikzcd}
S \times U \arrow[r, "\langle l_1, p'(g\times 1_U)\rangle"'] \arrow[rr, bend left, "p\langle l_1, p'(g\times 1_U)\rangle"] & S \times V \arrow[r, "p"'] & S
\end{tikzcd}$$

We omit the details showing that this Put function satisfies the three lens laws.

Definition 14. Given a pair of lenses $\Lambda = (S, V, g, p)$ and $\Lambda' = (V, U, g', p')$, their *composite* lens $\Lambda' \circ \Lambda \colon S \rightleftharpoons U$ consists of the Get function,

$$g' \circ g \colon S \longrightarrow U$$

together with the Put function:

$$p\langle l_1, p'(g \times 1_U)\rangle \colon S \times U \longrightarrow S.$$

The identity lens is given by the quadruple $1_S = (S, S, 1_S, r_1)$.

Now consider a pair of lenses $\Lambda \colon S \rightleftharpoons V$ and $\Lambda' \colon V \rightleftharpoons U$ and their representation as commuting triangles of functors:

$$\begin{tikzcd}
& \mathbf{\Lambda} \arrow[dl, "K"'] \arrow[dr, "Q"] & & \mathbf{\Lambda'} \arrow[dl, "K'"'] \arrow[dr, "Q'"] & \\
\mathrm{cd}(S) \arrow[rr, "\mathrm{cd}(g)"'] & & \mathrm{cd}(V) \arrow[rr, "\mathrm{cd}(g')"'] & & \mathrm{cd}(U)
\end{tikzcd} \tag{3.9}$$

Given the composite lens $\Lambda' \circ \Lambda \colon S \rightleftharpoons U$, by Theorem 2 there is a small category of view updates $\mathbf{\Lambda'} \circ \mathbf{\Lambda}$ defined by the sets,

$$(\Lambda' \circ \Lambda)_0 \coloneqq S \qquad\qquad (\Lambda' \circ \Lambda)_1 \coloneqq S \times U$$

and by Corollary 3.7 we have commutative diagrams for the sets of objects and the sets of morphisms, respectively:

$$\begin{tikzcd}
& & S \arrow[dl, "1_S"'] \arrow[dr, "g"] \arrow[dd, phantom, "\vee", very near start] & & \\
& S \arrow[dl, "1_S"'] \arrow[dr, "g"] & & V \arrow[dl, "1_V"'] \arrow[dr, "g'"] & \\
S \arrow[rr, "g"'] \arrow[rrrr, bend right, "g'\circ g"'] & & V \arrow[rr, "g'"'] & & U
\end{tikzcd}
\qquad
\begin{tikzcd}
& & S \times U \arrow[dl, "\langle l_1, p'(g\times 1_U)\rangle"'] \arrow[dr, "g\times 1_U"] \arrow[dd, phantom, "\vee", very near start] & & \\
& S \times V \arrow[dl, "\langle l_1, p\rangle"'] \arrow[dr, "g\times 1_V"] & & V \times U \arrow[dl, "\langle l_1, p'\rangle"'] \arrow[dr, "g'\times 1_U"] & \\
S \times S \arrow[rr, "g\times g"'] & & V \times V \arrow[rr, "g'\times g'"'] & & U \times U
\end{tikzcd}$$

These combine to yield the following commutative diagram of small categories and functors:

$$
\begin{tikzcd}
 & & \mathbf{\Lambda}' \circ \mathbf{\Lambda} \arrow[dl, "\widehat{K'}"'] \arrow[dr, "\widehat{Q}"] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & & \\
 & \mathbf{\Lambda} \arrow[dl, "K"'] \arrow[dr, "Q"] & & \mathbf{\Lambda}' \arrow[dl, "K'"'] \arrow[dr, "Q'"] & \\
\mathrm{cd}(S) \arrow[rr, "\mathrm{cd}(g)"'] & & \mathrm{cd}(V) \arrow[rr, "\mathrm{cd}(g')"'] & & \mathrm{cd}(U)
\end{tikzcd}
\tag{3.10}
$$

Thus the composition of a pair of lenses (3.9) given by Definition 14 may be characterised by the specific choice of pullback $\mathbf{\Lambda}' \circ \mathbf{\Lambda}$ to yield a commuting triangle of functors (3.10).

Definition 15. Let **Lens** be the *category of lenses* whose objects are sets, whose morphisms are lenses, and whose composition is given by Definition 14.

Remark. The category **Lens** is well-defined: the composition of lenses is unital and associative, as it arises from the composition of the underlying Get functions, which is unital and associative.

4

Internal characterisation of c-lenses

Consider a pair of small categories $\mathbf{S}$ and $\mathbf{V}$ called the *source category* and *view category*, respectively. The objects of both categories are called *states* while their morphisms are usually called *updates*.

Definition 16. A *c-lens* $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ is a quadruple $\Lambda = (\mathbf{S}, \mathbf{V}, G, P_1)$ consisting of a functor,

$$G\colon \mathbf{S} \longrightarrow \mathbf{V} \tag{Get}$$

together with a functor,

$$P_1\colon (G \downarrow \mathbf{V}) \longrightarrow \mathbf{S} \tag{Put}$$

satisfying the following commutative diagrams called the *c-lens laws*:

$$\begin{tikzcd} & (G \downarrow \mathbf{V}) \arrow[dl, "P_1"'] \arrow[dr, "R_1"] & \\ \mathbf{S} \arrow[rr, "G"'] & & \mathbf{V} \end{tikzcd} \qquad \begin{tikzcd} & (G \downarrow \mathbf{V}) \arrow[dr, "P_1"] & \\ \mathbf{S} \arrow[ur, "\eta"] \arrow[rr, "1_{\mathbf{S}}"'] & & \mathbf{S} \end{tikzcd} \qquad \begin{tikzcd} (R_1 \downarrow \mathbf{V}) \arrow[r, "P_2"] \arrow[d, "\mu"'] & (G \downarrow \mathbf{V}) \arrow[d, "P_1"] \\ (G \downarrow \mathbf{V}) \arrow[r, "P_1"'] & \mathbf{S} \end{tikzcd}$$

In order, these diagrams are known as Put-Get, Get-Put, and Put-Put.

We note the three c-lens laws expressed in the Introduction are exactly the explicit equations given by the diagrams above. However instead of defining the functors R_1, η, μ, P_2 explicitly, we prefer to use the universal property of the comma category in (4.1), (4.2), (4.4), and (4.7), respectively.

The definition of a c-lens was first stated in [4]. Among mathematicians however, such structures had already been considered and they amount to simply recognising G as a *split opfibration*. The above definition coincides with the characterisation of split opfibrations as strict algebras for the KZ-monad appearing in [5].

The goal of this chapter is to recast the definition of c-lenses within the context of internal category theory. Working internal to $\mathbf{Cat}$, we characterise a c-lens as a double category $\mathbb{\Lambda} = (\mathbf{S}, (G \downarrow \mathbf{V}))$ and show this category takes part in a commutative triangle of double functors. We show how these triangles compose, and conclude with a definition of the category $\mathbf{Clens}$ whose objects are categories and whose morphisms are c-lenses.

4.1 The Get functor

The forward direction of a c-lens $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ is simply a functor $G\colon \mathbf{S} \to \mathbf{V}$ called the Get functor. The Get functor induces a canonical double functor $\mathrm{sq}(G)\colon\ \mathrm{sq}(\mathbf{S}) \to \mathrm{sq}(\mathbf{V})$ which, by virtue of being a double functor, consists of a pair of functors,

$$G\colon \mathbf{S} \longrightarrow \mathbf{V} \qquad\qquad \Phi G\colon \Phi\mathbf{S} \longrightarrow \Phi\mathbf{V} \tag{Get}$$

satisfying the following commutative diagrams:

$$\begin{tikzcd}
\mathbf{S} \arrow[d, "G"'] & \Phi\mathbf{S} \arrow[l, "L_1"'] \arrow[r, "R_1"] \arrow[d, "\Phi G"] & \mathbf{S} \arrow[d, "G"] \\
\mathbf{V} & \Phi\mathbf{V} \arrow[l, "L_1"] \arrow[r, "R_1"'] & \mathbf{V}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{S} \arrow[r, "\Delta"] \arrow[d, "G"'] & \Phi\mathbf{S} \arrow[d, "\Phi G"] & \Phi\mathbf{S} \times_{\mathbf{S}} \Phi\mathbf{S} \arrow[l, "\mu"'] \arrow[d, "\Phi G \times \Phi G"] \\
\mathbf{V} \arrow[r, "\Delta"'] & \Phi\mathbf{V} & \Phi\mathbf{V} \times_{\mathbf{V}} \Phi\mathbf{V} \arrow[l, "\mu"]
\end{tikzcd}$$

The *category of view updates* for the Get functor is given by the pullback,

$$\begin{tikzcd}
(G \downarrow \mathbf{V}) \arrow[r, "Q_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[rr, bend left, "R_1"] & \Phi\mathbf{V} \arrow[r, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\Nearrow"] & \mathbf{V} \arrow[d, "1_{\mathbf{V}}"] \\
\mathbf{S} \arrow[r, "G"'] & \mathbf{V} \arrow[r, "1_{\mathbf{V}}"'] & \mathbf{V}
\end{tikzcd} \tag{4.1}$$

where the outer rectangle commutes up to a natural transformation $\alpha\colon GL_1 \Rightarrow R_1$. This pullback is identified as the comma category, hence the notation $(G \downarrow \mathbf{V})$, and we will freely interchange the use of the "universal property of the pullback" and the "universal property of the comma category" in this section. This universal property induces a canonical factorisation of functor $\Phi G\colon \Phi\mathbf{S} \to \Phi\mathbf{V}$, depicted in the following diagram:

$$\begin{tikzcd}
\Phi\mathbf{S} \arrow[r, dashed, "\overline{G}"'] \arrow[d, "L_1"'] \arrow[rr, bend left, "\Phi G"] & (G \downarrow \mathbf{V}) \arrow[r, "Q_1"'] \arrow[d, "L_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Phi\mathbf{V} \arrow[d, "L_1"] \\
\mathbf{S} \arrow[r, equal] & \mathbf{S} \arrow[r, "G"'] & \mathbf{V}
\end{tikzcd}$$

Remark. If the functor $\overline{G}\colon \Phi\mathbf{S} \to (G \downarrow \mathbf{V})$ is an isomorphism, the functor $G\colon \mathbf{S} \to \mathbf{V}$ is called a *discrete opfibration.*

The universal property for the category of view updates may be used to induce two other canonical functors which will be important in constructing a double category $\mathbb{\Lambda}$ whose category of morphisms is $(G \downarrow \mathbf{V})$ and whose category of objects is $\mathbf{S}$.

Firstly, we define a candidate for the identity map via the following diagram where the outer rectangle commutes:

$$\begin{tikzcd}
\mathbf{S} \arrow[r, dashed, "\eta"] \arrow[d, "1_{\mathbf{S}}"'] \arrow[rr, bend left, "G"] & (G \downarrow \mathbf{V}) \arrow[r, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\Nearrow\alpha"] & \mathbf{V} \arrow[d, "1_{\mathbf{V}}"] \\
\mathbf{S} \arrow[r, equal] & \mathbf{S} \arrow[r, "G"'] & \mathbf{V}
\end{tikzcd} \tag{4.2}$$

Next we extend the diagram (4.1) to define the category $(R_1 \downarrow \mathbf{V})$ as a pullback of the functors $R_1 \colon (G \downarrow \mathbf{V}) \to \mathbf{V}$ and $L_1 \colon \Phi\mathbf{V} \to \mathbf{V}$,

$$\begin{tikzcd}
(R_1 \downarrow \mathbf{V}) \arrow[r, "Q_2"] \arrow[d, "L_2"'] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[rrr, bend left, "R_2"] & \Phi\mathbf{V} \times_{\mathbf{V}} \Phi\mathbf{V} \arrow[r, "R_2"] \arrow[d, "L_2"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Phi\mathbf{V} \arrow[r, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\Nearrow"] & \mathbf{V} \arrow[d, "1_{\mathbf{V}}"] \\
(G \downarrow \mathbf{V}) \arrow[r, "Q_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Phi\mathbf{V} \arrow[r, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\Nearrow"] & \mathbf{V} \arrow[r, "1_{\mathbf{V}}"'] \arrow[d, "1_{\mathbf{V}}"] & \mathbf{V} \\
\mathbf{S} \arrow[r, "G"'] & \mathbf{V} \arrow[r, "1_{\mathbf{V}}"'] & \mathbf{V} &
\end{tikzcd} \tag{4.3}$$

where the rectangle formed by the three upper squares commutes up to a natural transformation $\beta \colon R_1 L_2 \Rightarrow R_2$. Here we again identify the comma category $(R_1 \downarrow \mathbf{V})$ as a particular pullback, as shown in Section 2.3, and use the *pullback pasting lemma.*

We define a candidate for the composition map via the following diagram,

$$\begin{tikzcd}
(R_1 \downarrow \mathbf{V}) \arrow[r, dashed, "\mu"] \arrow[d, "L_2"'] \arrow[rr, bend left, "R_2"] & (G \downarrow \mathbf{V}) \arrow[r, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\overset{\alpha}{\Nearrow}"] & \mathbf{V} \arrow[d, "1_{\mathbf{V}}"] \\
(G \downarrow \mathbf{V}) \arrow[r, "L_1"'] & \mathbf{S} \arrow[r, "G"'] & \mathbf{V}
\end{tikzcd} \tag{4.4}$$

where the outer rectangle commutes up to $\beta\alpha \colon G L_1 L_2 \Rightarrow R_2$.

Altogether the diagrams (4.1), (4.2), and (4.4) suggest the definition of a double category $\mathbb{\Lambda}$ whose category of objects is $\mathbf{\Lambda}_0 \coloneqq \mathbf{S}$ and whose category of morphisms is $\mathbf{\Lambda}_1 \coloneqq (G \downarrow \mathbf{V})$. We already have suitable candidates for a potential domain map $L_1 \colon (G \downarrow \mathbf{V}) \to \mathbf{S}$, a potential identity map $\eta \colon \mathbf{S} \to (G \downarrow \mathbf{V})$, and a potential composition map $\mu \colon (R_1 \downarrow \mathbf{V}) \to (G \downarrow \mathbf{V})$. A suitable candidate for the codomain map will be obtained from the Put functor in the next section.

4.2 The Put functor

The backwards direction of a c-lens $\Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{V}$ is given by the functor,

$$P_1 \colon (G \downarrow \mathbf{V}) \longrightarrow \mathbf{S} \tag{Put}$$

called the Put functor, satisfying the following commutative diagrams:

$$\begin{tikzcd}
& (G \downarrow \mathbf{V}) \arrow[dl, "P_1"'] \arrow[dr, "R_1"] & \\
\mathbf{S} \arrow[rr, "G"'] & & \mathbf{V}
\end{tikzcd}
\qquad
\begin{tikzcd}
& (G \downarrow \mathbf{V}) \arrow[dr, "P_1"] & \\
\mathbf{S} \arrow[ur, "\eta"] \arrow[rr, "1_{\mathbf{S}}"'] & & \mathbf{S}
\end{tikzcd}
\qquad
\begin{tikzcd}
(R_1 \downarrow \mathbf{V}) \arrow[r, "P_2"] \arrow[d, "\mu"'] & (G \downarrow \mathbf{V}) \arrow[d, "P_1"] \\
(G \downarrow \mathbf{V}) \arrow[r, "P_1"'] & \mathbf{S}
\end{tikzcd} \tag{4.5}$$

This is exactly a restatement of the Put functor together with the *c-lens laws* from Definition 16.

Theorem 4. *If the quadruple $\Lambda = (\mathbf{S}, \mathbf{V}, G, P_1)$ forms a c-lens, then the pair $\mathbb{\Lambda} = (\mathbf{\Lambda}_0, \mathbf{\Lambda}_1) = (\mathbf{S}, (G \downarrow \mathbf{V}))$ forms a double category.*

Proof. Given a c-lens $\Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{V}$ consider the functors,

$$\begin{tikzcd} \mathbf{\Lambda}_0 \arrow[r, "\eta" description] & \mathbf{\Lambda}_1 \arrow[l, bend right, "P_1"'] \arrow[l, bend left, "L_1"] & \mathbf{\Lambda}_2 \arrow[l, "\mu"'] \end{tikzcd}$$

denoting the candidates for the domain, codomain, identity, and composition maps for a prospective double category $\mathbb{\Lambda}$ with category of objects $\mathbf{\Lambda}_0 = \mathbf{S}$, category of morphisms $\mathbf{\Lambda}_1 = (G \downarrow \mathbf{V})$, and category of composable morphisms $\mathbf{\Lambda}_2 = (R_1 \downarrow \mathbf{V})$.

Given the functor $\eta \colon \mathbf{S} \to (G \downarrow \mathbf{V})$ defined in (4.2) we have,

$$\begin{tikzcd} & \mathbf{S} \arrow[dl, "1_{\mathbf{S}}"'] \arrow[d, "\eta"] \arrow[dr, "1_{\mathbf{S}}"] & \\ \mathbf{S} & (G \downarrow \mathbf{V}) \arrow[l, "L_1"] \arrow[r, "P_1"'] & \mathbf{S} \end{tikzcd}$$

where the left-hand triangle commutes by construction and the right-hand triangle commutes by the Get-Put law. Therefore the functor $\eta \colon \mathbf{S} \to (G \downarrow \mathbf{V})$ satisfies the diagrams for the identity map of a double category.

In order to show that the diagrams for composition are satisfied, we must first show that the prospective category of composable morphisms $\mathbf{\Lambda}_2 = (R_1 \downarrow \mathbf{V})$ arises as the pullback of the presumptive domain and codomain maps:

$$\begin{tikzcd} ? \arrow[r, dotted] \arrow[d, dotted] \arrow[dr, phantom, "\lrcorner" very near start] & (G \downarrow \mathbf{V}) \arrow[r, "Q_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\lrcorner" very near start] & \Phi\mathbf{V} \arrow[d, "L_1"] \\ (G \downarrow \mathbf{V}) \arrow[r, "P_1"'] & \mathbf{S} \arrow[r, "G"'] & \mathbf{V} \end{tikzcd} \tag{4.6}$$

Now given that the right-hand square of (4.6) is a pullback, the well-known *pullback pasting lemma* states that the left-hand square is a pullback if and only if the outer rectangle is a pullback. We have that $GP_1 = R_1 = R_1 Q_1 \colon (G \downarrow \mathbf{V}) \to \mathbf{V}$ by the Put-Get law, and the pullback of this functor along $L_1 \colon \Phi\mathbf{V} \to \mathbf{V}$ was computed in (4.3), yielding a solution to (4.6) given by the following diagram,

$$\begin{tikzcd} (R_1 \downarrow \mathbf{V}) \arrow[r, dashed, "P_2"] \arrow[d, "L_2"'] \arrow[dr, phantom, "\lrcorner" very near start] \arrow[rr, bend left, "R_2 Q_2"] & (G \downarrow \mathbf{V}) \arrow[r, "Q_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\lrcorner" very near start] & \Phi\mathbf{V} \arrow[d, "L_1"] \\ (G \downarrow \mathbf{V}) \arrow[r, "P_1"'] \arrow[rr, bend right, "R_1"'] & \mathbf{S} \arrow[r, "G"'] & \mathbf{V} \end{tikzcd} \tag{4.7}$$

where the functor $P_2 \colon (R_1 \downarrow \mathbf{V}) \to (G \downarrow \mathbf{V})$, occasionally known as the *iterated* Put *functor*, is induced by the universal property of the right-hand pullback square. Therefore $\mathbf{\Lambda}_2 = (R_1 \downarrow \mathbf{V})$ is well-defined as the category of composable morphisms.

Given the functor $\mu\colon (R_1 \downarrow \mathbf{V}) \to (G \downarrow \mathbf{V})$ defined in (4.4) we have,

$$\begin{tikzcd}
(G \downarrow \mathbf{V}) \arrow[d, "L_1"'] & (R_1 \downarrow \mathbf{V}) \arrow[l, "L_2"'] \arrow[r, "P_2"] \arrow[d, "\mu"] & (G \downarrow \mathbf{V}) \arrow[d, "P_1"] \\
\mathbf{S} & (G \downarrow \mathbf{V}) \arrow[l, "L_1"] \arrow[r, "P_1"'] & \mathbf{S}
\end{tikzcd}$$

where the left-hand square commutes by construction and the right-hand square commutes by the Put-Put law. Therefore the functor $\mu\colon (R_1 \downarrow \mathbf{V}) \to (G \downarrow \mathbf{V})$ satisfies the diagrams for the composition maps of a double category.

The proof that composition is unital and associative may be outlined as follows: first construct the appropriate universal functor as in Definition 1, then compose with the composition functor μ, and finally use the universal property of the comma category to show the corresponding diagrams hold. We omit the routine verification of these details. □

Definition 17. Given a c-lens $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$, its *double category of view updates* $\mathbb{\Lambda}$ is given by the following categories and functors:

$$\begin{tikzcd}
\mathbf{\Lambda}_0 \arrow[r, "\eta"] & \mathbf{\Lambda}_1 \arrow[l, bend right, "P_1"'] \arrow[l, bend left, "L_1"] & \mathbf{\Lambda}_2 \arrow[l, "\mu"'] \arrow[l, bend right, "P_2"'] \arrow[l, bend left, "L_2"]
\end{tikzcd}$$

Recall that the comma category $(G \downarrow \mathbf{V})$ whose comma square commutes up to a natural transformation $\alpha\colon GL_1 \Rightarrow R_1$, has objects given by pairs $(S, \alpha\colon GS \to V)$, where S is an object of $\mathbf{S}$ and α is a morphism of $\mathbf{V}$ (by an abuse of notation), and morphisms given by pairs $\langle f, g\rangle\colon (S, \alpha) \to (S', \alpha')$, where $f\colon S \to S'$ is a morphism in $\mathbf{S}$, and $g\colon V \to V'$ is a morphism in $\mathbf{V}$, such that $g \circ \alpha = \alpha' \circ Gf$.

We denote an arbitrary 2-cell of the double category $\mathbb{\Lambda}$ by a commutative square, where we recall by the Put-Get law that $GP = R_1$:

$$\begin{tikzcd}
GS \arrow[r, "\alpha"] \arrow[d, "G(f)"'] & V \arrow[d, "g"] \\
GS' \arrow[r, "\alpha'"'] & V'
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
GS \arrow[r, "\alpha"] \arrow[d, "G(f)"'] & GP(S,\alpha) \arrow[d, "GP\langle f,g\rangle"] \\
GS' \arrow[r, "\alpha'"'] & GP(S',\alpha')
\end{tikzcd}$$

Similarly to the double category of squares, the domain, identity, and codomain maps of the the double category of view updates form an adjoint triple in a result analogous to Lemma 1. A reference for this result is [14].

Lemma 5. *The unit is left-adjoint right-inverse to the left projection $\eta \dashv L_1$ with counit $\varepsilon\colon \eta L_1 \Rightarrow 1_{(G\downarrow\mathbf{V})}$ defined by the whiskered natural transformations:*

$$L_1\varepsilon = 1_{L_1}\colon L_1 \Rightarrow L_1 \qquad R_1\varepsilon = \alpha\colon GL_1 \Rightarrow R_1.$$

Dually, the unit is right-adjoint right-inverse to the Put functor $P_1 \dashv \eta$ with unit $\zeta\colon 1_{(G\downarrow\mathbf{V})} \Rightarrow \eta P_1$ defined by the whiskered natural transformations:

$$L_1\zeta = P_1\varepsilon\colon L_1 \Rightarrow P_1 \qquad R_1\zeta = 1_{R_1}\colon R_1 \Rightarrow R_1.$$

Given an object $(S, \alpha\colon GS \to V)$ of the category of view updates $(G \downarrow \mathbf{V})$, the components of the counit ε and the unit ζ are given by the following commutative squares, respectively:

$$\begin{tikzcd} GS \arrow[r, "1_{GS}"] \arrow[d, "G(1_S)"'] & GS \arrow[d, "\alpha"] \\ GS \arrow[r, "\alpha"'] & V \end{tikzcd} \qquad \begin{tikzcd} GS \arrow[r, "\alpha"] \arrow[d, "GP\langle 1_S, \alpha\rangle"'] & V \arrow[d, "1_V"] \\ GP(S,\alpha) \arrow[r, "1_V"'] & V \end{tikzcd}$$

4.3 A c-lens as a commuting triangle of double functors

Consider a c-lens $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ together with the corresponding view update double category $\mathbb{\Lambda} = (S, (G \downarrow \mathbf{V}))$. We now introduce two canonical double functors to the double categories of squares induced by the source and view categories, which will form a commuting triangle with the double functor induced Get functor.

Firstly, using Lemma 5 and the universal property of the comma category we may define a functor $K\colon (G \downarrow \mathbf{V}) \to \Phi\mathbf{S}$ via the following diagram:

$$\begin{tikzcd} & (G \downarrow \mathbf{V}) \arrow[dl, "L_1"'] \arrow[dr, "P_1"] & \\ \mathbf{S} \arrow[dr, "1_{\mathbf{S}}"'] \arrow[rr, phantom, "\xRightarrow{P_1\varepsilon}"] & & \mathbf{S} \arrow[dl, "1_{\mathbf{S}}"] \\ & \mathbf{S} & \end{tikzcd} \quad = \quad \begin{tikzcd} & (G \downarrow \mathbf{V}) \arrow[d, dashed, "K"] \arrow[ddl, bend right, "L_1"'] \arrow[ddr, bend left, "P_1"] & \\ & \Phi\mathbf{S} \arrow[dl, "L_1"'] \arrow[dr, "R_1"] & \\ \mathbf{S} \arrow[dr, "1_{\mathbf{S}}"'] \arrow[rr, phantom, "\xRightarrow{\phi}"] & & \mathbf{S} \arrow[dl, "1_{\mathbf{S}}"] \\ & \mathbf{S} & \end{tikzcd}$$

Recall from (4.1) we also have a functor $Q_1\colon (G \downarrow \mathbf{V}) \to \Phi\mathbf{V}$ from the category of view updates to the arrow category of the view.

There is a double functor $\mathcal{K}\colon \mathbb{\Lambda} \to \mathrm{sq}(\mathbf{S})$ consisting of a pair of functors,

$$1_{\mathbf{S}}\colon \mathbf{S} \longrightarrow \mathbf{S} \qquad K\colon (G \downarrow \mathbf{V}) \longrightarrow \Phi\mathbf{S}$$

which, due to the Get-Put and Put-Put laws, satisfies the following commutative diagrams:

$$\begin{tikzcd} \mathbf{S} \arrow[d, "1_{\mathbf{S}}"'] & (G \downarrow \mathbf{V}) \arrow[l, "L_1"'] \arrow[r, "P_1"] \arrow[d, "K"] & \mathbf{S} \arrow[d, "1_{\mathbf{S}}"] \\ \mathbf{S} & \Phi\mathbf{S} \arrow[l, "L_1"] \arrow[r, "R_1"'] & \mathbf{S} \end{tikzcd} \qquad \begin{tikzcd} \mathbf{S} \arrow[r, "\eta"] \arrow[d, "1_{\mathbf{S}}"'] & (G \downarrow \mathbf{V}) \arrow[d, "K"] & (R_1 \downarrow \mathbf{V}) \arrow[l, "\mu"'] \arrow[d, "K \times K"] \\ \mathbf{S} \arrow[r, "\Delta"'] & \Phi\mathbf{S} & \Phi\mathbf{S} \times_{\mathbf{S}} \Phi\mathbf{S} \arrow[l, "\mu"] \end{tikzcd}$$

There is also a double functor $\mathcal{Q}_1\colon \mathbb{\Lambda} \to \mathrm{sq}(\mathbf{V})$ consisting of a pair of functors,

$$G\colon \mathbf{S} \to \mathbf{V} \qquad Q_1\colon (G \downarrow \mathbf{V}) \to \Phi\mathbf{V}$$

which, due to the Put-Get law, satisfies the following commutative diagrams:

$$\begin{tikzcd} \mathbf{S} \arrow[d, "G"'] & (G \downarrow \mathbf{V}) \arrow[l, "L_1"'] \arrow[r, "P_1"] \arrow[d, "Q_1"] \arrow[dr, dotted, "R_1"] & \mathbf{S} \arrow[d, "G"] \\ \mathbf{V} & \Phi\mathbf{V} \arrow[l, "L_1"] \arrow[r, "R_1"'] & \mathbf{V} \end{tikzcd} \qquad \begin{tikzcd} \mathbf{S} \arrow[r, "\eta"] \arrow[d, "G"'] & (G \downarrow \mathbf{V}) \arrow[d, "Q_1"] & (R_1 \downarrow \mathbf{V}) \arrow[l, "\mu"'] \arrow[d, "Q_1 \times Q_1"] \\ \mathbf{V} \arrow[r, "\Delta"'] & \Phi\mathbf{V} & \Phi\mathbf{V} \times_{\mathbf{V}} \Phi\mathbf{V} \arrow[l, "\mu"] \end{tikzcd}$$

Furthermore, using the universal property of the comma category together with the Get-Put law we may show that $\mathrm{sq}(G)\mathcal{K} = \mathcal{Q} \colon \mathbb{A} \to \mathrm{sq}(\mathbf{V})$, which may be expanded into the following commutative triangle of double categories and double functors:

$$\begin{tikzcd}
& & (G \downarrow \mathbf{V}) \arrow[d, shift right, "P_1"'] \arrow[d, shift left, "L_1"] \arrow[dddll, "K"'] \arrow[dddrr, "Q_1"] & & \\
& & \mathbf{S} \arrow[dl, "1_{\mathbf{S}}"'] \arrow[dr, "G"] & & \\
& \mathbf{S} \arrow[rr, "G"'] & & \mathbf{V} & \\
\Phi\mathbf{S} \arrow[ur, shift left, "L_1"] \arrow[ur, shift right, "R_1"'] \arrow[rrrr, "\Phi G"'] & & & & \Phi\mathbf{V} \arrow[ul, shift right, "R_1"'] \arrow[ul, shift left, "L_1"]
\end{tikzcd} \tag{4.8}$$

Corollary 6. *The quadruple $\Lambda = (\mathbf{S}, \mathbf{V}, G, P_1)$ forms a c-lens if and only if the diagram* (4.8) *forms a commuting triangle of double categories and double functors.*

It is interesting that the construction of a c-lens which is based entirely on small categories and functors can be naturally stated as a commuting diagram in **Dbl**. One of the primary benefits of this formulation is the natural definition for composition of c-lenses as morphisms between categories.

4.4 The category Clens

We wish to construct a category **Clens** whose objects are categories and whose morphisms are c-lenses. Such composition is hinted at in [7] however it was not explored in detail. Here we motivate the composition of c-lenses from first principles, and show it coincides with the pullback of their representation as a commuting triangle of double functors.

Given a pair of c-lenses $\Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{V}$ and $\Lambda' \colon \mathbf{V} \rightleftharpoons \mathbf{U}$, we define the Get functor of the composite c-lens $\Lambda' \circ \Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{U}$ to be the composite functor $G'G \colon \mathbf{S} \to \mathbf{U}$. Mirroring the construction in (4.1), the category of view updates for the composite Get functor is given by the pullback:

$$\begin{tikzcd}
(G'G \downarrow \mathbf{U}) \arrow[r, "\widehat{Q}_1"] \arrow[d, "\widehat{L}_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] \arrow[rr, bend left, "\widehat{R}_1"] & \Phi\mathbf{U} \arrow[r, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\Nearrow"] & \mathbf{U} \arrow[d, "1_{\mathbf{U}}"] \\
\mathbf{S} \arrow[r, "G'G"'] & \mathbf{U} \arrow[r, "1_{\mathbf{U}}"'] & \mathbf{U}
\end{tikzcd} \tag{4.9}$$

To define the Put functor for the composite c-lens, we first consider the decomposition of the square (4.9) which, by the *pullback pasting lemma*, is a pullback if and only if the right-hand square below is a pullback:

$$\begin{tikzcd}
(G'G \downarrow \mathbf{U}) \arrow[r, dashed, "G \times 1_{\Phi\mathbf{U}}"] \arrow[d, "\widehat{L}_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] \arrow[rr, bend left, "\widehat{Q}_1"] & (G' \downarrow \mathbf{U}) \arrow[r, "Q'_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \Phi\mathbf{U} \arrow[d, "L_1"] \\
\mathbf{S} \arrow[r, "G"'] & \mathbf{V} \arrow[r, "G'"'] & \mathbf{U}
\end{tikzcd}$$

However using the definition of the functor $K' \colon (G' \downarrow \mathbf{U}) \to \Phi\mathbf{V}$, the left-hand square above is a pullback if and only if the top-left-hand square below is a pullback:

$$
\begin{tikzcd}
(G'G \downarrow \mathbf{U}) \arrow[r, dashed, "G \times 1_{\Phi\mathbf{U}}"] \arrow[d, dashed, "\langle \widehat{L}_1, K'(G \times 1_{\Phi\mathbf{U}}) \rangle"'] \arrow[dr, phantom, "\lrcorner", very near start] & (G' \downarrow \mathbf{U}) \arrow[r, "Q'_1"] \arrow[d, "K'"] \arrow[dd, bend right, "L_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Phi\mathbf{U} \arrow[dd, "L_1"] \\
(G \downarrow \mathbf{V}) \arrow[r, "Q_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Phi\mathbf{V} \arrow[d, "L_1"] & \\
\mathbf{S} \arrow[r, "G"'] & \mathbf{V} \arrow[r, "G'"'] & \mathbf{U}
\end{tikzcd}
$$

Thus we may define the Put functor for the composite c-lens to be the functor:

$$
\begin{tikzcd}
(G'G \downarrow \mathbf{U}) \arrow[r, "\langle \widehat{L}_1, K'(G \times 1_{\Phi\mathbf{U}}) \rangle"'] \arrow[rr, bend left, "P_1 \langle \widehat{L}_1, K'(G \times 1_{\Phi\mathbf{U}}) \rangle"] & (G \downarrow \mathbf{V}) \arrow[r, "P_1"'] & \mathbf{S}
\end{tikzcd}
$$

We omit the details showing that this Put functor satisfies the three c-lens laws.

Definition 18. Given c-lenses $\Lambda = (\mathbf{S}, \mathbf{V}, G, P_1)$ and $\Lambda' = (\mathbf{V}, \mathbf{U}, G', P'_1)$ their *composite* c-lens $\Lambda' \circ \Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{U}$ consists of the Get functor,

$$G'G \colon \mathbf{S} \longrightarrow \mathbf{U}$$

together with the Put functor:

$$P_1 \langle \widehat{L}_1, K'(G \times 1_{\Phi\mathbf{U}}) \rangle \colon (G'G \downarrow \mathbf{V}) \to \mathbf{U}.$$

The identity c-lens is given by the quadruple $1_{\mathbf{S}} = (\mathbf{S}, \mathbf{S}, 1_{\mathbf{S}}, R_1)$.

Now consider a pair of c-lenses $\Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{V}$ and $\Lambda' \colon \mathbf{V} \rightleftharpoons \mathbf{U}$ and their representation as commuting triangles of double functors:

$$
\begin{tikzcd}
& \mathbb{\Lambda} \arrow[dl, "\mathcal{K}"'] \arrow[dr, "\mathcal{Q}_1"] & & \mathbb{\Lambda}' \arrow[dl, "\mathcal{K}'"'] \arrow[dr, "\mathcal{Q}'_1"] & \\
\mathrm{sq}(\mathbf{S}) \arrow[rr, "\mathrm{sq}(G)"'] & & \mathrm{sq}(\mathbf{V}) \arrow[rr, "\mathrm{sq}(G')"'] & & \mathrm{sq}(\mathbf{U})
\end{tikzcd}
\tag{4.10}
$$

Given the composite c-lens $\Lambda' \circ \Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{U}$, by Theorem 4 there is a double category of view updates $\mathbb{\Lambda}' \circ \mathbb{\Lambda}$ defined by the categories,

$$(\mathbf{\Lambda}' \circ \mathbf{\Lambda})_0 \coloneqq \mathbf{S} \qquad\qquad (\mathbf{\Lambda}' \circ \mathbf{\Lambda})_1 \coloneqq (G'G \downarrow \mathbf{U})$$

and by Corollary 6 we have commutative diagrams for the categories of objects and the categories of morphisms, respectively:

$$
\begin{tikzcd}[column sep=small]
& & \mathbf{S} \arrow[dl, "1_{\mathbf{S}}"'] \arrow[dr, "G"] \arrow[d, phantom, "\vee"] & & \\
& \mathbf{S} \arrow[dl, "1_{\mathbf{S}}"'] \arrow[dr, "G"] & {} & \mathbf{V} \arrow[dl, "1_{\mathbf{V}}"'] \arrow[dr, "G'"] & \\
\mathbf{S} \arrow[rr, "G"'] & & \mathbf{V} \arrow[rr, "G'"'] & & \mathbf{U}
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
& & (G'G \downarrow \mathbf{V}) \arrow[dl, "\langle \widehat{L}_1, K'(G \times 1_{\Phi\mathbf{U}}) \rangle"'] \arrow[dr, "G \times 1_{\Phi\mathbf{U}}"] \arrow[d, phantom, "\vee"] & & \\
& (G \downarrow \mathbf{V}) \arrow[dl, "K"'] \arrow[dr, "Q_1"] & {} & (G' \downarrow \mathbf{U}) \arrow[dl, "K'"'] \arrow[dr, "Q'_1"] & \\
\Phi\mathbf{S} \arrow[rr, "\Phi G"'] & & \Phi\mathbf{V} \arrow[rr, "\Phi G'"'] & & \Phi\mathbf{U}
\end{tikzcd}
$$

These combine to yield the following commutative diagram of double categories and double functors:

$$
\begin{tikzcd}
 & & \mathbb{\Lambda}' \circ \mathbb{\Lambda} \arrow[dl, "\widehat{\mathcal{K}'}"'] \arrow[dr, "\widehat{\mathcal{Q}_1}"] \arrow[dd, phantom, very near start, "\vee"] & & \\
 & \mathbf{\Lambda} \arrow[dl, "\mathcal{K}"'] \arrow[dr, "\mathcal{Q}_1"] & & \mathbf{\Lambda}' \arrow[dl, "\mathcal{K}'"'] \arrow[dr, "\mathcal{Q}'_1"] & \\
\mathrm{sq}(\mathbf{S}) \arrow[rr, "\mathrm{sq}(G)"'] & & \mathrm{sq}(\mathbf{V}) \arrow[rr, "\mathrm{sq}(G')"'] & & \mathrm{sq}(\mathbf{U})
\end{tikzcd}
\tag{4.11}
$$

Thus the composition of a pair of c-lenses (4.10) given in Definition 18 may be characterised by the specific choice of pullback $\mathbb{\Lambda}' \circ \mathbb{\Lambda}$ to yield a commuting triangle of double functors (4.11).

Definition 19. Let **Clens** be the *category of c-lenses* whose objects are small categories, whose morphisms are c-lenses, and whose composition is given by Definition 18.

Remark. The category **Clens** is well-defined: the composition of c-lenses is unital and associative, as it arises from the composition of the underlying Get functors, which is unital and associative.

5 Internal characterisation of d-lenses

Consider a pair of small categories $\mathbf{S}$ and $\mathbf{V}$ called the *source category* and *view category*, respectively. The objects of both categories are called *states* while the morphisms are usually called *updates*.

Definition 20. A *d-lens* $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ is a quadruple $\Lambda = (\mathbf{S}, \mathbf{V}, G, k)$ consisting of a functor,

$$G\colon \mathbf{S} \longrightarrow \mathbf{V} \tag{Get}$$

together with a function,

$$k\colon S_0 \times_{V_0} V_1 \longrightarrow S_1 \tag{Put}$$

satisfying the following commutative diagrams called the *d-lens laws*:

$$\begin{tikzcd} & S_0 \times_{V_0} V_1 \arrow[dl, "k"'] \arrow[dr, "l_1"] & \\ S_1 \arrow[rr, "l_1"'] & & S_0 \end{tikzcd} \qquad \begin{tikzcd} & S_0 \times_{V_0} V_1 \arrow[dl, "k"'] \arrow[dr, "q_1"] & \\ S_1 \arrow[rr, "g_1"'] & & V_1 \end{tikzcd} \qquad \begin{tikzcd} & S_0 \times_{V_0} V_1 \arrow[dr, "k"] & \\ S_0 \arrow[ur, "i"] \arrow[rr, "i"'] & & S_1 \end{tikzcd}$$

$$\begin{tikzcd} S_0 \times_{V_0} V_1 \times_{V_0} V_1 \arrow[r, "1\times c"] \arrow[d, "k\times k"'] & S_0 \times_{V_0} V_1 \arrow[d, "k"] \\ S_1 \times_{S_0} S_1 \arrow[r, "c"'] & S_1 \end{tikzcd}$$

In order, these diagrams are known as Put-Dom, Put-Get, Get-Put, and Put-Put.

The definition of a d-lens was first stated equationally by Diskin *et al.* [6] motivated by practical considerations, and was later revised by Johnson and Rosebrugh [7] to remove redundant assumptions and highlight similarities with c-lenses while still retaining the equational style. While our focus is entirely theoretical, we encourage the reader to consult these papers, and references therein, for additional motivation behind the practical use of d-lenses.

Definition 20 is based upon standard definition in [7] with a number of notational and stylistic differences we now remark upon. Firstly we refer to the Put function as the composite,

$$\begin{tikzcd} S_0 \times_{V_0} V_1 \arrow[r, "k"'] \arrow[rr, bend left, "p_1"] & S_1 \arrow[r, "r_1"'] & S_0 \end{tikzcd}$$

whereas the literature refers to $k\colon S_0 \times_{V_0} V_1 \to S_1$ as the Put function and uses an uppercase P notation. What we refer to as the Put-Dom and Put-Get laws for a d-lens are usually called the Put-Inc and Put-Id laws in the literature. The pullback $S_0 \times_{V_0} V_1$ is usually denoted as the set of objects underlying the comma category $(G \downarrow \mathbf{V})$ in the literature, however by definition these sets are equal. Finally, the functions i and $k \times k$ are defined using the universal property in (5.2) and (5.8), respectively, however may also be easily understood in the context of the explicit Get-Put and Put-Put laws for a d-lens in the Introduction.

The goal of this chapter is to provide an entirely theoretical motivation for the definition of a d-lens within the context of internal category theory. Working internal to **Set**, we characterise a d-lens as small category $\mathbf{\Lambda} = (S_0, S_0 \times_{V_0} V_1)$, and show this category takes part in commutative triangle of functors, analogous to the construction for set-based lenses and c-lenses. We show how these commuting triangles compose, and conclude with a definition of the category **Dlens** whose objects are small categories and whose morphisms are d-lenses.

5.1 The Get functor

Consider a d-lens $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ whose forward direction is given by the Get functor $G\colon \mathbf{S} \to \mathbf{V}$ which, by virtue of being a functor, consists of a pair of functions,

$$g_0\colon S_0 \longrightarrow V_0 \qquad\qquad g_1\colon V_0 \longrightarrow V_1 \tag{Get}$$

satisfying the following commutative diagrams:

$$\begin{tikzcd} S_0 \arrow[d, "g_0"'] & S_1 \arrow[l, "l_1"'] \arrow[r, "r_1"] \arrow[d, "g_1"] & S_0 \arrow[d, "g_0"] \\ V_0 & V_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] & V_0 \end{tikzcd} \qquad \begin{tikzcd} S_0 \arrow[r, "i"] \arrow[d, "g_0"'] & S_1 \arrow[d, "g_1"] & S_2 \arrow[l, "c"'] \arrow[d, "g_1 \times g_1"] \\ V_0 \arrow[r, "i"'] & V_1 & V_2 \arrow[l, "c"] \end{tikzcd}$$

The *set of view updates* for the Get functor is given by the pullback,

$$\begin{tikzcd} S_0 \times_{V_0} V_1 \arrow[r, "g_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[rr, bend left, "r_1"] & V_1 \arrow[r, "r_1"] \arrow[d, "l_1"] & V_0 \\ S_0 \arrow[r, "g_0"'] & V_0 & \end{tikzcd} \tag{5.1}$$

henceforth denoted by $\Lambda_1 \coloneqq S_0 \times_{V_0} V_1$. The universal property of the pullback induces a canonical factorisation of the morphism assignment $g_1\colon S_1 \to V_1$ of a Get functor, depicted in the following diagram:

$$\begin{tikzcd}
S_1 \arrow[r, dashed, "\langle l_1, g_1\rangle"'] \arrow[d, "l_1"'] \arrow[rr, bend left, "g_1"] & \Lambda_1 \arrow[r, "q_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V_1 \arrow[d, "l_1"] \\
S_0 \arrow[r, equal] & S_0 \arrow[r, "g_0"'] & V_0
\end{tikzcd}$$

The universal property of the pullback for the set of view updates may be used to induce two other canonical functions, which will be important in constructing a small category $\mathbf{\Lambda}$ whose set of morphisms is Λ_1, thus justifying the notation.

Firstly, we define a candidate for the identity map via the following diagram:

$$\begin{tikzcd}
S_0 \arrow[r, dashed, "i"'] \arrow[d, "1_{S_0}"'] \arrow[rr, bend left, "ig_0"] & \Lambda_1 \arrow[r, "q_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V_1 \arrow[d, "l_1"] \\
S_0 \arrow[r, equal] & S_0 \arrow[r, "g_0"'] & V_0
\end{tikzcd} \tag{5.2}$$

Next we extend the diagram (5.1) to include an additional pullback,

$$\begin{tikzcd}
\Lambda_2 \arrow[r, "q_2"] \arrow[d, "l_2"'] \arrow[rrr, bend left, "r_2"] \arrow[dr, phantom, "\lrcorner", very near start] & V_2 \arrow[r, "r_2"] \arrow[d, "l_2"'] \arrow[dr, phantom, "\lrcorner", very near start] & V_1 \arrow[r, "r_1"] \arrow[d, "l_1"] & V_0 \\
\Lambda_1 \arrow[r, "q_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V_1 \arrow[r, "r_1"'] \arrow[d, "l_1"] & V_0 & \\
S_0 \arrow[r, "g_0"'] & V_0 & &
\end{tikzcd} \tag{5.3}$$

where we denote $\Lambda_2 \coloneqq S_0 \times_{V_0} V_1 \times_{V_0} V_1$. We define a candidate for the composition map via the following diagram:

$$\begin{tikzcd}
\Lambda_2 \arrow[r, dashed, "c"'] \arrow[d, "l_2"'] \arrow[rr, bend left, "cq_2"] & \Lambda_1 \arrow[r, "q_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V_1 \arrow[d, "l_1"] \\
\Lambda_1 \arrow[r, "l_1"'] & S_0 \arrow[r, "g_0"'] & V_0
\end{tikzcd} \tag{5.4}$$

Altogether the diagrams (5.1), (5.2) and (5.4) suggest the definition of a small category $\mathbf{\Lambda}$ whose set of objects is $\Lambda_0 \coloneqq S_0$ and whose set of morphisms is Λ_1. We already have suitable candidates for the domain map $l_1 \colon \Lambda_1 \to S_0$, the identity map $i \colon S_0 \to \Lambda_1$ and the composition map $c \colon \Lambda_2 \to \Lambda_1$. A suitable candidate for the codomain map will be obtained from the Put function in the next section.

5.2 The Put function

The backwards direction of a d-lens $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ is given by a function $k\colon \Lambda_1 \to S_1$ together with the composite,

$$\begin{tikzcd} \Lambda_1 \arrow[r, "k"'] \arrow[rr, bend left, "p_1"] & S_1 \arrow[r, "r_1"'] & S_0 \end{tikzcd} \tag{Put}$$

called the Put function, satisfying the following commutative diagrams:

$$\begin{tikzcd} S_0 \arrow[d, "1_{S_0}"'] & \Lambda_1 \arrow[l, "l_1"'] \arrow[r, "p_1"] \arrow[d, "k"] & S_0 \arrow[d, "1_{S_0}"] \\ S_0 & S_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] & S_0 \end{tikzcd} \quad \begin{tikzcd} S_0 \arrow[r, "i"] \arrow[d, "1_{S_0}"'] & \Lambda_1 \arrow[d, "k"] & \Lambda_2 \arrow[l, "c"'] \arrow[d, "k\times k"] \\ S_0 \arrow[r, "i"'] & S_1 & S_2 \arrow[l, "c"] \end{tikzcd} \quad \begin{tikzcd} & \Lambda_1 \arrow[dl, "k"'] \arrow[dr, "q_1"] & \\ S_1 \arrow[rr, "g_1"'] & & V_1 \end{tikzcd} \tag{5.5}$$

The above diagrams are exactly the *d-lens laws* stated in Definition 20 with the notation Λ_1 and Λ_2 substituted where appropriate. The only addition is the second commutative square from the left, which is simply the definition of the Put function.

Note. Both in Definition 20 and in (5.5) the function $k \times k\colon \Lambda_2 \to S_2$ remains ambiguously undefined. The concerned reader may substitute the well-defined function $\langle kl_2, k(r_1k \times r_2)\rangle\colon \Lambda_2 \to S_2$ for this expression, or wait until (5.8) for the simpler notation to be defined.

Remark. The definition of the Put function together with the Put-Dom, Get-Put, and Put-Put law stated in (5.5) appear exactly like the diagrams satisfied by a functor between small categories, except that $\mathbf{\Lambda} = (\Lambda_0, \Lambda_1)$ is *not* a small category. However we will now show that the d-lens laws (5.5) *induce* the structure of small category on the pair $\mathbf{\Lambda} = (\Lambda_0, \Lambda_1)$ such the pair of functions $1_{S_0}\colon \Lambda_0 \to S_0$ and $k\colon \Lambda_1 \to S_1$ form a functor $K\colon \mathbf{\Lambda} \to \mathbf{S}$.

Theorem 7. *If the quadruple* $\Lambda = (\mathbf{S}, \mathbf{V}, G, k)$ *forms a d-lens, then the pair* $\mathbf{\Lambda} = (\Lambda_0, \Lambda_1) = (S_0, S_0 \times_{V_0} V_1)$ *forms a small category.*

Proof. Given a d-lens $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ consider the functions,

$$\begin{tikzcd} \Lambda_0 \arrow[r, "i"] & \Lambda_1 \arrow[l, bend right, "p_1"'] \arrow[l, bend left, "l_1"] & \Lambda_2 \arrow[l, "c"'] \end{tikzcd}$$

denoting the candidates for the domain, codomain, identity, and composition maps for a prospective small category $\mathbf{\Lambda}$ with set of objects $\Lambda_0 = S_0$, set of morphisms $\Lambda_1 = S_0 \times_{V_0} V_1$, and set of composable morphisms $\Lambda_2 = S_0 \times_{V_0} V_1 \times_{V_0} V_1$.

Given the function $i\colon S_0 \to \Lambda_1$ defined in (5.2) we have,

$$\begin{tikzcd} & S_0 \arrow[dl, "1_{S_0}"'] \arrow[d, "i"] \\ S_0 & \Lambda_1 \arrow[l, "l_1"] \end{tikzcd} \qquad \begin{tikzcd} S_0 \arrow[d, "i"'] \arrow[dr, "i"] \arrow[drr, "1_{S_0}"] & & \\ \Lambda_1 \arrow[r, "k"] \arrow[rr, bend right, "p_1"'] & S_1 \arrow[r, "r_1"] & S_0 \end{tikzcd}$$

where the left-hand diagram commutes by construction and the right-hand diagram commutes by the Get-Put law and the structure of $\mathbf{S}$ as a small category. Therefore the function $i\colon S_0 \to \Lambda_1$ satisfies the diagrams for the identity map of a small category.

In order to show the diagrams for composition are satisfied, we must first show that the prospective set of composable morphisms $\Lambda_2 = S_0 \times_{V_0} V_1 \times_{V_0} V_1$ arises as the following pullback of the presumptive domain and codomain maps:

$$\begin{tikzcd}
? \arrow[rr, dotted] \arrow[d, dotted] \arrow[dr, phantom, very near start, "\lrcorner"] & & \Lambda_1 \arrow[r, "q_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & V_1 \arrow[d, "l_1"] \\
\Lambda_1 \arrow[r, "k"] \arrow[rr, bend right, "p_1"'] & S_1 \arrow[r, "r_1"] & S_0 \arrow[r, "g_0"'] & V_0
\end{tikzcd} \tag{5.6}$$

Now given the right-hand square of (5.6) is a pullback, the well-known *pullback pasting lemma* states that left-hand square is a pullback if and only if the outer rectangle is a pullback. By the Put-Get law and using that the Get functor $G\colon \mathbf{S} \to \mathbf{V}$ preserves codomains, we have the following commutative pentagon:

$$\begin{tikzcd}
& & \Lambda_1 \arrow[dll, "k"'] \arrow[drr, "q_1"] \arrow[ddl, "p_1"'] \arrow[ddr, "r_1"] & & \\
S_1 \arrow[rrrr, "g_1"] \arrow[dr, "r_1"'] & & & & V_1 \arrow[dl, "r_1"] \\
& S_0 \arrow[rr, "g_0"'] & & V_0 &
\end{tikzcd}$$

Thus the pullback of the outer rectangle of (5.6) may be computed from the cospan:

$$\begin{tikzcd}
& & V_1 \arrow[d, "l_1"] \\
\Lambda_1 \arrow[r, "p_1"] \arrow[rr, bend right, "r_1 q_1"'] & S_0 \arrow[r, "g_0"] & V_0
\end{tikzcd}$$

This pullback was constructed in (5.3), yielding the solution to (5.6) given by the following diagram,

$$\begin{tikzcd}
\Lambda_2 \arrow[r, dashed, "p_2"] \arrow[rr, bend left, "r_2 q_2"] \arrow[d, "l_2"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \Lambda_1 \arrow[r, "q_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & V_1 \arrow[d, "l_1"] \\
\Lambda_1 \arrow[r, "p_1"'] \arrow[rr, bend right, "r_1 q_1"'] & S_0 \arrow[r, "g_0"'] & V_0
\end{tikzcd} \tag{5.7}$$

where function $p_2\colon \Lambda_2 \to \Lambda_1$ is induced by the universal property of the right-hand pullback square. Therefore Λ_2 is well-defined as the set of composable morphisms.

Note. We define the function $k \times k\colon \Lambda_2 \to S_2$ used in the Put-Put law between the sets of composable morphisms via the universal property of the pullback,

$$\begin{tikzcd}
\Lambda_2 \arrow[r, dashed, "k \times k"'] \arrow[rr, bend left, "kp_2"] \arrow[d, "l_2"'] & S_2 \arrow[r, "r_2"] \arrow[d, "l_2"'] \arrow[dr, phantom, "\lrcorner", very near start] & S_1 \arrow[d, "l_1"] \\
\Lambda_1 \arrow[r, "k"'] \arrow[rr, bend right, "p_1"'] & S_1 \arrow[r, "r_1"'] & S_0
\end{tikzcd} \tag{5.8}$$

where the outer rectangle equals the left-hand square of (5.7) using the Put-Dom law.

Given the function $c \colon \Lambda_2 \to \Lambda_1$ defined in (5.4) we have,

$$\begin{tikzcd}
\Lambda_1 \arrow[d, "l_1"'] & \Lambda_2 \arrow[l, "l_2"'] \arrow[d, "c"] \\
S_0 & \Lambda_1 \arrow[l, "l_1"]
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & \Lambda_1 \arrow[d, "k"'] \arrow[dd, bend left, "p_1"] \\
\Lambda_2 \arrow[urr, "p_2"] \arrow[r, "k \times k"'] \arrow[d, "c"'] & S_2 \arrow[r, "r_2"'] \arrow[d, "c"] & S_1 \arrow[d, "r_1"'] \\
\Lambda_1 \arrow[r, "k"'] \arrow[rr, bend right, "p_1"'] & S_1 \arrow[r, "r_1"'] & S_0
\end{tikzcd}$$

where the left-hand diagram commutes by construction and the right-hand diagram commutes by the Put-Put law, the structure of $\mathbf{S}$ as a category, and diagram (5.8). Therefore the function $c \colon \Lambda_2 \to \Lambda_1$ satisfies the diagrams for the composition map of a small category.

The proof that composition is unital and associative may be outlined as follows. First construct the appropriate universal function as indicated in Definition 1, considering Λ_2 as the set of composable morphisms for $\mathbf{\Lambda}$, then compose with the composition map defined in (5.4), and finally show the corresponding diagrams hold through the universal property of Λ_1 as a pullback in (5.1). We omit the routine verification of these details. $\square$

Definition 21. Given a d-lens $\Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{V}$, its *category of view updates* $\mathbf{\Lambda}$ is given by the following sets and functions:

$$\begin{tikzcd}
\Lambda_0 \arrow[r, "i" description] & \Lambda_1 \arrow[l, bend right, "p_1"'] \arrow[l, bend left, "l_1"] \arrow[r, bend right, "p_2"'] \arrow[r, bend left, "l_2"] & \Lambda_2 \arrow[l, "c" description]
\end{tikzcd}$$

5.3 A d-lens as a commuting triangle of functors

Consider a d-lens $\Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{V}$ together with the corresponding view update category $\mathbf{\Lambda} = (\Lambda_0, \Lambda_1) = (S_0, S_0 \times_{V_0} V_1)$. We now introduce two canonical functors which will form a commuting triangle with the Get functor.

Given the Put-Dom, Get-Put, and Put-Put laws in (5.5), there exists a functor $K \colon \mathbf{\Lambda} \to \mathbf{S}$ consisting of a pair of functions,

$$1_{S_0} \colon S_0 \longrightarrow S_0 \qquad\qquad k \colon \Lambda_1 \longrightarrow S_1$$

satisfying the following commutative diagrams:

$$
\begin{tikzcd}
S_0 \arrow[d, "1_{S_0}"'] & \Lambda_1 \arrow[l, "l_1"'] \arrow[r, "p_1"] \arrow[d, "k"] & S_0 \arrow[d, "1_{S_0}"] \\
S_0 & S_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] & S_0
\end{tikzcd}
\qquad
\begin{tikzcd}
S_0 \arrow[r, "i"] \arrow[d, "1_{S_0}"'] & \Lambda_1 \arrow[d, "k"] & \Lambda_2 \arrow[l, "c"'] \arrow[d, "k \times k"] \\
S_0 \arrow[r, "i"'] & S_1 & S_2 \arrow[l, "c"]
\end{tikzcd}
$$

Given the Put-Get law in (5.5) together with the diagrams (5.1), (5.2), and (5.4), there exists a functor $Q\colon \mathbf{\Lambda} \to \mathbf{V}$ consisting of a pair of functions,

$$
g_0 \colon S_0 \longrightarrow V_0 \qquad\qquad q_1 \colon \Lambda_1 \longrightarrow V_1
$$

satisfying the following commutative diagrams:

$$
\begin{tikzcd}
S_0 \arrow[d, "g_0"'] & \Lambda_1 \arrow[l, "l_1"'] \arrow[r, "p_1"] \arrow[d, "q_1"] \arrow[dl, phantom, "\llcorner", very near start] & S_0 \arrow[d, "g_0"] \\
V_0 & V_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] & V_0
\end{tikzcd}
\qquad
\begin{tikzcd}
S_0 \arrow[r, "i"] \arrow[d, "g_0"'] & \Lambda_1 \arrow[d, "q_1"] & \Lambda_2 \arrow[l, "c"'] \arrow[d, "q_2"] \\
V_0 \arrow[r, "i"'] & V_1 & V_2 \arrow[l, "c"]
\end{tikzcd}
$$

Furthermore, we can see from the Put-Get law that $GK = Q\colon \mathbf{\Lambda} \to \mathbf{V}$, which may be expanded into the following commutative triangle of small categories and functors:

$$
\begin{tikzcd}
 & & \Lambda_1 \arrow[d, shift right, "p_1"'] \arrow[d, shift left, "l_1"] \arrow[ddll, "k"'] \arrow[ddrr, "q_1"] & & \\
 & & S_0 \arrow[dl, "1_{S_0}"'] \arrow[dr, "g_0"] & & \\
 & S_0 \arrow[rr, "g_0"'] & & V_0 & \\
S_1 \arrow[ur, shift left, "l_1"] \arrow[ur, shift right, "r_1"'] \arrow[rrrr, "g_1"'] & & & & V_1 \arrow[ul, shift right, "r_1"'] \arrow[ul, shift left, "l_1"]
\end{tikzcd}
\tag{5.9}
$$

Corollary 8. *The quadruple* $\Lambda = (\mathbf{S}, \mathbf{V}, G, k)$ *forms a d-lens if and only if the diagram* (5.9) *forms a commuting triangle of small categories and functors.*

Thus a d-lens $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ may be understood as simultaneously a *functor* and a *span of functors* between the source and view categories:

$$
\begin{tikzcd}
 & \mathbf{\Lambda} \arrow[dl, "K"'] \arrow[dr, "Q"] & \\
\mathbf{S} \arrow[rr, "G"'] & & \mathbf{V}
\end{tikzcd}
$$

This allows us characterise a d-lens as a morphism rather than just an object and allows for a natural definition for composition of d-lenses as morphisms between categories.

5.4 The category Dlens

We wish to construct a category **Dlens** whose objects are categories and whose morphisms are d-lenses. In the literature [6, 7] an explicit definition for the composite of d-lenses is stated, however there is little justification provided outside of practical considerations. Here we motivate the composition of d-lenses from first principles and show it coincides with the pullback of their representation as a commuting triangle of functors.

Given a pair of d-lenses $\Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{V}$ and $\Lambda'\colon \mathbf{V} \rightleftharpoons \mathbf{U}$, we define the Get functor of the composite d-lens $\Lambda' \circ \Lambda\colon \mathbf{S} \rightleftharpoons \mathbf{U}$ to be the composite functor $G'G\colon \mathbf{S} \to \mathbf{U}$ consisting of a pair of functions:

$$g'_0 \circ g_0\colon S_0 \longrightarrow U_0 \qquad\qquad g'_1 \circ g_1\colon S_1 \longrightarrow U_1$$

Mirroring the construction in (5.1), the set of view updates for the composite Get functor is given by the pullback:

$$\begin{tikzcd}
S_0 \times_{U_0} U_1 \arrow[r, dashed, "\widehat{q_1}"] \arrow[d, dashed, "\widehat{l_1}"'] \arrow[dr, phantom, "\lrcorner", very near start] & U_1 \arrow[d, "l_1"] \\
S_0 \arrow[r, "g'_0 g_0"'] & U_0
\end{tikzcd} \tag{5.10}$$

Defining a natural choice of function $S_0 \times_{U_0} U_1 \to S_1$ satisfying the d-lens laws leads us to consider the decomposition of the square (5.10) which, by the *pullback pasting lemma*, is a pullback if and only if the right-hand square below is a pullback:

$$\begin{tikzcd}
S_0 \times_{U_0} U_1 \arrow[r, dashed] \arrow[d, dashed] \arrow[dr, phantom, "\lrcorner", very near start] & \Lambda'_1 \arrow[r, "q'_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & U_1 \arrow[d, "l_1"] \\
S_0 \arrow[r, "g_0"'] & V_0 \arrow[r, "g'_0"'] & U_0
\end{tikzcd}$$

However using the Put-Dom law for the d-lens $\Lambda'\colon \mathbf{V} \rightleftharpoons \mathbf{U}$, the left-hand square above is a pullback if and only if the top-left-hand square below is a pullback:

$$\begin{tikzcd}
S_0 \times_{U_0} U_1 \arrow[r, dashed] \arrow[d, dashed] \arrow[dr, phantom, "\lrcorner", very near start] & \Lambda'_1 \arrow[r, "q'_1"] \arrow[d, "k'"] \arrow[dd, bend right=60, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & U_1 \arrow[dd, "l_1"] \\
\Lambda_1 \arrow[r, "q_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V_1 \arrow[d, "l_1"] & \\
S_0 \arrow[r, "g_0"'] & V_0 \arrow[r, "g'_0"'] & U_0
\end{tikzcd} \tag{5.11}$$

The dashed functions in (5.11) above defined using the universal property of the pullbacks via the following diagrams:

$$\begin{tikzcd}
S_0 \times_{U_0} U_1 \arrow[r, dashed, "g_0 \times 1_{U_1}"'] \arrow[rr, bend left, "\widehat{q_1}"] \arrow[d, "\widehat{l_1}"'] & V_0 \times_{U_0} U_1 \arrow[r, "q'_1"'] \arrow[d, "l_1"] \arrow[dr, phantom, "\lrcorner", very near start] & U_1 \arrow[d, "l_1"] \\
S_0 \arrow[r, "g_0"'] & V_0 \arrow[r, "g'_0"'] & U_0
\end{tikzcd}$$

$$
\begin{tikzcd}
& V_0 \times_{U_0} U_1 \arrow[dr, "k'"] & \\
S_0 \times_{U_0} U_1 \arrow[ur, "g_0 \times 1_{U_1}"] \arrow[r, dashed, "\langle \widehat{l_1}, k'(g_0 \times 1_{U_1}) \rangle"'] \arrow[d, "\widehat{l_1}"'] & S_0 \times_{V_0} V_1 \arrow[r, "q_1"'] \arrow[d, "l_1"] \arrow[dr, phantom, very near start, "\lrcorner"] & V_1 \arrow[d, "l_1"] \\
S_0 \arrow[r, equal] & S_0 \arrow[r, "g_0"'] & V_0
\end{tikzcd}
$$

We now define the Put function for the composite d-lens $\Lambda' \circ \Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{U}$ to be:

$$
\begin{tikzcd}
S_0 \times_{U_0} U_1 \arrow[r, "\langle \widehat{l_1}, k'(g_0 \times 1_{U_1}) \rangle"'] \arrow[rrr, bend left, "p_1 \langle l_1, k'(g_0 \times 1_{U_1}) \rangle"] & \Lambda_1 \arrow[r, "k"'] & S_1 \arrow[r, "r_1"'] & S_0
\end{tikzcd}
\tag{5.12}
$$

We omit the routine verification that this Put function satisfies the d-lens laws, however note that it matches exactly the composite Put function given explicitly in the literature.

Definition 22. Given d-lens $\Lambda = (\mathbf{S}, \mathbf{V}, G, k)$ and $\Lambda' = (\mathbf{V}, \mathbf{U}, G', k')$ their *composite d-lens* $\Lambda' \circ \Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{U}$ consists of the Get functor,

$$G'G \colon \mathbf{S} \longrightarrow \mathbf{U}$$

together with the function:

$$k\langle \widehat{l_1}, k'(g_0 \times 1_{U_1}) \rangle \colon S \times_{U_0} U_1 \longrightarrow S_1$$

The identity d-lens is given by the quadruple $1_{\mathbf{S}} = (\mathbf{S}, \mathbf{S}, 1_{\mathbf{S}}, 1_{S_1})$.

Now consider a pair of d-lenses $\Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{V}$ and $\Lambda' \colon \mathbf{V} \rightleftharpoons \mathbf{U}$ and their representation as commuting triangles:

$$
\begin{tikzcd}
& \mathbf{\Lambda} \arrow[dl, "K"'] \arrow[dr, "Q"] & & \mathbf{\Lambda'} \arrow[dl, "K'"'] \arrow[dr, "Q'"] & \\
\mathbf{S} \arrow[rr, "G"'] & & \mathbf{V} \arrow[rr, "G'"'] & & \mathbf{U}
\end{tikzcd}
\tag{5.13}
$$

Given the composite d-lens $\Lambda' \circ \Lambda \colon \mathbf{S} \rightleftharpoons \mathbf{U}$, by Theorem 7 there is a category of view updates $\mathbf{\Lambda' \circ \Lambda}$ defined by,

$$(\Lambda' \circ \Lambda)_0 \coloneqq S_0 \qquad\qquad (\Lambda' \circ \Lambda)_1 \coloneqq S_0 \times_{U_0} U_1$$

with Put function (5.12), and by Corollary 8 have commutative diagrams for the sets of objects and the sets of morphisms, respectively,

$$
\begin{tikzcd}
& & (\Lambda' \circ \Lambda)_0 \arrow[dl, "1_{S_0}"'] \arrow[dr, "g_0"] \arrow[dd, phantom, very near start, "\vee"] & & \\
& \Lambda_0 \arrow[dl, "1_{S_0}"'] \arrow[dr, "g_0"] & & \Lambda'_0 \arrow[dl, "1_{V_0}"'] \arrow[dr, "g'_0"] & \\
S_0 \arrow[rr, "g_0"'] & & V_0 \arrow[rr, "g'_0"'] & & U_0
\end{tikzcd}
\qquad
\begin{tikzcd}
& & (\Lambda' \circ \Lambda)_1 \arrow[dl, "\langle \widehat{l_1}, k'(g_0 \times 1_{U_1}) \rangle"'] \arrow[dr, "g_0 \times 1_{U_1}"] \arrow[dd, phantom, very near start, "\vee"] & & \\
& \Lambda_1 \arrow[dl, "k"'] \arrow[dr, "q_1"] & & \Lambda'_1 \arrow[dl, "k'"'] \arrow[dr, "q'_1"] & \\
S_1 \arrow[rr, "g_1"'] & & V_1 \arrow[rr, "g'_1"'] & & U_1
\end{tikzcd}
$$

which combine to yield the following commutative diagram of small categories and functors:

$$
\begin{tikzcd}
 & & \mathbf{\Lambda}' \circ \mathbf{\Lambda} \arrow[dl, "\widehat{K'}"'] \arrow[dr, "\widehat{Q}"] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & & \\
 & \mathbf{\Lambda} \arrow[dl, "K"'] \arrow[dr, "Q"] & & \mathbf{\Lambda}' \arrow[dl, "K'"'] \arrow[dr, "Q'"] & \\
\mathbf{S} \arrow[rr, "G"'] & & \mathbf{V} \arrow[rr, "G'"'] & & \mathbf{U}
\end{tikzcd}
\tag{5.14}
$$

Thus the composition of a pair of d-lenses (5.13) given in Definition 22 may be characterised by the specific choice of pullback $\mathbf{\Lambda}' \circ \mathbf{\Lambda}$ to yield a commuting triangle of functors (5.14).

Definition 23. Let **Dlens** be the *category of d-lenses* whose objects are small categories, whose morphisms are d-lenses, and whose composition is given by Definition 22.

Remark. The category **Dlens** is well-defined: the composition of lenses is unital and associative, as it arises from the composition of the underlying Get functors, which is unital and associative.

Remark. The category **Lens** forms a full subcategory of **Dlens**. Every d-lens between codiscrete categories induces a set-based lens, and vice versa.

Remark. The category **Clens** forms a subcategory of **Dlens**. Every c-lens is a d-lens by considering the underlying horizontal category of each double category.

6 Conclusion

In this thesis, we have shown how set-based lenses, c-lenses, and d-lenses are all different instances of the same internal construction. In each case, an internal *object of view updates* is formed via pullback, as was shown in diagrams (3.1), (4.1), and (5.1):

$$
\begin{tikzcd}
S \times V \arrow[r, dashed, "g \times 1_V"] \arrow[rr, bend left, "r_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \times V \arrow[r, "r_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V \arrow[d, "!"] \\
S \arrow[r, "g"'] & V \arrow[r, "!"'] & \{*\}
\end{tikzcd}
\qquad
\begin{tikzcd}
(G \downarrow \mathbf{V}) \arrow[r, "Q_1"] \arrow[rr, bend left, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Phi\mathbf{V} \arrow[r, "R_1"] \arrow[d, "L_1"'] \arrow[dr, phantom, "\Nearrow"] & \mathbf{V} \arrow[d, "1_{\mathbf{V}}"] \\
\mathbf{S} \arrow[r, "G"'] & \mathbf{V} \arrow[r, "1_{\mathbf{V}}"'] & \mathbf{V}
\end{tikzcd}
\qquad
\begin{tikzcd}
S_0 \times_{V_0} V_1 \arrow[r, "q_1"] \arrow[rr, bend left, "r_1"] \arrow[d, "l_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & V_1 \arrow[r, "r_1"] \arrow[d, "l_1"] & V_0 \\
S_0 \arrow[r, "g_0"'] & V_0 &
\end{tikzcd}
$$

When placed side-by-side, the similarities between these diagrams are evident; for example, a set-based lens is exactly a d-lens when the view (and source) categories are codiscrete $\mathbf{V} = (V_0, V_1) = (V, V \times V)$, while the universal properties of the product and comma categories exemplify set-based lenses and c-lenses as very special kinds of lenses internal to **Set** and **Cat**, respectively. It is also clear that taking the underlying objects of the comma category $(G \downarrow \mathbf{V})$ yields the pullback $S_0 \times_{V_0} V_1$, thus providing another way of seeing that every c-lens is a d-lens.

The commonality between the pullbacks defining the object of view updates extends to the internal "double triangle" diagrams in (3.7), (4.8), and (5.9), which become the focal point of the thesis. It is the "if and only if" statements of Corollary 3, Corollary 6, and Corollary 8, to which the title refers — we are characterising set-based lenses, c-lenses, and d-lenses, respectively, using internal categories. The insight that lenses can be informally understood as both functors and spans between internal categories which form a canonical commuting triangle was a surprise. This reinforces the idea that lenses are morphisms rather than objects. Furthermore there is an aesthetic utility to defining a lens as a single commuting diagram rather than the previous axiomatic or equational definitions provided in the literature.

We also wish to emphasise that while heuristically lenses can be said to compose via pullback of the corresponding commuting triangle representation, this is simply a convenient way of noticing an isomorphism with the composite category of view updates. In reality, the composite lens arises from the composition of the associated Get functors which of course is strictly associative and unital. This forces the composition of lenses to be strictly associative and unital, as the particular choice of view update category is always derived from the pullback of the composed Get along the relevant domain map.

While it is tempting to draw a definition of *internal lenses* and see the three kinds of lens explored as special cases, there are subtle differences between them and this justifies their individual treatment. For example, while Theorem 4 and Theorem 7 require the Put-Get law to ensure the object of composable morphisms is well-defined, this is not the case for set-based lenses in Theorem 2 where the product can always be constructed without reference to the Get or Put; however the Put-Get law is still required for the right-unitality axiom. Meanwhile 2-categorical aspects of **Cat** are essential in constructing the arrow and comma categories used for c-lenses, and for the statements of Lemma 1 and Lemma 5 which arise from the KZ-monad aspects considered in [5, 14].

Perhaps the most significant difference between these three types of lenses is the necessity of the function $k\colon \Lambda_1 \to S_1$ used to induce the Put for a d-lens rather than just defining the Put alone. The proof of Theorem 7 requires delicate use of the the d-lens laws to construct a category structure on the pair $\mathbf{\Lambda} = (\Lambda_0, \Lambda_1)$, providing a strong contrast to the direct proofs for set-based lenses and c-lenses. However this difference essentially arises from the specialness of the universal property associated with the object of view updates for set-based lenses and c-lenses, so it is not unexpected that alternative methods are available. It was very surprising however to see the required d-lens laws were unchanged from the definition in [7], and the fact that each is used *exactly once* in the proof of Theorem 7 indicates both the correctness of the internal characterisation and the appropriateness of d-lenses, which were previously thought to be an unpleasant practical compromise in constrast to the universality of c-lenses.

There are a number of results which are outside the scope of this thesis but are worth mentioning briefly. Given that c-lenses are equivalent to split opfibrations, there is another well-known characterisation as functors $\mathbf{V} \to \mathbf{Cat}$ which assign to each view state its fibre category. Therefore a c-lens requires there to be functors between the fibre categories, while the Grothendieck construction produces a c-lens from each functor $\mathbf{V} \to \mathbf{Cat}$, providing an incidental connection to work on so-called Put-based lenses [15, 16] by the bidirectional transformation community. For the case of lenses where the view category is codiscrete, it was shown in [12] that the fibres are all isomorphic to each other, forcing the Get to simply be a projection from a product of sets. The case of d-lenses, however, had not been previously explored, and was found to correspond to *functions* (or object assignments) between the fibre categories. This seemingly abnormal point-of-view warrants further investigation, particularly the possible formation of a 2-category of small categories, “functions”, and “unnatural transformations” in which d-lenses could be fibred.

There are also a number of 2-categorical and double categorical aspects of c-lenses which deserve further treatment in an internal context. Notably the *Chevalley Criterion*, first recorded by Gray [17] and treated abstractly by Street [5], which states that the functor $K\colon (G \downarrow \mathbf{V}) \to \Phi\mathbf{S}$ is a left-adjoint right-inverse functor. The counit of the adjunction can be shown to arise from Lemma 5 and depicts the universal or “least-change” nature of the opcartesian lifts of a c-lens. The primary difference between c-lenses and d-lenses is the presence of this universal property, and its importance for possible practical applications provides the rationale for further study here. In addition, we recall a c-lens can be understood as a double category of view updates, whose horizontal morphisms are the opcartesian lifts, and whose vertical morphisms are all source updates. The Grothendieck construction suggests an alternative where the vertical morphisms are simply those contained in the fibres, and exploration of this potential sub-double category is planned.

Finally, this thesis inspires many enticing possibilities for future work. Extending our internal treatment of asymmetric lenses to their symmetric counterparts is a priority, and it is hoped using internal profunctors may provide a link to the growing body of work [13, 18, 19] on profunctor optics. With most research focused on the local properties of individual lenses, there are many unexplored avenues into the global properties of the categories **Lens**, **Clens**, and **Dlens**; one interesting thread could be to show these categories have all pullbacks and thus allow the possibility of constructing lenses out of lenses. This thesis began as a quest to find a way to incorporate uncertainty into d-lenses inspired by the work of Diskin [20], and recent work by DeWolf [21] on restriction double categories together with further study into internal lenses may provide the missing link to achieve this goal.

References

[1] F. Bancilhon and N. Spyratos. *Update Semantics of Relational Views.* ACM Transactions on Database Systems **6**(4) (1981).

[2] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt. *Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem.* ACM Transactions on Programming Languages and Systems **29**(3) (2007).

[3] J. Gibbons and M. Johnson. *Relating Algebraic and Coalgebraic Descriptions of Lenses.* ECEASST **49** (2012).

[4] M. Johnson, R. Rosebrugh, and R. J. Wood. *Lenses, fibrations and universal translations.* Mathematical Structures in Computer Science **22**(1) (2012).

[5] R. Street. *Fibrations and Yoneda's lemma in a 2-category.* Lecture Notes in Mathematics **420** (1974).

[6] Z. Diskin, Y. Xiong, and K. Czarnecki. *From state- to delta-based bidirectional model transformations: the asymmetric case.* Journal of Object Technology **10** (2011).

[7] M. Johnson and R. Rosebrugh. *Delta lenses and opfibrations.* Electronic Communications of the EASST **57** (2013).

[8] M. Johnson and R. Rosebrugh. *Unifying set-based, delta-based and edit-based lenses.* CEUR Workshop Proceedings **1571** (2016).

[9] M. Hofmann, B. Pierce, and D. Wagner. *Symmetric lenses.* SIGPLAN Not. **46**(1) (2011).

[10] Z. Diskin, Y. Xiong, K. Czarnecki, H. Ehrig, F. Hermann, and F. Orejas. *From State- to Delta-Based Bidirectional Model Transformations: The Symmetric Case.* In *Model Driven Engineering Languages and Systems* (2011).

[11] M. Johnson and R. Rosebrugh. *Cospans and symmetric lenses.* In *Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming* (2018).

[12] M. Johnson, R. Rosebrugh, and R. J. Wood. *Algebras and Update Strategies.* Journal of Universal Computer Science **12**(5) (2010).

[13] M. Riley. *Categories of Optics.* ArXiv e-prints (2018). 1809.00738.

[14] A. Kock. *Fibrations as Eilenberg-Moore algebras.* ArXiv e-prints (2013). 1312.1608.

[15] S. Fischer, Z. Hu, and H. Pacheco. *Monadic Combinators for Putback Style Bidirectional Programming.* In *Proceedings of the ACM SIGPLAN 2014 Workshop on Partial Evaluation and Program Manipulation*, PEPM '14 (2014).

[16] S. Fischer, Z. Hu, and H. Pacheco. *A Clear Picture of Lens Laws.* In *Mathematics of Program Construction* (2015).

[17] J. W. Gray. *Fibred and Cofibred Categories.* Proceedings of the Conference on Categorical Algebra (1966).

[18] M. Pickering, J. Gibbons, and N. Wu. *Profunctor Optics: Modular Data Accessors.* The Art, Science, and Engineering of Programming **1**(2) (2017).

[19] G. Boisseau and J. Gibbons. *What You Needa Know About Yoneda: Profunctor Optics and the Yoneda Lemma.* PACMPL **2**(ICFP) (2018). Functional Pearl.

[20] Z. Diskin, R. Eramo, A. Pierantonio, and K. Czarnecki. *Incorporating Uncertainty into Bidirectional Model Transformations and their Delta-Lens Formalization.* In *Bx@ETAPS* (2016).

[21] D. DeWolf. *Restriction Category Perspectives of Partial Computation and Geometry.* Ph.D. thesis, Dalhousie University (2017).