# Internal Lenses as Functors and Cofunctors

Bryce Clarke
CoACT, Macquarie University, bryce.clarke1@students.mq.edu.au
ACT2019, July 17

# Outline

# Motivation

# What is a lens, and why should I care?

MACQUARIE University

- Lenses are a mathematical structure which aim to capture the notion of synchronisation between a pair of systems.
- Since the introduction of lenses to solve the view-update problem for databases, they have found a variety of uses including:
    - Bidirectional transformations;
    - Haskell, functional programming and optics;
    - Compositional game theory and economics;
    - Model-driven engineering;
    - Supervised learning.
- The goal of this talk is to demonstrate how using the diagrammatic methods of internal category theory reveals lenses as a functor and cofunctor pair.

# The view-update problem and terminology

- A system (e.g. database) may be modelled as a set of *states*.
- Assume we can *update* or transition between any pair of states.
- The only information retained from the update is the initial and final states.
- We call $S$ the set of source states and $V$ the set of view states.
- Each view state is determined by a source state via a function $g \colon S \to V$.
- The view update problem states:

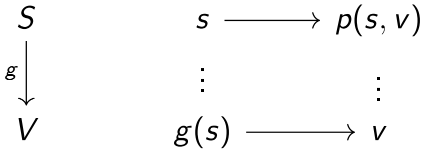
  *Given a source state $s \in S$ and a view update $g(s) \rightsquigarrow v$ of states $g(s), v \in V$, find a source update $s \rightsquigarrow s'$ such that $g(s') = v$.*

- Solving this problem provided one of the original inspirations for the development of lenses, in particular state-based lenses[1].

[1] Also known as *very well-behaved Set-based lenses*, or *lawful monomorphic lenses*

# State-based lenses (schematically)



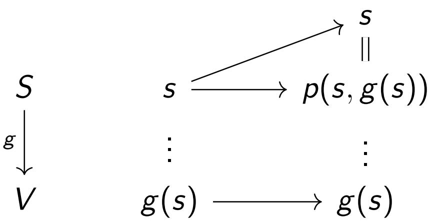



A state-based lens consists of a Get function $g\colon S \to V$ and a Put function $p\colon S \times V \to S$ such that for all $s \in S$ and $v, v' \in V$:

- The Put-Get law holds: $gp(s, v) = v$ as depicted.

# State-based lenses (schematically)

MACQUARIE University

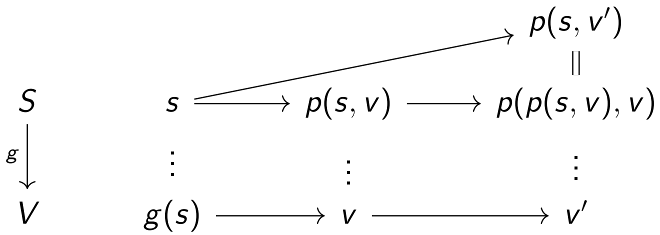



A state-based lens consists of a Get function $g\colon S \to V$ and a Put function $p\colon S \times V \to S$ such that for all $s \in S$ and $v, v' \in V$:

- The Put-Get law holds: $gp(s, v) = v$ .
- The Get-Put law holds: $p(s, g(s)) = s$ as depicted.

# State-based lenses (schematically)






A state-based lens consists of a Get function $g\colon S \to V$ and a Put function $p\colon S \times V \to S$ such that for all $s \in S$ and $v, v' \in V$:

- The Put-Get law holds: $gp(s, v) = v$ .
- The Get-Put law holds: $p(s, g(s)) = s$ .
- The Put-Put law holds: $p(p(s, v), v') = p(s, v')$ as depicted.

# State-based lenses (equationally)

A state-based lens between sets $S$ and $V$ consists of functions,

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

called Get and Put, respectively, satisfying three *lens laws*:

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

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

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

Given a pair of state-based lenses $(g, p) \colon S \rightleftharpoons V$ and $(h, q) \colon V \rightleftharpoons U$ their composite has Get function $hg \colon S \to U$ and Put function $S \times U \to S$ defined by the formula:

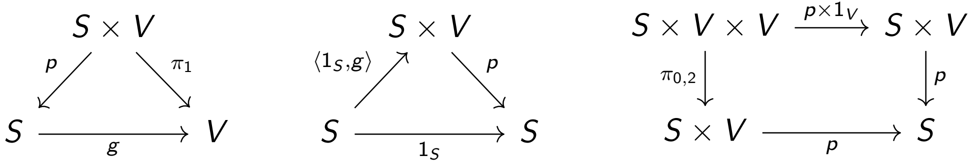
$$(s, u) \longmapsto p(s, q(g(s), u))$$

# State-based lenses (diagramatically)

- A state-based lens between sets $S$ and $V$ consists of functions,

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

called Get and Put, respectively, satisfying three *lens laws*:




- May be generalised internal to any category with products.

# State-based lenses as algebras for a monad

The diagrams for a state-based lens arise as algebras for a well-known monad,

$$\begin{array}{ccc} \mathbf{Set}/V & \longrightarrow & \mathbf{Set}/V \\ g\colon S \to V & \longmapsto & \pi_1\colon S \times V \to V \end{array}$$

where the Put function $p\colon S \times V \to S$ forms the structure map.

- How may we understand lenses as morphisms between sets, rather than *objects* in the category of Eilenberg-Moore algebras?
- Is there a mathematical way of understanding how lenses compose?
- Can we develop a unified framework for lenses which encompasses other known examples such as split opfibrations[2] and delta lenses?

[2] Also known as *c-lenses*.

# Looking further into lenses

# Lenses induce (internal) categories



- The sets $S$ and $V$ induce codiscrete categories $\mathbf{S}$ and $\mathbf{V}$ given by the spans,

$$S \xleftarrow{\pi_0} S \times S \xrightarrow{\pi_1} S \qquad\qquad V \xleftarrow{\pi_0} V \times V \xrightarrow{\pi_1} V$$

  whose morphisms are determined by pairs $(s, s')$ and $(v, v')$ respectively.
- Given a lens $\Lambda \colon S \rightleftharpoons V$ there is a category, denoted $\boldsymbol{\Lambda}$, given by the span:

$$S \xleftarrow{\pi_0} S \times V \xrightarrow{p} S$$

  which we will call the category of lifts – is well-defined using the lens laws.

# Lenses induce (internal) functors

- Given a function $g\colon S \to V$ we obtain a canonical functor $G\colon \mathbf{S} \to \mathbf{V}$ between the codiscrete categories:

$$\begin{array}{ccccc}
S & \xleftarrow{\pi_0} & S \times S & \xrightarrow{\pi_1} & S \\
{\scriptstyle g}\downarrow & & \downarrow{\scriptstyle g \times g} & & \downarrow{\scriptstyle g} \\
V & \xleftarrow[\pi_0]{} & V \times V & \xrightarrow[\pi_1]{} & V
\end{array}$$

- Given a lens consisting of functions $g\colon S \to V$ and $p\colon S \times V \to S$ we obtain functors $K\colon \mathbf{\Lambda} \to \mathbf{S}$ and $\overline{G}\colon \mathbf{\Lambda} \to \mathbf{V}$ given by:

$$\begin{array}{ccccc}
S & \xleftarrow{\pi_0} & S \times V & \xrightarrow{p} & S \\
{\scriptstyle 1_S}\downarrow & & \downarrow{\scriptstyle \langle \pi_0, p \rangle} & & \downarrow{\scriptstyle 1_S} \\
S & \xleftarrow[\pi_0]{} & S \times S & \xrightarrow[\pi_1]{} & S
\end{array}
\qquad
\begin{array}{ccccc}
S & \xleftarrow{\pi_0} & S \times V & \xrightarrow{p} & S \\
{\scriptstyle g}\downarrow & \lrcorner & \downarrow{\scriptstyle g \times 1_V} & \searrow{\scriptstyle \pi_1} & \downarrow{\scriptstyle g} \\
V & \xleftarrow[\pi_0]{} & V \times V & \xrightarrow[\pi_1]{} & V
\end{array}$$

# Central idea: Lenses as commuting triangles

A state-based lens $\Lambda \colon S \rightleftharpoons V$ may be characterised as a commuting triangle of functors,

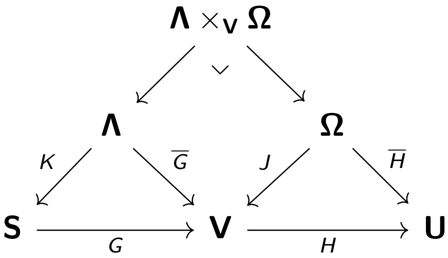
$$\begin{array}{ccc} & \mathbf{\Lambda} & \\ {}^{K}\swarrow & & \searrow^{\overline{G}} \\ \mathbf{S} & \xrightarrow[G]{\quad\quad} & \mathbf{V} \end{array}$$

where

- $\mathbf{S}$ and $\mathbf{V}$ are codiscrete categories,
- $K$ is an identity-on-objects functor,
- $\overline{G}$ is a discrete opfibration.

# Composing lenses via pullback

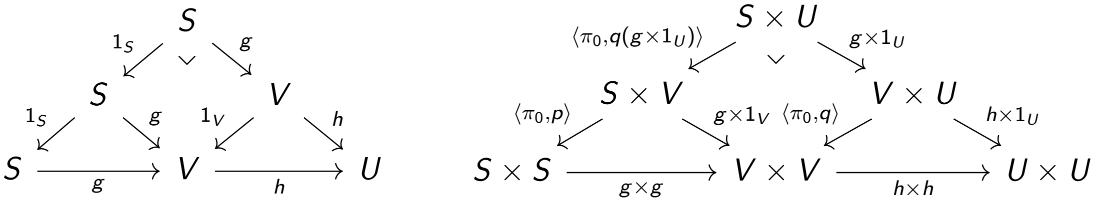
Given a pair of state-based lenses $\Lambda\colon S \rightleftharpoons V$ and $\Omega\colon V \rightleftharpoons U$ their composite may be characterised using the pullback:




Note that both identity-on-objects functors and discrete opfibrations are stable under pullback.

# Composing lenses diagramatically

Given a pair of state-based lenses $(g, p)\colon S \rightleftharpoons V$ and $(h, q)\colon V \rightleftharpoons U$ their composite may be computed using the diagrams:




Thus the composite lens has Get function $hg\colon S \to U$ and Put function:

$$p\langle \pi_0, q(g \times 1_U)\rangle\colon S \times U \longrightarrow S$$

# Internal lenses as functors and cofunctors

# What is a cofunctor between categories?



- A cofunctor should be understood as a kind of lifting.

$$
\begin{array}{c}
\mathbf{S} \\
g \downarrow \;\; \uparrow k \\
\mathbf{V}
\end{array}
\qquad\qquad
\begin{array}{ccc}
s & \xrightarrow{k(s,\alpha)} & p(s,\alpha) \\
\vdots & & \vdots \\
g(s) & \xrightarrow{\alpha} & v
\end{array}
$$

- The codomain $p(s, \alpha)$ of the lift should satisfy $g(p(s, \alpha)) = v$.
- The lifting should respect both identities and composition.
- Becomes a delta lens if there is a functor $G : \mathbf{S} \to \mathbf{V}$ such that $Gk(s, \alpha) = \alpha$.
- Examples of cofunctors include:
  - State-based lenses and delta lenses;
  - Discrete opfibrations and split opfibrations;
  - Identity-on-objects functors, such as monoid and group homomorphisms.

MACQUARIE University

- Every (internal) cofunctor may be represented as a span of (internal) functors,

$$\begin{array}{ccc} & \mathbf{\Lambda} & \\ {\scriptstyle \overline{G}}\swarrow & & \searrow{\scriptstyle K} \\ \mathbf{V} & & \mathbf{S} \end{array}$$

where $\overline{G}$ is a discrete opfibration and $K$ is an identity-on-objects functor.

- Internally a cofunctor between $\mathbf{V}$ and $\mathbf{S}$ may be defined using the diagrams:

$$\begin{array}{ccccc} S_0 & \xleftarrow{\pi_0} & S_0 \times_{V_0} V_1 & \xrightarrow{p_0} & S_0 \\ {\scriptstyle g_0}\downarrow & \text{(pullback)} & \downarrow{\scriptstyle \pi_1} & & \downarrow{\scriptstyle g_0} \\ S_0 & \xleftarrow[l_0]{} & V_1 & \xrightarrow[r_0]{} & V_0 \end{array} \qquad \begin{array}{ccccc} S_0 & \xleftarrow{\pi_0} & S_0 \times_{V_0} V_1 & \xrightarrow{p_0} & S_0 \\ {\scriptstyle 1_{S_0}}\downarrow & & \downarrow{\scriptstyle k_1} & & \downarrow{\scriptstyle 1_{S_0}} \\ S_0 & \xleftarrow[l_0]{} & S_1 & \xrightarrow[r_0]{} & S_0 \end{array}$$

- An internal cofunctor determined by the pair $(g_0, k_1)$ satisfying some axioms.

# Internal lenses as functors and cofunctors



- An internal lens (in a category with pullbacks) is a Get functor $(g_0, g_1)$ and a Put cofunctor $(g_0, k_1)$ such that the following diagram commutes:

$$
\begin{array}{ccc}
 & S_0 \times_{V_0} V_1 & \\
{}^{k_1}\swarrow & & \searrow^{\pi_1} \\
S_1 & \xrightarrow[g_1]{} & V_1
\end{array}
$$

- An internal lens is a commuting diagram of internal functors,

$$
\begin{array}{ccc}
 & \mathbf{\Lambda} & \\
{}^{K}\swarrow & & \searrow^{\overline{G}} \\
\mathbf{S} & \xrightarrow[G]{} & \mathbf{V}
\end{array}
$$

where $\overline{G}$ is a discrete opfibration and $K$ is an identity-on-objects functor.

# Examples of internal lenses

- A delta lens is exactly an internal lens in **Set**.
- A state-based lens consisting of functions,

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

is a delta lens between codiscrete categories, where:

$$k_1 = \langle \pi_0, p \rangle \colon S \times V \to S \times S$$

- A delta lens between monoids is exactly a retraction.
- A discrete opfibration is an internal lens in **Set** where $k_1$ is an isomorphism.
- A split opfibration is an internal lens in **Cat** between double categories of squares, where $k_1$ is a left-adjoint right-inverse functor between categories:

$$1 \circlearrowright (G \downarrow \mathbf{V}) \underset{\longleftarrow}{\overset{k_1}{\underset{\bot}{\longrightarrow}}} \mathbf{S}^2$$

# Summary

MACQUARIE University

- An (internal) lens consists of a Get functor and a Put cofunctor satisfying an axiom akin to a Put-Get law.
- A cofunctor is just a span of functors whose left leg is a discrete opfibration, and whose right leg is an identity-on-object functor.
- A lens may be represented as a commuting diagram of functors:

$$\begin{array}{ccc} & \mathbf{\Lambda} & \\ {\scriptstyle K}\swarrow & & \searrow{\scriptstyle \overline{G}} \\ \mathbf{S} & \xrightarrow[G]{} & \mathbf{V} \end{array}$$

- Composition of lenses is defined by composing the Get functors and Put cofunctors.
- This captures a wide range of lenses which appear in the literature . . . and provides a robust framework for working with future examples!