Profunctor optics: a categorical update

Mario Román

June 14, 2019

SUPERVISOR: Jeremy Gibbons

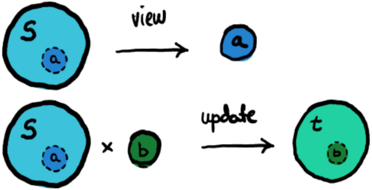
University of Oxford
Mathematical Institute

Oxford Mathematics

Part 1: Motivation

Definition (Lens)

$$\mathbf{Lens}\left(\begin{pmatrix} s \\ t \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}\right) = \mathbf{Sets}(s, a) \times \mathbf{Sets}(s \times b, t).$$

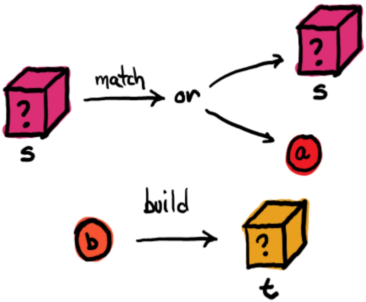



```
-- Example: A postal address contains a ZIP code.
viewZip   : PostalAddr -> ZipCode
updateZip : PostalAddr * ZipCode -> PostalAddr
```

Prisms (alternatives!)

Definition (Prism)

$$\mathbf{Prism}\left(\begin{pmatrix} s \\ t \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}\right) = \mathbf{Sets}(s, s + a) \times \mathbf{Sets}(b, t).$$

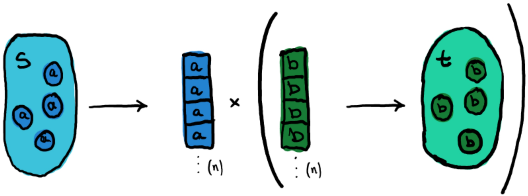



```
-- An addess can be both a postal address or an email.
matchPostal : Address    -> Address + PostalAddr
buildEmail  : EmailAddr  -> Address
```

Traversals (multiple foci!)

Definition (Traversal)

$$\mathbf{Traversal}\left(\begin{pmatrix} s \\ t \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}\right) = \mathbf{Sets}(s, \sum_n a^n \times (b^n \to t)).$$




```
-- A sorted listing of addresses.
extract : MailingList -> Vect n EmailAddress * (Vect n PostalAddress -> PostalList)
```

This is not modular

How to compose a Prism with a Lens? How to get/set a `Zip` from an `Address`?

```
getPostal : Address     -> Address + PostalAddr
setPostal : PostalAddr -> Address
getZip : PostalAddr -> ZipCode
setZip : PostalAddr * ZipCode -> PostalAddr
```

This is not modular

How to compose a Prism with a Lens? How to get/set a `Zip` from an `Address`?

```
getPostal : Address    -> Address + PostalAddr
setPostal : PostalAddr -> Address
getZip : PostalAddr -> ZipCode
setZip : PostalAddr * ZipCode -> PostalAddr
```

The naive solution is not modular. Every case (Prism+Lens, Lens+Prism, Traversal+Prism+Other, ...) needs special attention.

```
-- This is boilerplate code we would rather not write.
getZipFromAddr :: Address -> Address + ZipCode
getZipFromAddr = (\ a -> case getPostal a of
                     Right postal -> Right (getZip postal)
                     Left add -> Left add)

setZipAddr :: Address * ZipCode -> Address
setZipAddr = (\ a z -> case getPostal a of
                 Right postal -> setPostal (setZip (postal , z))
                 Left add     -> add)
```

Profunctor optics

Some optics are equivalent to parametric functions over profunctors!

A lens	$\mathbf{Sets}(s, a) \times \mathbf{Sets}(s \times b, t)$	is also	$p(a, b) \to p(s, t), \forall p \in \mathbf{Mod}(\times)$
A prism	$\mathbf{Sets}(s, s + a) \times \mathbf{Sets}(b, t)$	is also	$p(a, b) \to p(s, t), \forall p \in \mathbf{Mod}(+)$

Where $p \in \mathbf{Tamb}(\otimes)$ means we have a transformation $p(a, b) \to p(c \otimes a, c \otimes b)$.

Profunctor optics

Some optics are equivalent to parametric functions over profunctors!

A lens	$\mathbf{Sets}(s, a) \times \mathbf{Sets}(s \times b, t)$	is also	$p(a, b) \to p(s, t), \forall p \in \mathbf{Mod}(\times)$
A prism	$\mathbf{Sets}(s, s + a) \times \mathbf{Sets}(b, t)$	is also	$p(a, b) \to p(s, t), \forall p \in \mathbf{Mod}(+)$

Where $p \in \mathbf{Tamb}(\otimes)$ means we have a transformation $p(a, b) \to p(c \otimes a, c \otimes b)$.

This solves composition

Now composition of optics is just *function composition*. From $p(a, b) \to p(s, t)$ and $p(x, y) \to p(a, b)$ we can get $p(x, y) \to p(s, t)$.

Goals of this dissertation

- Gathering the literature on this topic.
  - What is a general definition of optic?
  - How does the profunctor representation work in general?
  - Try to provide new proofs, as general as possible (actions of monoidal categories as in *[Riley, 2018]*).
- Description of the traversal from first principles.
  - Problem proposed by *[Milewski, 2017]*: get a description of the traversal and the concrete representation from a single application of Yoneda.
  - Unification of optics, including the traversal.

Outline

- A definition of optics: the (co)end representation.
- Unification with the traversal: derivation of the traversal and new optics.
- How to compose optics: the profunctor representation theorem.
- Further work: formal verification and new directions.

Part 2: A definition of "optic"

(Co)ends

Ends and *Coends* are special kinds of (co)limits over a profunctor $p\colon \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{Sets}$, (co)equalizing its right and left mapping.

$$\int_{x \in \mathbf{C}} p(x,x) \longrightarrow \prod_{x \in \mathbf{C}} p(x,x) \underset{p(f,\mathrm{id})}{\overset{p(\mathrm{id},f)}{\rightrightarrows}} \prod_{f\colon a \to b} p(a,b)$$

$$\coprod_{f\colon b \to a} p(a,b) \underset{p(f,\mathrm{id})}{\overset{p(\mathrm{id},f)}{\rightrightarrows}} \coprod_{x \in \mathbf{C}} p(x,x) \longrightarrow \int^{x \in \mathbf{C}} p(x,x)$$

We can think of them as encoding `forall` (ends) and `exists` (coends).

Fosco Loregian. “This is the (co)end, my only (co)friend”. In: *arXiv preprint arXiv:1501.02503* (2015).

(Co)end calculus

Natural transformations can be rewritten in terms of ends. For any $F, G \colon \mathbf{C} \to \mathbf{D}$,

$$\mathrm{Nat}(F, G) = \int_{x \in \mathbf{C}} \mathbf{D}(Fx, Gx).$$

We can compute (co)ends using Yoneda lemma.

$$Fa \cong \int^{x \in \mathbf{C}} Fx \times \mathbf{C}(a, x), \qquad Ga \cong \int_{x \in \mathbf{C}} \mathbf{Sets}(\mathbf{C}(x, a), Gx).$$

We have a well-behaved formal calculus for (co)ends.

Fosco Loregian. "This is the (co)end, my only (co)friend". In: *arXiv preprint arXiv:1501.02503* (2015).

A definition of "optic"

Fix some action $\mathbf{M} \times \mathbf{C} \to \mathbf{C}$ of a monoidal category $\mathbf{M}$ on $\mathbf{C}$.

Definition (Riley, 2018)

The $\mathbf{Optic}$ category has pairs on $\mathbf{C}$ as objects and morphisms as follows.

$$\mathbf{Optic}_{\mathbf{M}}\left(\begin{pmatrix} s \\ t \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}\right) = \int^{c \in \mathbf{M}} \mathbf{C}(s, c \cdot a) \times \mathbf{C}(c \cdot b, t).$$

Intuition: The optic splits into some focus a and some *context* c. We cannot access that context, but we can use it to update.

Lenses are optics

$$\mathbf{Lens}\left(\begin{pmatrix} s \\ t \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}\right) = \int^{c \in \mathbf{Sets}} \mathbf{Sets}(s, c \times a) \times \mathbf{Sets}(c \times b, t).$$

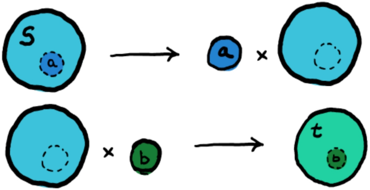
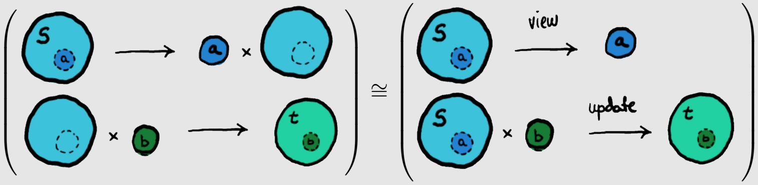


Figure 1: A **lens** is given by $(s \to c \times a)$ and $(c \times b \to t)$ for some c we cannot access.

Lenses are optics

Theorem (from Milewski, 2017)



Proof. By Yoneda lemma.

$$\int^{c \in \mathbf{Sets}} \mathbf{Sets}(s, c \times a) \times \mathbf{Sets}(c \times b, t) \cong \qquad (\textit{ Product })$$

$$\int^{c \in \mathbf{Sets}} \mathbf{Sets}(s, c) \times \mathbf{Sets}(s, a) \times \mathbf{Sets}(c \times b, t) \cong \qquad (\textit{ Yoneda })$$

$$\mathbf{Sets}(s, a) \times \mathbf{Sets}(s \times b, t)$$

Prisms are optics

$$\mathbf{Prism}\left(\begin{pmatrix} s \\ t \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}\right) = \int^{c \in \mathbf{Sets}} \mathbf{Sets}(s, c+a) \times \mathbf{Sets}(c+b, t).$$

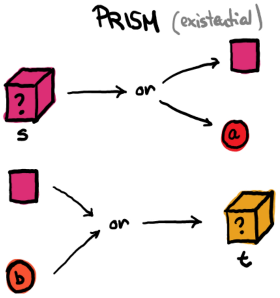
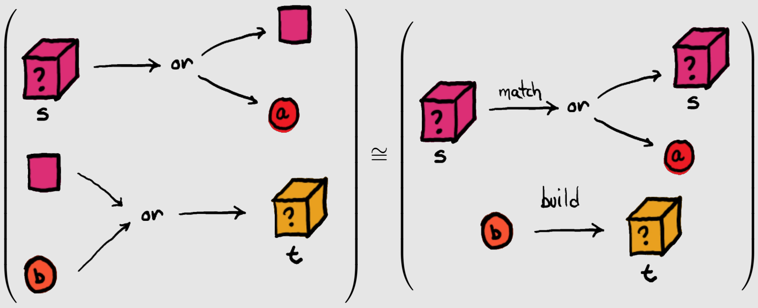


Figure 2: A **prism** is given by $(s \to c + a)$ and $(c + b \to t)$ for some c we cannot access.

Prisms are optics

Theorem (from Milewski, 2017)



Proof. By Yoneda lemma.

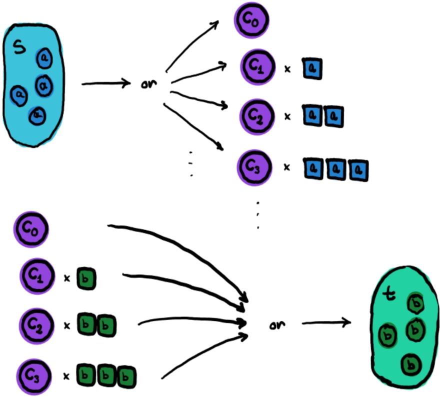
$$\int^{m \in \mathbf{Sets}} \mathbf{Sets}(s, m + a) \times \mathbf{Sets}(m + b, t) \cong \quad \text{(Coproduct)}$$

$$\int^{m \in \mathbf{Sets}} \mathbf{Sets}(s, m + a) \times \mathbf{Sets}(m, t) \times \mathbf{Sets}(m \times b, t) \cong \quad \text{(Yoneda)}$$

$$\mathbf{Sets}(s, t + a) \times \mathbf{Sets}(b, t)$$

Traversals are optics (with a new derivation)

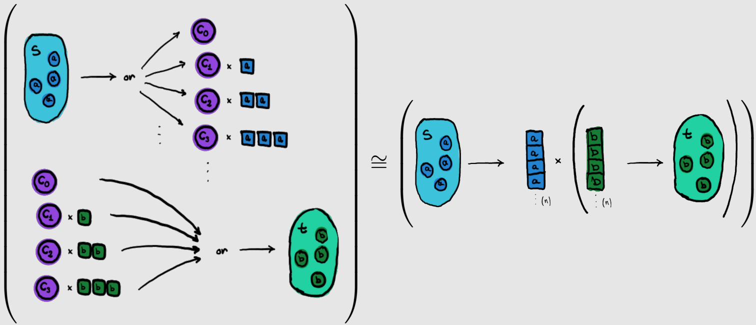
$$\mathbf{Traversal}\left(\binom{s}{t},\binom{a}{b}\right) = \int^{c\in[\mathsf{Nat},\mathbf{Sets}]} \mathbf{Sets}\left(s, \sum_n c_n \times a^n\right) \times \mathbf{Sets}\left(\sum_n c_n \times b^n, t\right).$$



To our knowledge, this is an original formulation of traversals. It should be related to the description in terms of Traversables *[Pickering/Gibbons/Wu, 2016].*

Traversals are optics (with a new derivation)

Theorem



That is,

$$\int^{c\in[\mathsf{Nat},\mathbf{Sets}]} \mathbf{Sets}(s, \sum_n c_n \times a^n) \times \mathbf{Sets}(\sum_n c_n \times b^n, t) \cong \mathbf{Sets}(s \to \sum_n a^n \times (b^n \to t)).$$

Traversals are optics (with a new derivation)

This is Yoneda, this time for functors $c \colon \mathsf{Nat} \to \mathbf{Sets}$.

$$\int^{c} \mathbf{Sets}\left(s, \sum_{n\in\mathbf{N}} c_n \times a^n\right) \times \mathbf{Sets}\left(\sum_{n\in\mathbf{N}} c_n \times b^n, t\right) \cong \quad \textit{(cocontinuity)}$$

$$\int^{c} \mathbf{Sets}\left(s, \sum_{n\in\mathbf{N}} c_n \times a^n\right) \times \prod_{n\in\mathbf{N}} \mathbf{Sets}\left(c_n \times b^n, t\right) \cong \quad \textit{(cartesian closedness)}$$

$$\int^{c} \mathbf{Sets}\left(s, \sum_{n\in\mathbf{N}} c_n \times a^n\right) \times \prod_{n\in\mathbf{N}} \mathbf{Sets}\left(c_n, b^n \to t\right) \cong \quad \textit{(natural transf. as an end)}$$

$$\int^{c} \mathbf{Sets}\left(s, \sum_{n\in\mathbf{N}} c_n \times a^n\right) \times [\mathsf{Nat}, \mathbf{Sets}]\left(c, b^{(-)} \to t\right) \cong \quad \textit{(Yoneda lemma)}$$

$$\mathbf{Sets}\left(s, \sum_{n\in\mathbf{N}} a^n \times (b^n \to t)\right)$$

This solves the problem posed by *[Milewski, 2017]*.

Unification of optics

All the usual optics are of this form.

Name	Concrete	Action
Adapter	$(s \to a) \times (b \to t)$	$\mathrm{id} \colon [\mathbf{Set}, \mathbf{Set}]$
Lens	$(s \to a) \times (b \times s \to t)$	$(\times) \colon \mathbf{Set} \to [\mathbf{Set}, \mathbf{Set}]$
Prism	$(s \to t + a) \times (b \to t)$	$(+) \colon \mathbf{Set} \to [\mathbf{Set}, \mathbf{Set}]$
Grate	$((s \to a) \to b) \to t$	$(\to) \colon \mathbf{Set}^{op} \to [\mathbf{Set}, \mathbf{Set}]$
Affine Traversal	$s \to t + a \times (b \to t)$	$(\times, +) \colon (\mathbf{Set} \times \mathbf{Set}) \to [\mathbf{Set}, \mathbf{Set}]$
Fixed Traversal	$\Sigma n.s \to (a^n \times (b^n \to t))$	$(\times, \square^n) \colon (\mathbf{Set} \times \mathsf{Nat}) \to [\mathbf{Set}, \mathbf{Set}]$
Traversal	$s \to \Sigma n.a^n \times (b^n \to t)$	$\Sigma_n \colon [\mathsf{Nat}, \mathbf{Set}] \to [\mathbf{Set}, \mathbf{Set}]$
Glass	$((s \to a) \to b) \to s \to t$	$(\times, \to) \colon (\mathbf{Set} \times \mathbf{Set}) \to [\mathbf{Set}, \mathbf{Set}]$
Setter	$(a \to b) \to (s \to t)$	$\mathrm{ev} \colon [\mathbf{Set}, \mathbf{Set}] \to [\mathbf{Set}, \mathbf{Set}]$

In particular, we have new derivations of traversal, fixed traversal, and glass; this expands on previous work by Milewski, Boisseau/Gibbons and Riley.

Preorder on optics

Every action gives a submonoid of endofunctors. Join corresponds to the action of the coproduct (pseudo)monoid. This generalizes the lattice described in Pickerings-Gibbons-Wu.

$$
\begin{tikzcd}
 & \mathrm{Setter}_{(ev)} & \\
 & & \mathrm{Traversal}_{(\mathbf{Poly}(!))} \arrow[ul] \\
\mathbf{Glass}(!)_{(\to,\times)} \arrow[uur] & & \mathrm{Affine}_{(\times,+)} \arrow[u] \\
\mathrm{Grate}_{(\to)} \arrow[u] & \mathrm{Lens}_{(\times)} \arrow[ul] \arrow[ur] & \mathrm{Prism}_{(+)} \arrow[u] \\
 & \mathrm{Adapter}_{(id)} \arrow[ul] \arrow[u] \arrow[ur] &
\end{tikzcd}
$$

Part 3: the Profunctor representation theorem

Promonads and the optics category

A promonad $\psi \in [\mathbf{C}^{op} \times \mathbf{C}, \mathbf{Sets}]$ is a monoid in the 2-category of profunctors.

Lemma (Kleisli construction in Prof, e.g. in Pastro-Street 2008)

The Kleisli object for the promonad, $\mathrm{Kl}(\psi)$, is a category with the same objects, but hom-sets given by the promonad, $\mathrm{Kl}(\psi)(a,b) = \psi(a,b)$.

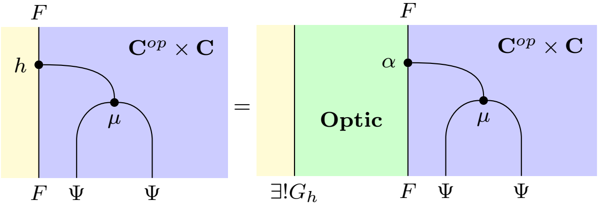
For some fixed kind of optic, we can create a category with the same objects as $\mathbf{C}^{op} \times \mathbf{C}$, but where morphisms are optics of that kind.

$$\psi((s,t),(a,b)) = \int^{c \in \mathbf{M}} \mathbf{C}(s, c \cdot a) \times \mathbf{C}(c \cdot b, t)$$

That is, $\mathbf{Optic} = \mathrm{Kl}(\psi)$.

Craig Pastro and Ross Street. “Doubles for monoidal categories”. In: *Theory and applications of categories* 21.4 (2008), pp. 61–75.

Kleisli object



Theorem

Functors $[\mathbf{Optic}, \mathbf{Set}]$ *are equivalent to right modules on the terminal object for the promonad* $\mathrm{Mod}(\psi)$*, which are algebras for an associated monad.*

This follows from the universal property of the Kleisli object,
$\mathbf{Cat}(\mathbf{Optic}, \mathbf{Set}) \cong \mathbf{Prof}(1, \mathbf{Optic}) \cong \mathbf{Mod}(\psi)$.

Dan Marsden. "Category Theory Using String Diagrams". In: *CoRR* abs/1401.7220 (2014). arXiv: `1401.7220`. URL: `http://arxiv.org/abs/1401.7220`.

Profunctor representation theorem

Theorem (Riley 2018, Boisseau/Gibbons 2018, with a different proof)

Optics given by ψ correspond to parametric functions over profunctors that have module structure over ψ.

$$\int_{p \in \mathbf{Mod}(\psi)} p(a,b) \to p(a,b) \cong \mathbf{Optic}_\psi((s,t),(a,b))$$

Proof. Applying Yoneda lemma again.

$$\begin{aligned}
\int_{p \in \mathbf{Mod}(\psi)} p(a,b) \to p(a,b) &\cong && \textit{(lemma)} \\
\int_{p \in [\mathbf{Optic},\mathbf{Sets}]} p(a,b) \to p(a,b) &\cong && \textit{(by definition)} \\
\mathrm{Nat}(-(a,b), -(s,t)) &\cong && \textit{(Yoneda embedding)} \\
\mathrm{Nat}(\mathrm{Nat}(\mathbf{Optic}((a,b),\Box),-), \mathrm{Nat}(\mathbf{Optic}((s,t),\Box),-)) &\cong && \textit{(Yoneda embedding)} \\
\mathrm{Nat}(\mathbf{Optic}((a,b),\Box), \mathbf{Optic}((s,t),\Box)) &\cong && \textit{(Yoneda embedding)} \\
\mathbf{Optic}((s,t),(a,b)) & &&
\end{aligned}$$

Bartosz Milewski. *Profunctor optics: the categorical view.* https://bartoszmilewski.com/2017/07/07/profunctor-optics-the-categorical-view/. 2017.

Profunctor representation: lenses

Theorem (Profunctor representation theorem)

$$\int_{p \in \mathbf{Mod}(\psi)} p(a,b) \to p(a,b) \cong \mathbf{Optic}_{\psi}((s,t),(a,b))$$

In particular, for lenses, modules associated to the action $(\times)$ are profunctors with a natural transformation

$$\int_{c \in \mathbf{C}} p(a,b) \to p(c \times a, c \times b),$$

which were called cartesian profunctors.

```
-- Haskell definition.
class Cartesian p where
  cartesian :: p a b -> p (c , a) (c , b)
Lens s t a b = (forall p . Cartesian p => p a b -> p s t)
```

Guillaume Boisseau and Jeremy Gibbons. "What you needa know about Yoneda: profunctor optics and the Yoneda lemma (functional pearl)". In: *PACMPL* 2.ICFP (2018), 84:1–84:27. DOI: 10.1145/3236779. URL: https://doi.org/10.1145/3236779.

Profunctor representation: prisms

Theorem (Profunctor representation theorem)

$$\int_{p \in \mathbf{Mod}(\psi)} p(a,b) \to p(a,b) \cong \mathbf{Optic}_{\psi}((s,t),(a,b))$$

In particular, for prisms, modules associated to the action $(+)$ are profunctors with a natural transformation

$$\int_{c \in \mathbf{C}} p(a,b) \to p(c+a, c+b),$$

which were called cocartesian profunctors.

```
-- Haskell definition.
class Cocartesian p where
  cocartesian :: p a b -> p (Either c a) (Either c b)
Prism s t a b = (forall p . Cocartesian p => p a b -> p s t)
```

Guillaume Boisseau and Jeremy Gibbons. "What you needa know about Yoneda: profunctor optics and the Yoneda lemma (functional pearl)". In: *PACMPL* 2.ICFP (2018), 84:1–84:27. DOI: `10.1145/3236779`. URL: `https://doi.org/10.1145/3236779`.

Profunctor representation: traversals

Theorem (Profunctor representation theorem)

$$\int_{p \in \mathbf{Mod}(\psi)} p(a,b) \to p(a,b) \cong \mathbf{Optic}_{\psi}((s,t),(a,b))$$

In particular, for traversals, modules associated to the action $(\sum_n)$ are profunctors with a natural transformation

$$\int_{c \in \mathbf{C}} p(a,b) \to p\left(\sum_n c_n \times a^n, \sum_n c_n \times b^n\right),$$

which we can call analytic profunctors.

Guillaume Boisseau and Jeremy Gibbons. “What you needa know about Yoneda: profunctor optics and the Yoneda lemma (functional pearl)”. In: *PACMPL* 2.ICFP (2018), 84:1–84:27. DOI: `10.1145/3236779`. URL: `https://doi.org/10.1145/3236779`.

Part 4: Further work

Formal verification and constructive proofs

Our proofs are all based in applications of Yoneda lemma and are all constructive. Taking a perspective of mathematics where proofs have a content (proof relevance), we can extract algorithms transforming optics from the formal proofs.

```
lensDerivation {s} {t} {a} {b} =
  begin
   ((∫exists c ∈ Set , ((s -> c × a) × (c × b -> t))))          ≅⟨ ≅-coend (λ c -> trivial) ⟩
   ((∫exists c ∈ Set , (((s -> c) × (s -> a)) × (c × b -> t))))  ≅⟨ ≅-coend (λ c -> trivial) ⟩
   ((∫exists c ∈ Set , ((s -> c) × (s -> a) × (c × b -> t))))    ≅⟨ yoneda ⟩
   ((s -> a) × (s × b -> t))
  qed
```

Figure 3: Derivation of a lens in Agda.

We are using Agda's Instance Resolution algorithm to reconstruct the formal proof from these hints.

Summary of results

- Optics: a zoo of accessors used by programmers *[Kmett, lens library, 2012]*.
  - We have a definition that captures all of them *[Riley, 2018]*.
  - We give a new derivation of Traversal as the optic for analytic functors.
  - We give a description of the fixed Traversal.
- Profunctor optics, equivalence: for Tambara *[Pastro/Street, 2008]*, *[Milewski, 2017]* and endofunctors *[Riley]*, *[Boisseau/Gibbons]*.
  - We provide a new proof for $[\mathbf{Optic}, \mathbf{Set}] \cong \mathbf{Mod}$ from general principles in 2-category theory.
  - With this, we can directly extend the proof of *[Pastro/Street, 2008]* to any arbitrary action (same result in *[Boisseau/Gibbons]* with a different proof technique).
- Composition of optics: lattice described in *[Pickering/Gibbons/Wu, 2016]*.
  - We construct the optics that arise by composition using coproducts of the actions.
  - We get the Affine traversal as in *[Boisseau/Gibbons]* as a particular case.
  - We get a new optic composing Lenses and Grates.
- Formal verification: development of a library of optics in Agda.
  - We formally verify proofs of equivalence.
  - We automate reasoning with isomorphisms in $\mathbf{Sets}$.
  - We extract the translation algorithms from the formal proofs.

Further work

- Generalizations: in which other settings do this theorems apply?
    - Our proof works over any enrichment. Study optics over other enrichments.
    - In fact, this seems to work for any pseudomonoid. Can we do a formal theory of optics for categories other than $\mathbf{Cat}$?
    - Consider unidirectional optics, everything that works for $\mathbf{C}^{op} \times \mathbf{C}$ works also for just $\mathbf{C}$.
- Simplify the theory with categories: our proofs should be as simple as possible.
    - We almost exclusively rely on Yoneda and definitions.
    - Simpler proofs mean simpler formalizations and simpler implementations.
- Other directions:
    - Teleological categories *[Hedges, 2019]* and their relations to optics.
    - Van Laarhoven representations *[Van Laarhoven, 2009]* and study the connection in *[Riley, 2018]*.

- Applications: which optics are useful to programmers?
    - Once the framework has been established, it should be easier to come up with new optics.
    - Develop a formal library of optics in Agda.