Profunctor optics and traversals

Mario Román

University of Oxford

A thesis submitted for the degree of

MSc in Mathematics and Foundations of Computer Science

Trinity 2019

Contents

Abstract

Optics are bidirectional accessors of data structures. They provide a powerful abstraction of many common data transformations. This abstraction is compositional thanks to a representation in terms of profunctors endowed with an algebraic structure called *Tambara module* [Mil17].

There exists a general definition of optic [Boi17, Ril18] in terms of coends that, after some elementary application of the Yoneda lemma, particularizes in each one of the basic optics. *Traversals* used to be the exception; we show an elementary derivation of traversals and discuss some other new derivations for optics. We relate our characterization of traversals to the previous ones showing that the coalgebras of a comonad that represents and split into shape and contents are traversable functors.

The representation of optics in terms of profunctors has many different proofs in the literature; we discuss two ways of proving it, generalizing both to the case of mixed optics for an arbitrary action. Categories of optics can be seen as Eilenberg-Moore categories for a monad described by Pastro and Street [PS08]. This gives us two different approaches to composition between profunctor optics of different *families*: using distributive laws between the monads defining them, and using coproducts of monads. The second one is the one implicitly used in Haskell programming; but we show that a refinement of the notion of optic is required in order to model it faithfully.

We provide experimental implementations of a library of optics in Haskell and partial Agda formalizations of the profunctor representation theorem.

Acknowledgements

I am first grateful to my dissertation supervisor Jeremy Gibbons for generously taking time to make this dissertation possible, for the advice, patience, ideas, pointers to the literature, and all the suggestions that made it readable. His work both on optics [BG18, PGW17] and traversable functors [GdSO09] was many times, and sometimes in unexpected ways, an inspiration for this text.

My study of optics has been part of a joint project with a wonderful group of people. I want to thank them all for their ideas, encouragement and kindness. Coordinating conversations on optics over four different timezones was not easy, and they put a lot of effort into making that happen. Bartosz Milewski crafted a theory [Mil17], posed a nice problem, and then wholeheartedly shared with us all his intuitions and insights. Derek Elkins put time, wisdom and patience into guiding me on how to transform ideas into actual mathematics. Bryce Clarke and Emily Pillmore were fantastic colleagues during those days, and discussing both category theory and the intricacies of Haskell with them was particularly stimulating. I learned the art of *coend-fu* through the lucid teachings of Fosco Loregian [Lor15], who also took the time to clarify many of my doubts. Daniel Cicala, Jules Hedges and Destiny Chen organized the ACT School and made this collaboration possible in the first place. Thanks also go to Guillaume Boisseau, Fatimah Ahmadi, Giovanni De Felice, Dan Marsden, Bob Coecke, David Reutter, David J. Myers, Brendan Fong, Jamie Vicary, and Carmen Constantin for their engaging teachings on category theory and/or conversations during the writing of this dissertation. The text relies on Dan Marsden's macros for string diagrams. I thank Sam Staton and Paul Blain Levy for corrections and comments on the earlier versions of this dissertation.

On a more personal note, friends and family in Granada have been supporting me for years. Besides them, I would like to thank Andrés, Daphne, Joel, Jean and Lewis, who made enjoyable my day-to-day life during the MFoCS. Special thanks go to Elena for all her affection and for making me recover the passion for mathematics and category theory. A. P. A. E. and her made the final days of writing this dissertation cozy. Grazie mille.

Chapter 1

Introduction

1.1 Background and scope

In programming, **optics** are a compositional representation of data accessors provided by libraries such as Kmett's [Kme18]. Code needs to deal with complex and nested data structures: *records* with multiple fields, *union types* with multiple alternative contents, *containers* with lists of elements inside, and many other similar examples. In all of these cases, we want to be able to focus on some internal part and access it *bidirectionally*, that is, we want to be able to read it, but also to update it with new contents, propagating the changes to the bigger data structure. The most obvious pattern of this kind is a *lens*: a data accessor for the particular subfield of some data structure.

1.1.1 Lenses

Definition 1.1.1. In a cartesian category $\mathbf{C}$, a **lens** from a pair of objects (s, t) with *focus* in a pair of objects (a, b) is given by two morphisms: a *view* $\mathbf{C}(s, a)$, representing the reading operation, and an *update* $\mathbf{C}(s \times b, t)$, representing the writing operation.

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

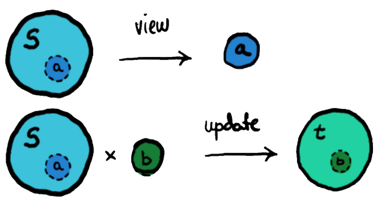


Figure 1.1: In **Sets**, a big data structure s contains a subfield a and the first function $(s \to a)$ provides a way of accessing it. The second function $(s \times b \to t)$ plugs something of type b in the place of a, updating s and getting a new complex structure of type t.

Example 1.1.2. Suppose we have a data structure representing a *postal address*, with the *ZIP code* being a subfield. We want to be able to view the ZIP code inside an address and to modify it inside the bigger address. This can be encoded as an element of $\mathbf{Lens}((\mathbf{Postal}, \mathbf{Postal}), (\mathbf{Zip}, \mathbf{Zip}))$, which is in turn given by two functions with the following signature.

$$\begin{aligned} \text{viewZipCode}\colon \mathbf{Postal} &\to \mathbf{Zip} \\ \text{updateZipCode}\colon \mathbf{Postal} \times \mathbf{Zip} &\to \mathbf{Postal} \end{aligned}$$

Remark 1.1.3. This definition leaves open the possibility of updating a subfield of type a in a bigger data structure s with an element of a different type b, yielding a structure of a new type t. The reader not interested in changing types can just consider optics from (s, s) to (a, a); these are sometimes called *monomorphic lenses* (for instance, in [Hed18]) or *simple lenses* in [Kme18].

Lenses were first described in categorical terms in Oles' thesis [Ole82]. They are considered as data accessors in [PGW17]. They have been also used in *compositional game theory*, as described in [GHWZ18], and in *supervised learning*, as described in [FJ19].

1.1.2 Prisms

We can consider, however, other patterns for data accessing: *lenses are optics, but not all optics are lenses! Prisms* give a second data accessing pattern whose focusing deals with alternatives.

Definition 1.1.4. In a cocartesian category $\mathbf{C}$, a **prism** from a pair of sets (s, t) with focus in (a, b) is given by two functions: a *match* $s \to t + a$ representing pattern-matching on a, and a *build* $b \to t$ representing the construction of the data structure from one of the alternatives.

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

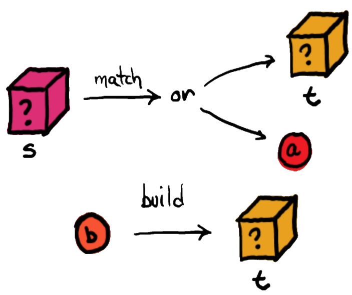


Figure 1.2: An abstract data structure s could have different internal structures. In particular, we can try to see if it is of type a and, in case of failure, return something of type t. The build function takes some concrete data structure b to some more abstract structure t.

Example 1.1.5. Suppose we have a string we want to parse as an *address*. That address could be in particular a postal address like in Example 1.1.2 and in that case we want to return a specific type capturing this. That is, we can build an address from a postal address; and we can try to fit an address into a postal address. This can be encoded as an element of $\mathbf{Prism}((\mathbf{Postal}, \mathbf{Postal}), (\mathbf{String}, \mathbf{Address}))$, which is, in turn, given by two functions with the following signature.

$$\begin{aligned} \text{matchAddress}\colon \mathbf{String} &\to \mathbf{Address} + \mathbf{Postal} \\ \text{buildAddress}\colon \mathbf{Postal} &\to \mathbf{Address} \end{aligned}$$

Remark 1.1.6. Prisms are dual to lenses. A prism is precisely a lens in the opposite category. As we will see later, they are to the coproduct what lenses are to the product.

1.1.3 Traversals

We can go further: optics do not necessarily need to deal with a single focus. Traversals are optics that access a list of foci at the same time.

Definition 1.1.7. In a cartesian closed category with natural number-indexed limits and colimits $\mathbf{C}$, a **traversal** from a pair of sets (s, t) with focus in (a, b) is given by a single function: an *extract* $\mathbf{C}(s, \sum_{n \in \mathbb{N}} a^n \times (b^n \to t))$ that represents extracting a list of some length and a function that takes a list of the same size to create a new structure.

$$\mathbf{Traversal}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} s \\ t \end{pmatrix}\right) = \mathbf{C}\left(s, \sum\nolimits_{n \in \mathbb{N}} a^n \times (b^n \to t)\right).$$

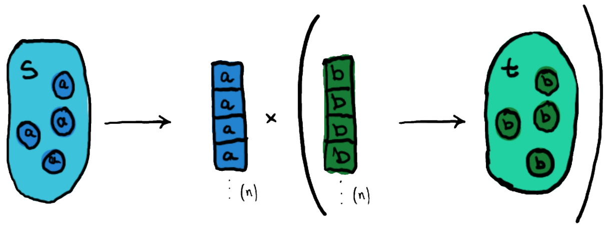


Figure 1.3: From a complex data structure s we can extract (1) some list of values of type a of length n for some $n \in \mathbb{N}$, and (2) a way of updating this structure with a new list of length n.

Example 1.1.8. Suppose we have now a *mailing list* containing multiple email addresses associated to other data, such as names or subscription options. An accessor for the email addresses is an element of $\mathbf{Traversal}((\mathbf{MailList}, \mathbf{MailList}), (\mathbf{Email}, \mathbf{Email}))$, which is given by a single function with the following signature.

$$\text{extract}\colon \mathbf{MailList} \to \sum_{n \in \mathbb{N}} \mathbf{Email}^n \times (\mathbf{Email}^n \to \mathbf{MailList}).$$

Remark 1.1.9. One could think that the traversal can be rewritten in the same style as a lens: that is, as two functions $s \to a^n$ and $s \times b^n \to t$. The fact that the two $n \in \mathbb{N}$ need to be the same prevents this split. This is also what makes a traversal fundamentally different from a lens focusing on a list or tuple type. For the traversal, the length of the output can vary, but we need a list of the same length to be able to update.

1.1.4 The problem of modularity

A problem arises when accessing compound data structures. The focus of an optic can be itself a complex data structure in which we can make use of an optic again. We would expect optics to behave *compositionally*, in the sense that it should be straightforward to give a description of the optic that, from the bigger data structure, applies two optics to focus on the innermost subfield.

Example 1.1.10. More concretely, consider the lens in Example 1.1.2 and the prism in Example 1.1.5. We would like to have a composite optic that tries to access the Zip code from any arbitrary string. This problem can be formulated and solved for any *lens* and *prism*, but there is no obvious unified way to solve it for any pair of optics. Moreover, note that the resulting accessor need be neither a lens nor a prism.

Explicitly, in a bicartesian closed category, assume we have a prism from (s,t) to (a,b) given by $m \in \mathbf{C}(s, t+a)$ and $l \in \mathbf{C}(b,t)$; and a lens (a,b) to (x,y) given by $v \in \mathbf{C}(a,x)$ and $u \in \mathbf{C}(a \times y, b)$. We want to compose them into a combined optic from (s,t) to (x,y). A suitable composition is the following, although it is tedious and not particularly illuminating to write it down. Here, we write Λ for the product-exponential adjunction.

$$\mathbf{m} \circ [\mathrm{id}_t, \mathbf{v} \times \Lambda(\mathbf{b} \circ \mathbf{u})] \in \mathbf{C}(s, t + x \times (y \to t)).$$

Note that $\Lambda(l \circ u) \in \mathbf{C}(a, y \to t)$, and the rest is just a combination of morphisms that uses the product and the coproduct. This naive approach to composition is not practical: we would need to write down a composition for every pair of optics, and its output could be each time a completely new kind of optic. We need a general way of solving the problem of composition of optics.

1.1.5 Solving modularity

The problem has been solved in programming libraries such as [HFM+19] using **profunctor optics**. Perhaps surprisingly, some optics turn out to be equivalent to parametric functions over certain families of profunctors. For instance, we can prove an isomorphism between lenses $\mathbf{C}(s,a) \times \mathbf{C}(s \times b, t)$ and families of morphisms $p(a,b) \to p(s,t)$ for profunctors p of a certain class that we will describe later. This provides a solution for the composition of optics: composing an optic from (x,y) to (a,b) with an optic from (a,b) to (s,t) becomes function composition of the form $p(x,y) \to p(a,b) \to p(s,t)$.

In summary, the profunctor representation of optics transforms all the complicated cases of composition of optics into *just ordinary function composition*. This technique is documented originally in [PGW17]. We will justify in the following sections why something like this works in terms of category theory and look into particular cases.

1.2 Outline

- **Chapter 2** introduces main results that are not specific to the theory of optics. It starts with a brief summary of the (co)end calculus as in [Lor15], underlining the importance of the Yoneda lemma and of Kan extensions. It describes the formal theory of monoids capturing also the multiple uses of monads and monoidal categories that we will encounter in this text. We take a moment to describe replete subcategories and how they provide a well-behaved notion of 2-categorical image of a functor. Finally, we describe the bicategory of profunctors, promonads and Kleisli categories.
- **Chapter 3** provides the definition of optic and shows that it actually captures the examples we were interested in. Apart from these main ones, it uses the definition to provide new examples of optic.
- **Chapter 4** is centered around traversals, a particular kind of optic for which we give two different derivations. We relate both describing traversals as coalgebras for a comonad that represents a split into shape and contents. We will show how they

are related to flat combinatorial species and how to use this to construct yet another new optic.

- **Chapter 5** introduces and proves the main result of the theory of optics: the *profunctor representation theorem* (Theorem 5.3.1), that links every optic to a profunctor optic in an uniform way. We also describe optics both as Kleisli categories for a monad and Kleisli objects for a promonad.
- **Chapter 6** discusses how function composition of profunctor optics works. The first part is motivated by how monads can be composed when a suitable distributive law between them exists. The second part is motivated by how Haskell joins constraints on parametric functions to compose profunctor optics. These two *compositions* do not necessarily give the same results, but we can reconcile them considering some subclass of well-behaved optics that we call *clear optics*.
- **Chapter 7** concerns applications. In particular, we have built a minimalistic optics library in Haskell that directly implements the profunctor representation theorem. We also formalize the main optic derivations in the Agda proof assistant, showing that the approach based on Yoneda is particularly suited for formal verification.

1.3 Contributions

Parts of this dissertation started as a joint project; this section lists the original contributions of the author of this dissertation and gives acknowledgement for the rest.

- **Chapter 2** comes with no claim of originality apart from the presentation of the ideas. The propositions and proofs can be found in the literature and we put special care into acknowledge them. Fosco Loregian helped me understand the coproduct of monoidal actions.
- In **Chapter 3**, I slightly generalize Mitchell Riley's construction of the category of optics [Ril18] in a direction already proposed there. I derive traversals as the optic for power series functors, solving a problem posed by Milewski [Mil17]; Bartosz Milewski suggested to me that the traversal was possibly an optic for an action involving multiple products or exponentials. In §3.4, I define and derive concrete forms of *Kaleidoscopes* and *Algebraic lenses*, optics not present on the literature to the best of my knowledge; Emily Pillmore helped me collect definitions of optics from Haskell libraries. Guillaume Boisseau let me know about the *achromatic lens*. I prove that the *generalized lens* is a mixed optic; this notion was introduced to me by David Jaz Myers. I present a mechanism for getting optics for (co)free, which is a specialization of the coalgebraic optics of [Ril18], both requiring more hypothesis and getting a stronger result.
- In **Chapter 4**, I prove in §4.1 that traversals are coalgebras for a *shape-contents* comonad that I define there; a related result using parameterized comonads can be found in [JO15]. In Lemma 4.1.3, I prove that the natural family of transformations defining a traversal is given by any coalgebra for the *shape-contents* functor, trying to simplify the approach in [JO15]. Using there that ends distribute over discrete colimits to complete the derivation was suggested to me by Fosco Loregian and Derek Elkins. I connect in this way the derivation using traversables that could be found in Riley's [Ril18] to the derivation using power series functors. I describe the *unsorted traversal* and derive a concrete form for it. In the related Appendix 8.2, I construct cofree traversables.
- **Chapter 5** is mostly an expository chapter. I make explicit the proof that can

be found in [PS08] of the profunctor representation theorem, extending it to mixed optics over arbitrary actions; a proof of the same result with a different technique can be found in [Ril18] and [BG18].

- In **Chapter 6**, I study distributive laws for Pastro-Street comonads and propose a composition of optics based on Kleisli categories. I construct *glasses* with this technique and I show that *affine traversals* can be also obtained with it. In §6.3, I study how composition of optics works in Haskell and provide a categorical description in terms of coproduct actions. The fact that a naive composition of Tambara modules does not work as one would wish in this case was pointed to me by Bryce Clarke. I discuss the need for a restricted definition of optic in order to recover the lattice of optics and I propose a definition of *clear optics*, addressing this problem. I prove that some of the common optics are clear and that the composition of profunctor *lenses* and *prisms* is precisely an *affine traversal* in this setting.
- In **Chapter 7**, I present a library of optics in Haskell, a formalization of some (co)end derivations in Agda, and an example usage of the *kaleidoscope*.

Chapter 2

Preliminaries

2.1 (co)End calculus

Our basic tool will be (co)end calculus as described by Loregian [Lor15]. For completeness, we replicate here the main definitions and results we are going to use.

2.1.1 (co)Ends

The intuition is that *ends* should be a sort of universal quantifier over the objects of a category plus some *naturality* conditions; whereas *coends* can be thought of as their existential version. In fact, when encoding optics in a programming language that provides parametricity and existential types, these are used in place of coends, expecting that the syntax of the language will enforce their naturality conditions. (co)Ends provide a rich calculus based on the Yoneda lemma that we will use throughout this text, being the basic building block of many of our proofs. A description on how to turn coends into a calculus that can be used in theorem provers is described in [CW01].

Definition 2.1.1. The **end** of a functor $p\colon \mathbf{C}^{op}\times\mathbf{C} \to \mathbf{D}$ is the universal object $\int_{c\in\mathbf{C}} p(c,c)$ endowed with morphisms $\pi_a\colon \left(\int_{c\in\mathbf{C}} p(c,c)\right) \to p(a,a)$ for every $a \in \mathbf{C}$ such that, for any morphism $f\colon a \to b$ in $\mathbf{C}$, they satisfy $p(\mathrm{id}, f) \circ \pi_a = p(f, \mathrm{id}) \circ \pi_b$. It is universal in the sense that any other object d endowed with morphisms $j_a\colon d \to p(a,a)$ satisfying the same condition factors uniquely through it.

$$\begin{tikzcd}
 & d \arrow[d, dashed, "\exists!"] \arrow[ddl, bend right, "j_a"'] \arrow[ddr, bend left, "j_b"] & \\
 & \int_c p(c,c) \arrow[dl, "\pi_a"'] \arrow[dr, "\pi_b"] & \\
p(a,a) \arrow[dr, "p(\mathrm{id},f)"'] & & p(b,b) \arrow[dl, "p(f,\mathrm{id})"] \\
 & p(a,b) &
\end{tikzcd}$$

In other words, the end is the equalizer of the action of morphisms on both arguments of the functor.

$$\int_{c\in\mathbf{C}} p(c,c) \cong \mathrm{eq}\left(\prod_{c\in\mathbf{C}} p(c,c) \rightrightarrows \prod_{f\colon a\to b} p(a,b) \right)$$

Definition 2.1.2. Dually, the **coend** of a functor $p\colon \mathbf{C}^{op}\times\mathbf{C}\to\mathbf{D}$ is the universal object $\int^{c\in\mathbf{C}} p(c,c)$ endowed with morphisms $i_a\colon p(a,a)\to \left(\int^{c\in\mathbf{C}} p(c,c)\right)$ for every $a\in\mathbf{C}$ such that, for any morphism $f\colon b\to a$ in $\mathbf{C}$, they satisfy $i_b\circ p(f,\mathrm{id}) = i_a\circ p(\mathrm{id},f)$. It is universal in the sense that any other object d endowed with morphisms $j_a\colon p(a,a)\to d$ satisfying the same condition factors uniquely through it.

$$\begin{tikzcd}
 & p(a,b) \arrow[dl, "p(\mathrm{id},f)"'] \arrow[dr, "p(f,\mathrm{id})"] & \\
p(a,a) \arrow[dr, "i_a"] \arrow[ddr, bend right, "j_a"'] & & p(b,b) \arrow[dl, "i_b"'] \arrow[ddl, bend left, "j_b"] \\
 & \int^c p(c,c) \arrow[d, dashed, "\exists!"] & \\
 & d &
\end{tikzcd}$$

In other words, the coend is the coequalizer of the action on morphisms on both arguments of the functor.

$$\int^{c\in\mathbf{C}} p(c,c) \cong \mathrm{coeq}\left(\bigsqcup_{f\colon b\to a} p(a,b) \rightrightarrows \bigsqcup_{x\in\mathbf{C}} p(x,x) \right)$$

Remark 2.1.3. (Co)ends are particular cases of (co)limits. As such, they are unique up to isomorphism when they exist. (Co)continuous functors preserve them and, in particular, the co(ntra)variant hom-functor commutes with them. For any $p\colon \mathbf{C}^{op}\times\mathbf{C}\to\mathbf{D}$ and every $d\in\mathbf{D}$, there exist canonical isomorphisms with the following signatures.

$$\mathbf{D}\left(\int^{c\in\mathbf{C}} p(c,c), d\right) \cong \int_{c\in\mathbf{C}} \mathbf{D}(p(c,c),d), \qquad \mathbf{D}\left(d, \int_{c\in\mathbf{C}} p(c,c)\right) \cong \int_{c\in\mathbf{C}} \mathbf{D}(d,p(c,c)).$$

Note also that all (co)ends exist in (co)complete categories.

Proposition 2.1.4 ([Lor15], Remark 1.4)**.** *(co)Ends are functorial. We can define a functor* $[\mathbf{C}^{op}\times\mathbf{C},\mathbf{D}]\to\mathbf{D}$ *that sends a profunctor* $p\colon \mathbf{C}^{op}\times\mathbf{C}\to\mathbf{D}$ *to its end* $\int_c p(c,c)\in\mathbf{D}$.

Proof. Let $\eta\colon p\Rightarrow q$ be a natural transformation between two profunctors. By the universal property of the ends, we have an induced $\eta_*\colon \int_c p(c,c)\to\int_c q(c,c)$ constructed as the unique morphism making the following diagram commute for any $x,x'\in\mathbf{C}$ and any

$f\colon x \to x'$.

$$\begin{tikzcd}
 & \int_c q(c,c) \arrow[rr, "\pi_{x'}"] \arrow[dd, "\pi_x"] & & q(x',x') \arrow[dd, "q(f,\mathrm{id})"] \\
\int_c p(c,c) \arrow[ur, dashed, "\exists!\eta_*"] \arrow[rr, "\pi_{x'}"] \arrow[dd, "\pi_x"'] & & p(x',x') \arrow[ur, "\eta_{x',x'}"] \arrow[dd, "p(f,\mathrm{id})"] & \\
 & q(x,x) \arrow[rr, "q(\mathrm{id},f)"] & & q(x,x') \\
p(x,x) \arrow[ur, "\eta_{x,x}"] \arrow[rr, "p(\mathrm{id},f)"'] & & p(x,x') \arrow[ur, "\eta_{x,x'}"'] &
\end{tikzcd}$$

We can check that the identity makes the diagram commute for the identity natural transformation, getting $\mathrm{Id}_* = \mathrm{id}$; and that composition is preserved because it makes the composite diagram commute, getting $(\eta \circ \sigma)_* = \eta_* \circ \sigma_*$ for any pair of natural transformations $\eta\colon p \Rightarrow q$ and $\sigma\colon r \Rightarrow p$. $\square$

2.1.2 Fubini rule

Writing ends and coends as integrals suggests the following sort of Fubini rule. Note however that, contrary to what happens with the theorem of classical analysis, this rule can be *always* applied.

Lemma 2.1.5 (Fubini rule for ends, [Lor15, §1, Exercise 10])**.** *Let $p\colon \mathbf{C}^{op} \times \mathbf{C} \times \mathbf{D}^{op} \times \mathbf{D} \to \mathbf{E}$ be a functor. We can consider the following three ends, that are different in principle.*

- *Identify $\mathbf{C}^{op} \times \mathbf{C} \times \mathbf{D}^{op} \times \mathbf{D} \to \mathbf{E}$ with $\mathbf{C}^{op} \times \mathbf{C} \to (\mathbf{D}^{op} \times \mathbf{D} \to \mathbf{E})$. Take the end over it to obtain $\int_c p(c,c,-,-)\colon \mathbf{D}^{op} \times \mathbf{D} \to \mathbf{E}$, and then the end over the resulting functor to get $\int_d \int_c p(c,c,d,d)$.*
- *Identify $\mathbf{C}^{op} \times \mathbf{C} \times \mathbf{D}^{op} \times \mathbf{D} \to \mathbf{E}$ with $\mathbf{D}^{op} \times \mathbf{D} \to (\mathbf{C}^{op} \times \mathbf{C} \to \mathbf{E})$. Take the end over it to obtain $\int_d p(-,-,d,d)\colon \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{E}$, and then the end over the resulting functor to get $\int_c \int_d p(c,c,d,d)$.*
- *Identify $\mathbf{C}^{op} \times \mathbf{C} \times \mathbf{D}^{op} \times \mathbf{D} \to \mathbf{E}$ with $(\mathbf{C} \times \mathbf{D})^{op} \times (\mathbf{C} \times \mathbf{D}) \to \mathbf{E}$. Take the end over this product category to get $\int_{(c,d)} p(c,c,d,d)$.*

*The **Fubini rule** for coends states the following isomorphisms.*

$$\int_{(c,d)\in\mathbf{C}\times\mathbf{D}} p(c,c,d,d) \cong \int_{c\in\mathbf{C}} \int_{d\in\mathbf{D}} p(c,c,d,d) \cong \int_{d\in\mathbf{D}} \int_{c\in\mathbf{C}} p(c,c,d,d).$$

Proof. We will just prove that $\int_{(c,d)\in\mathbf{C}\times\mathbf{D}} p(c,c,d,d) \cong \int_{c\in\mathbf{C}} \int_{d\in\mathbf{D}} p(c,c,d,d)$, the other isomorphism is similar. First, we construct the morphism from the left hand side to the right hand side. We start by constructing a family of morphisms $\sigma_x\colon \int_{(c,d)\in\mathbf{C}\times\mathbf{D}} p(c,c,d,d) \to \int_{d\in\mathbf{D}} p(x,x,d,d)$ as the unique ones making the following diagram commute for any $g\colon y \to y'$. Note that the external square commutes as a particular case of dinaturality of the coend over the product.

$$\begin{tikzcd}
& \int_{c,d} p(c,c,d,d) \arrow[dl, "\pi_{x,y}"'] \arrow[d, dashed, "\exists!\sigma_x"] \arrow[dr, "\pi_{x,y'}"] & \\
p(x,x,y,y) \arrow[dr, "{p(-,-,-,g)}"'] & \int_d p(x,x,d,d) \arrow[l, "\pi_y"] \arrow[r, "\pi_{y'}"'] & p(x,x,y',y') \arrow[dl, "{p(-,-,g,-)}"] \\
& p(x,x,y,y') &
\end{tikzcd}$$

We need to prove that the family σ_x is actually dinatural in x. This comes also as a particular case of the dinaturality of the end over the product, as the following diagram shows.

$$\begin{tikzcd}
& \int_{c,d} p(c,c,d,d) \arrow[dl, "\sigma_x"'] \arrow[dr, "\sigma_{x'}"] \arrow[ddl, bend right, "\pi_{x,y}"] \arrow[ddr, bend left, "\pi_{x',y'}"'] & \\
\int_d p(x,x,d,d) \arrow[d, "\pi_x"'] \arrow[dr, "f_*"] & & \int_d p(x',x',d,d) \arrow[d, "\pi_{x'}"] \arrow[dl, "f_*"'] \\
p(x,x,y,y) \arrow[dr, "{p(f,-,-,-)}"'] & \int_d p(x',x,d,d) \arrow[d, "\pi_y"] & p(x',x,y,y) \arrow[dl, "{p(-,f,-,-)}"] \\
& p(x',x,y,y) &
\end{tikzcd}$$

Secondly, we construct a morphism from the right hand side to the left hand side using the projections. The following diagram shows dinaturality for any $f\colon x \to x'$ and any $g\colon y \to y'$, which amounts to saying that it is dinatural for any $(f,g)\colon (x,y) \to (x',y')$. Commutativity for this diagram uses that $\pi_y\colon \int_d p(-,-,d,d) \Rightarrow p(-,-,y,y)$ must be given by a natural transformation for any $y \in \mathbf{D}$ because we are considering an end over a category of functors.

$$\begin{tikzcd}
& & & \int_c \int_d p(c,c,d,d) \arrow[dll, "\pi_x"'] \arrow[drr, "\pi_{x'}"] & & & \\
& \int_d p(x,x,d,d) \arrow[ddl, "\pi_y"'] \arrow[drr, "f"] & & & & \int_d p(x',x',d,d) \arrow[dll, "f"'] \arrow[ddr, "\pi_{y'}"] & \\
& & & \int_d p(x,x',d,d) \arrow[ddl, "\pi_y"'] \arrow[ddr, "\pi_{y'}"] & & & \\
p(x,x,y,y) \arrow[drr, "f"] \arrow[ddr, "g"'] & & & & & & p(x',x',y',y') \arrow[dll, "f"'] \arrow[ddl, "g"] \\
& & p(x,x',y,y) \arrow[ddr, "g"] & & p(x,x',y',y') \arrow[ddl, "g"'] & & \\
& p(x,x,y,y') \arrow[drr, "f"] & & & & p(x',x',y,y) \arrow[dll, "f"'] & \\
& & & p(x,x',y,y') & & &
\end{tikzcd}$$

$\square$

Remark 2.1.6. It can be shown that (co)ends over profunctors that are mute in the contravariant variable are canonically isomorphic to (co)limits. As a consquence, the Fubini rule is also valid for limits. Given any $F\colon I \times J \to \mathbf{C}$, we have the following isomorphisms.

$$\mathsf{lim}_I \mathsf{lim}_J F \cong \mathsf{lim}_{I\times J} F \cong \mathsf{lim}_J \mathsf{lim}_I F$$

In particular, ends distribute over products.

2.1.3 Yoneda lemma

We will now partially justify the intuition that ends are universal quantifiers plus some naturality conditions showing that natural transformations are particular cases of ends. This motivates a rewrite of the Yoneda lemma in terms of (co)ends.

Proposition 2.1.7. *The set of natural transformations between two functors $F, G\colon \mathbf{C} \to \mathbf{D}$ is given by the following end.*

$$\mathrm{Nat}(F,G) = \int_{c\in\mathbf{C}} \mathbf{D}(Fc, Gc).$$

Proof. A natural transformation $\alpha\colon F \Rightarrow G$ is equivalently a family of morphisms $\alpha_c\colon 1 \to \mathbf{D}(Fc, Gc)$ indexed by $c \in \mathbf{C}$ and such that, for any $f\colon c \to d$, it holds that $\alpha_c \circ \mathbf{D}(\mathrm{id}, f) = \mathbf{D}(f, \mathrm{id}) \circ \alpha_d$. This means that the elements of the set that the end defines, $1 \to \int_{c\in\mathbf{C}} \mathbf{D}(Fc, Gc)$, are precisely the natural transformations $F \Rightarrow G$. □

Remark 2.1.8. In the case of **Sets**, we will allow ourselves to sometimes write the internal hom as $(a \to b) \cong \mathbf{Sets}(a, b)$. This makes many (co)end derivations less noisy and closely follows the arrow notation of some functional programming languages. Natural transformations between copresheaves $F, G\colon \mathbf{C} \to \mathbf{Sets}$ can be written as

$$\mathrm{Nat}(F,G) = \int_{c\in\mathbf{C}} Fc \to Gc.$$

In the enriched case, the same could be said for any cartesian Benabou cosmos $\mathcal{V}$, taking it to be enriched over itself.

In an arbitrary category $\mathbf{C}$, consider the representable functors $\mathbf{C}(-, a)$ and $\mathbf{C}(a, -)$ for some $a \in \mathbf{C}$. In the cases where we want to keep the category $\mathbf{C}$ implicit, we will use the hiragana "yo" to write them as $\text{よ}^a\colon \mathbf{C}^{op} \to \mathbf{Sets}$ and $\text{よ}_a\colon \mathbf{C} \to \mathbf{Sets}$, respectively. This notation is inspired both from the one used in [Lor15] and from the main role the *Yoneda lemma* will play in this text. The Yoneda lemma says that, for any copresheaf $F\colon \mathbf{C} \to \mathbf{Sets}$ and any $c \in \mathbf{C}$, the set of natural transformations $\mathrm{Nat}(\text{よ}_a, F)$ is in bijection with the set Fa. The bijection is natural in both F and a, and it is constructed from the fact that any natural transformation is uniquely determined by the image of $\mathrm{id} \in \text{よ}_a(a)$.

Using (co)ends, we can rephrase this result as the fact that every (co)presheaf can be written as a (co)end. This is called the *ninja Yoneda lemma* after a comment by T. Leinster [TLu].

Lemma 2.1.9 (Ninja Yoneda lemma)**.** *For any functor $F\colon \mathbf{C} \to \mathbf{Sets}$, we have canonical isomorphisms*

$$Fa \cong \int_{c\in\mathbf{C}} \text{よ}_a c \to Fc, \qquad Fa \cong \int^{c\in\mathbf{C}} Fc \times \text{よ}^a c,$$

which we call the Yoneda reduction *and* Coyoneda reduction*.*

Proof. The Yoneda reduction follows from the usual statement of the Yoneda lemma, knowing that natural transformations can be written as ends as in Proposition 2.1.7. We will prove the Coyoneda reduction from the Yoneda reduction on the opposite category. Take an arbitrary set $s \in \mathbf{Sets}$, and consider the following chain of natural isomorphisms.

$$\begin{aligned}
&\mathbf{Sets}\left(\int^{c\in\mathbf{C}} Fc \times よ^a c \ , \ s\right) \\
\cong &\qquad \text{(Continuity)} \\
&\int_{c\in\mathbf{C}} \mathbf{Sets}(Fc \times よ^a c, s) \\
\cong &\qquad \text{(Exponential)} \\
&\int_{c\in\mathbf{C}} \mathbf{Sets}(よ^a c, \mathbf{Sets}(Fc, s)) \\
\cong &\qquad \text{(Yoneda reduction in } \mathbf{C}^{op}) \\
&\mathbf{Sets}(Fa, s).
\end{aligned}$$

Because of (the usual) Yoneda lemma, this means that Fa is isomorphic to $\int^{c\in\mathbf{C}} Fc \times よ^a c$. □

Remark 2.1.10. An intuition on the Yoneda lemma (see [Lor15] Remark 2.6) is that it allows us to integrate interpreting the よ as a Dirac's delta for ends.

2.1.4 Kan extensions

In this text we will be working with *global* Kan extensions that arise as adjoints to functor precomposition. However, *local* Kan extensions with their usual definition can exist in particular cases even if the adjunction that we are using to define them does not. We refer, for instance, to Chapter 6 of [Rie17] for the more usual definition and study of *local* Kan extensions.

Definition 2.1.11. The *left* and *right* **Kan extensions** along a functor $F : \mathbf{C} \to \mathbf{D}$ are the left and right adjoints, respectively, to the functor given by precomposition $(- \circ F) : [\mathbf{D}, \mathbf{Sets}] \to [\mathbf{C}, \mathbf{Sets}]$. In other words, we have the following natural isomorphisms.

$$[\mathbf{D}, \mathbf{Sets}](\mathsf{Lan}_F G, H) \cong [\mathbf{C}, \mathbf{Sets}](G, H \circ F).$$
$$[\mathbf{C}, \mathbf{Sets}](H \circ F, G) \cong [\mathbf{D}, \mathbf{Sets}](H, \mathsf{Ran}_F G).$$

We write Lan_F for the left Kan extension along F and Ran_F for the right Kan extension along F. We say that $\mathsf{Lan}_F G$ and $\mathsf{Ran}_F G$ are the *left/right* Kan extensions of G *along* F.

In some specially well-behaved categories, and particularly in **Sets**, the left and right Kan extensions exist and have a formula in terms of (co)ends [Lor15, §2.1].

Proposition 2.1.12. *For any* $F : \mathbf{C} \to \mathbf{D}$ *and any* $G : \mathbf{C} \to \mathbf{Sets}$*, the left and right Kan extensions of* G *along* F *exist and are canonically isomorphic to the following (co)ends.*

$$\mathsf{Ran}_F G \cong \int_{c\in\mathbf{C}} \mathbf{Sets}(\mathbf{D}(-, Fc), Gc), \qquad \mathsf{Lan}_F G \cong \int^{c\in\mathbf{C}} \mathbf{D}(Fc, -) \times Gc.$$

Proof. [Lor15, §2.1] The following chain of natural isomorphisms proves the bijection. It relies on the previous results in (co)end calculus. We will first prove the adjunction for the right Kan extension.

$$\begin{aligned}
&\mathrm{Nat}\left(H, \int_{c\in\mathbf{C}} \mathbf{Sets}(\mathbf{D}(-, Fc), Gc)\right) \\
\cong\quad & \text{(Natural transformation as an end)} \\
&\int_{d\in\mathbf{D}} \mathbf{Sets}\left(Hd, \int_{c\in\mathbf{C}} \mathbf{Sets}(\mathbf{D}(d, Fc), Gc)\right) \\
\cong\quad & \text{(Continuity)} \\
&\int_{d\in\mathbf{D}} \int_{c\in\mathbf{C}} \mathbf{Sets}\left(Hd, \mathbf{Sets}(\mathbf{D}(d, Fc), Gc)\right) \\
\cong\quad & \text{(Exponential)} \\
&\int_{d\in\mathbf{D}} \int_{c\in\mathbf{C}} \mathbf{Sets}\left(\mathbf{D}(d, Fc) \times Hd, Gc\right) \\
\cong\quad & \text{(Fubini)} \\
&\int_{c\in\mathbf{C}} \int_{d\in\mathbf{D}} \mathbf{Sets}\left(\mathbf{D}(d, Fc) \times Hd, Gc\right) \\
\cong\quad & \text{(Continuity)} \\
&\int_{c\in\mathbf{C}} \mathbf{Sets}\left(\int^{d\in\mathbf{D}} \mathbf{D}(d, Fc) \times Hd, Gc\right) \\
\cong\quad & \text{(Yoneda lemma)} \\
&\int_{c\in\mathbf{C}} \mathbf{Sets}\left(HFc, Gc\right) \\
\cong\quad & \text{(Natural transformation as an end)} \\
&\mathrm{Nat}\left(H \circ F, G\right).
\end{aligned}$$

A similar reasoning can be used for the case of the left Kan extension.

$$\begin{aligned}
&\mathrm{Nat}\left(\int^{c\in\mathbf{C}} \mathbf{D}(Fc, -) \times Gc, H\right) \\
\cong\quad & \text{(Natural transformation as an end)} \\
&\int_{d\in\mathbf{D}} \mathbf{Sets}\left(\int^{c\in\mathbf{C}} \mathbf{D}(Fc, d) \times Gc, Hd\right) \\
\cong\quad & \text{(Continuity)} \\
&\int_{d\in\mathbf{D}} \int_{c\in\mathbf{C}} \mathbf{Sets}\left(\mathbf{D}(Fc, d) \times Gc, Hd\right) \\
\cong\quad & \text{(Exponential)} \\
&\int_{d\in\mathbf{D}} \int_{c\in\mathbf{C}} \mathbf{Sets}\left(Gc, \mathbf{Sets}(\mathbf{D}(Fc, d), Hd)\right) \\
\cong\quad & \text{(Fubini)} \\
&\int_{c\in\mathbf{C}} \int_{d\in\mathbf{D}} \mathbf{Sets}\left(Gc, \mathbf{Sets}(\mathbf{D}(Fc, d), Hd)\right) \\
\cong\quad & \text{(Continuity)}
\end{aligned}$$

$$
\begin{aligned}
& \int_{c\in\mathbf{C}} \mathbf{Sets}\left(Gc, \int_{d\in\mathbf{D}} \mathbf{Sets}(\mathbf{D}(Fc,d), Hd)\right) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int_{c\in\mathbf{C}} \mathbf{Sets}\left(Gc, HFc\right) \\
\cong \quad & \text{(Natural transformation as an end)} \\
& \mathrm{Nat}\left(G, H\circ F\right).
\end{aligned}
$$

As adjoints are unique up to isomorphism, these two derivations imply the result. □

2.2 Monoids

Monoids will appear repeteadly in this text in various forms. Instead of describing each one of these separately, we will make use of the formal theory of monoids and monads [Str72], that defines them as objects in an arbitrary 2-category. We will even consider *pseudomonoids* (see [nLa18]), monoids up to isomorphism in a monoidal 2-category.

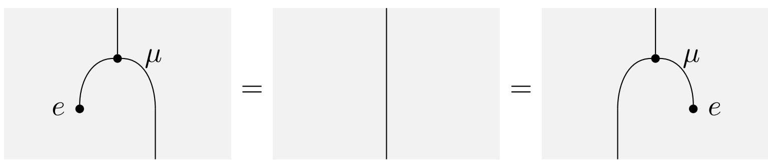
2.2.1 The category of monoids

Definition 2.2.1. In a monoidal category $(\mathbf{M}, \otimes, i, \lambda, \rho, \alpha)$, a **monoid** is an object $m \in \mathbf{M}$ endowed with morphisms $e\colon i \to m$ and $\mu\colon m\otimes m \to m$, called respectively *unit* and *multiplication*; and such that the following equalities hold. These are called *left/right unitality* and *associativity*.

$$
\begin{tikzcd}
m \arrow[r, "\lambda"] & i\otimes m \arrow[d, "e\otimes\mathrm{id}"] \\
& m\otimes m \arrow[ul, "\mu"]
\end{tikzcd}
\qquad
\begin{tikzcd}
m\otimes i \arrow[d, "\mathrm{id}\otimes e"'] & m \arrow[l, "\rho"'] \\
m\otimes m \arrow[ur, "\mu"'] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
m\otimes(m\otimes m) \arrow[rr, "\alpha"] \arrow[d, "\mu"] & & (m\otimes m)\otimes m \arrow[d, "\mu"] \\
m\otimes m \arrow[r, "\mu"] & m & m\otimes m \arrow[l, "\mu"']
\end{tikzcd}
$$

Diagramatically, following [Mar14], these equations are the following.



The same definition can be done in a bicategory after fixing a 0-cell. Recall that a monoidal category can be seen as a bicategory with a single object. If we relax the equalities to be isomorphisms, we get the notion of **pseudomonoid** in a monoidal 2-category. Finally, **comonoids** are monoids in the opposite category.

Definition 2.2.2. A **morphism of monoids** between (m, μ, e) and (n, μ', e') in the monoidal category $\mathbf{M}$ is given by a morphism $f\colon m \to n$ such that the following diagrams commute.

$$
\begin{tikzcd}
m \arrow[rr, "f"] & & n \\
& i \arrow[ul, "e"] \arrow[ur, "e'"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
m \otimes m \arrow[r, "f \otimes f"] \arrow[d, "\mu"'] & n \otimes n \arrow[d, "\mu'"] \\
m \arrow[r, "f"] & n
\end{tikzcd}
$$

In other words, it preserves the unit and the multiplication.

We can construct a category of monoids, $\mathbf{Mon}(\mathbf{M})$, over the original monoidal category. This makes it possible to talk about freely generated monoids. In some cases, we have a formula for constructing them.

Proposition 2.2.3 ([ML71, §VII.3, Theorem 2])**.** *Let $\mathbf{C}$ be a category with coproducts indexed by the natural numbers and where the tensor product distributes over the coproducts. The forgetful functor $U\colon \mathbf{Mon}(\mathbf{M}) \to \mathbf{M}$ has a left adjoint $(-)^*\colon \mathbf{Mon}(\mathbf{M}) \to \mathbf{M}$ which is given by the following* geometric series*.*

$$x^* = i + x + x \otimes x + x \otimes x \otimes x + \dots$$

Finally, we can consider monoids acting on another object of the bicategory, called a *monoid module*. This can also be extended to the case of a 2-category. In the more general case of a monoidal 2-category, the equalities are relaxed again to isomorphisms and we can define actions of a pseudomonoid, which are called *actegories* [nLa18].

Definition 2.2.4. Let (m, e, μ) be a monoid in a 2-category $\mathbf{M}$ with $m\colon A \to A$. A **module** for this monoid is a 1-cell $n\colon B \to A$ endowed with a 2-cell $h\colon m \otimes n \to n$. This requires n to be composable with m on one side. The module must satisfy some axioms saying that it interplays nicely with unitality and multiplication, as described in the following diagrams and in Figure 2.1.

$$
\begin{tikzcd}
(m \otimes m)n \arrow[rr, "\alpha"] \arrow[d, "\mu"'] & & m \otimes (m \otimes n) \arrow[d, "h"] \\
m \otimes n \arrow[r, "h"] & n & m \otimes n \arrow[l, "h"']
\end{tikzcd}
\qquad
\begin{tikzcd}
n \arrow[r, "\lambda"] & i \otimes n \arrow[d, "e \otimes \mathrm{id}"] \\
& m \otimes n \arrow[ul, "h"]
\end{tikzcd}
$$

The graphical representation of a monoid $m \in \mathbf{M}$ acting on some object $a \in \mathbf{M}$ where $\mathbf{M}$ is a 2-category can be seen in Figure 2.1.

Definition 2.2.5 ([Str72])**.** Let (m, e, μ) be a monoid in a 2-category with $m\colon A \to A$. The **Kleisli object** for this monoid is given by a module (A_m, f_m, λ) where $f_m\colon A \to A_m$ is a 1-cell and $\lambda\colon f_m \circ m \Rightarrow f_m$ is a 2-cell. The Kleisli object is the universal module, in the sense that any other module (B, g, κ) can be written uniquely as this module in parallel with some $u\colon A_m \to B$.

In the particular case of monads in $\mathbf{Cat}$, one recovers the usual notion of Kleisli category.

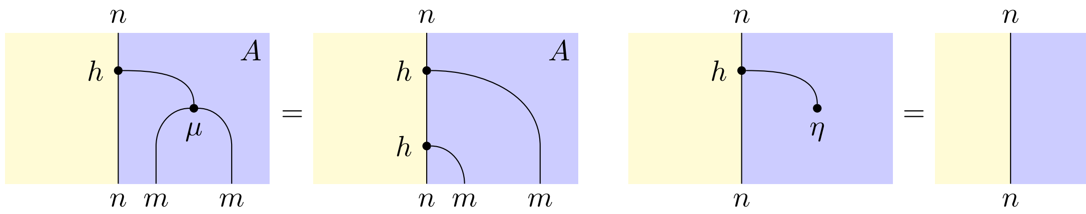


Figure 2.1: Axioms for a module in a 2-category, following Definition 2.2.4

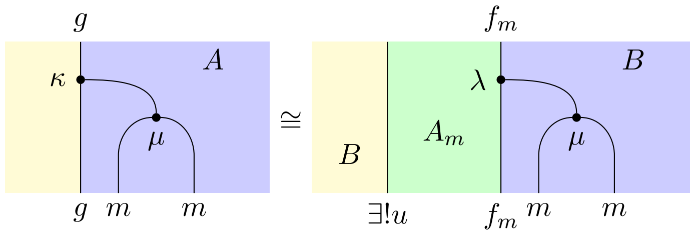


Figure 2.2: A diagrammatic description of the universal property of the Klesli object, following Definition 2.2.5.

2.2.2 Applicative functors

Definition 2.2.6. Let $(\mathbf{M}, \otimes, i)$ be a monoidal category. The category of copresheaves over it, $[\mathbf{M}, \mathbf{Sets}]$ can be endowed with the structure of a monoidal category using **Day convolution**. We write Day convolution of two copresheaves $F, G\colon \mathbf{M} \to \mathbf{Sets}$ as $F * G$; it is given by the following coend.

$$(F * G)(m) = \int^{x,y \in \mathbf{M}} \mathbf{M}(x \otimes y, m) \times F(x) \times G(y)$$

Using coend calculus, we can check both associativity and the fact that the unit for this monoidal category is $よ_i$, the representable copresheaf on the unit of the monoidal category. See [Lor15, §6] for details. It makes sense now to ask what are the monoids of this new monoidal category. In the case where we also take our base monoidal category to be **Sets**, these are the *applicative functors* widely used in programming contexts and defined in [MP08]. The description of applicatives as monoids for Day convolution can be found in [RJ17].

Definition 2.2.7. An **applicative functor** is a monoid on the category of endofunctors on **Sets** endowed with Day convolution as a tensor product.

This means that it is endowed with natural transformations $e\colon よ_i \Rightarrow F$ and $\mu\colon F*F \Rightarrow F$. Considering we are working in a cartesian closed category, we can reduce the first to a family $u_a\colon a \to Fa$ natural in a; and the second can be reduced via (co)end calculus to a natural family of morphisms $w_{a,b}\colon Fa \times Fb \to F(a \times b)$ natural on $a, b \in \mathbf{Sets}$.

$$\begin{aligned} & F * F \Rightarrow F \\ \cong \quad & \text{(Natural transformation as an end)} \end{aligned}$$

$$
\begin{aligned}
&\int_c \mathbf{Sets}((F * F)c, Fc) \\
\cong \quad & \text{(Definition of Day convolution)} \\
&\int_c \mathbf{Sets}\left(\left(\int^{a,b} Fa \times Fb \times \mathbf{Sets}(a \times b, c)\right), Fc\right) \\
\cong \quad & \text{(Continuity)} \\
&\int_c \int_{a,b} \mathbf{Sets}(Fa \times Fb \times \mathbf{Sets}(a \times b, c), Fc) \\
\cong \quad & \text{(Fubini)} \\
&\int_{a,b} \int_c \mathbf{Sets}(Fa \times Fb \times \mathbf{Sets}(a \times b, c), Fc) \\
\cong \quad & \text{(Yoneda lemma)} \\
&\int_{a,b} \mathbf{Sets}(Fa \times Fb, F(a \times b)).
\end{aligned}
$$

We will write now the monoid axioms in terms of this representation. Knowing how the Yoneda lemma is constructed, one can see how to write μ in terms of u. The laws then become the following diagrams; we left implicit the unitors and associators of the cartesian product.

$$
\begin{tikzcd}
Fa \arrow[r, "u_1"] \arrow[dr, "\mathrm{id}"'] & Fa \times F1 \arrow[d, "\mu_{a,1}"] \\
& Fa
\end{tikzcd}
\qquad
\begin{tikzcd}
F1 \times Fa \arrow[d, "\mu_{1,a}"'] & Fa \arrow[l, "u_1"'] \arrow[dl, "\mathrm{id}"] \\
Fa &
\end{tikzcd}
$$

$$
\begin{tikzcd}
Fa \times Fb \times Fc \arrow[r, "w_{a,b}"] \arrow[d, "w_{b,c}"'] & F(a \times b) \times Fc \arrow[d, "w_{a \times b, c}"] \\
Fa \times F(b \times c) \arrow[r, "w_{a, b \times c}"] & F(a \times b \times c)
\end{tikzcd}
$$

Applicative functors are also defined sometimes with a family of morphisms of the form $\mu'_{b,c} \colon F(b \to c) \times Fb \to Fc$. This is again equivalent via the Yoneda lemma.

$$
\begin{aligned}
&\int_{a,b} \mathbf{Sets}(Fa \times Fb, F(a \times b)) \\
\cong \quad & \text{(Yoneda lemma)} \\
&\int_{a,b} \mathbf{Sets}\left(Fa \times Fb, \int_c \mathbf{Sets}\left(\mathbf{Sets}(a \times b, c), F(c)\right)\right) \\
\cong \quad & \text{(Exponential)} \\
&\int_{a,b} \mathbf{Sets}\left(Fa \times Fb, \int_c \mathbf{Sets}\left(\mathbf{Sets}(a, b \to c), F(c)\right)\right) \\
\cong \quad & \text{(Continuity)} \\
&\int_{a,b} \int_c \mathbf{Sets}\left(Fa \times Fb, \mathbf{Sets}\left(\mathbf{Sets}(a, b \to c), F(c)\right)\right) \\
\cong \quad & \text{(Fubini)} \\
&\int_{b,c} \int_a \mathbf{Sets}\left(Fa \times Fb, \mathbf{Sets}\left(\mathbf{Sets}(a, b \to c), F(c)\right)\right) \\
\cong \quad & \text{(Exponential)}
\end{aligned}
$$

$$
\begin{aligned}
& \int_{b,c} \int_a \mathbf{Sets}\,(Fa \times Fb \times \mathbf{Sets}(a, b \to c), F(c)) \\
\cong \quad & \text{(Exponential)} \\
& \int_{b,c} \int_a \mathbf{Sets}\,(\mathbf{Sets}(a, b \to c), \mathbf{Sets}(Fa \times Fb, F(c))) \\
\cong \quad & \text{(Yoneda)} \\
& \int_{b,c} \mathbf{Sets}(F(b \to c) \times Fb, F(c)).
\end{aligned}
$$

In any of these cases, we can define a category **App** of applicative functors with monoid morphisms between them. Being monoids, the next question is if we can generate *free* applicative functors. We will apply Proposition 2.2.3, but we first need a lemma showing that that our case actually satisfies the conditions of the theorem.

Lemma 2.2.8. *Day convolution distributes over coproducts. Recall that colimits in a category of presheaves are computed pointwise. We are saying then that* $F * (G + H) \cong (F * G) + (F * H)$ *for any functors* $F, G, H \in [\mathbf{C}, \mathbf{Sets}]$.

Proof. The proof is straightforward using (co)end calculus and making the coend distribute over the coproduct. We compute the following chain of isomorphisms natural in $c \in \mathbf{Sets}$.

$$
\begin{aligned}
& F * (G + H) \\
\cong \quad & \text{(Definition of Day convolution)} \\
& \int^{x,y} \mathbf{Sets}(a \times b, -) \times (G + H)a \times Fb \\
\cong \quad & \text{(The coproduct of functors is computed pointwise)} \\
& \int^{x,y} \mathbf{Sets}(a \times b, -) \times (Ga + Ha) \times Fb \\
\cong \quad & \text{(Product distributes over sum)} \\
& \int^{x,y} \mathbf{Sets}(a \times b, -) \times Ga \times Fb + \mathbf{Sets}(a \times b, -) \times Ha \times Fb \\
\cong \quad & \text{(End distributes over sum)} \\
& \left(\int^{x,y} \mathbf{Sets}(a \times b, -) \times Ga \times Fb \right) + \left(\int^{x,y} \mathbf{Sets}(a \times b, -) \times Ha \times Fb \right) \\
\cong \quad & \text{(Definition of Day convolution)} \\
& (F * G) + (F * H).
\end{aligned}
$$

□

Theorem 2.2.9. *Let* $X \colon \mathbf{Sets} \to \mathbf{Sets}$ *be an endofunctor. The free applicative functor over it,* X^**, can be computed as the following colimit.*

$$X^* = \mathrm{Id} + X + X * X + X * X * X + \dots$$

In particular, we have the following adjunction $\mathbf{App}(X^*, F) \cong [\mathbf{C}, \mathbf{Sets}](X, F)$ *natural in* F*, an applicative functor; and* X*, an arbitrary functor.*

Proof. We apply Proposition 2.2.3. The category of copresheaves has all small coproducts and Day convolution distributes over coproducts because of Lemma 2.2.8. Note also that the unit of Day convolution in this case is given by $\mathbf{Sets}(1, -)$, which is naturally isomorphic to the identity functor. $\square$

We will be interested in the following particular case both during our study of traversable functors and while describing an optic for applicative functors.

Corollary 2.2.10. *Let $S(c) = a \times (b \to c)$ be a functor for some fixed $a, b \in \mathbf{Sets}$. Its free applicative functor is $S^*(c) = \sum_n a^n \times (b^n \to c)$.*

Proof. We will show that the Day convolution of $a^n \times (b^n \to -)$ with $a \times (b \to -)$ is precisely $a^{n+1} \times (b^{n+1} \to -)$.

$$
\begin{aligned}
& \int^{x,y} \mathbf{Sets}(x \times y, c) \times (a \times (b \to x)) \times (a^n \times (b^n \to y)) \\
\cong \quad & \text{(Fubini rule)} \\
& a^{n+1} \times \int^{x,y} \mathbf{Sets}(x \times y, c) \times (b \to x) \times (b^n \to y) \\
\cong \quad & \text{(Exponential)} \\
& a^{n+1} \times \int^{x,y} \mathbf{Sets}(x, y \to c) \times (b \to x) \times (b^n \to y) \\
\cong \quad & \text{(Yoneda lemma)} \\
& a^{n+1} \times \int^{y} (b \to (y \to c)) \times (b^n \to y) \\
\cong \quad & \text{(Exponential)} \\
& a^{n+1} \times \int^{y} (y \to (b \to c)) \times (b^n \to y) \\
\cong \quad & \text{(Yoneda lemma)} \\
& a^{n+1} \times (b^n \to (b \to c)) \\
\cong \quad & \text{(Exponential)} \\
& a^{n+1} \times (b^{n+1} \to c)
\end{aligned}
$$

We note also that the identity functor can be written as $a^0 \times (b^0 \to -)$. Thus, we can apply Lemma 2.2.9 and complete the proof by induction. $\square$

2.2.3 Morphisms of monads

Definition 2.2.11 ([Red95]). A **monad morphism** between two monads, (S, μ, η) and (T, μ', η') is a natural transformation $\alpha \colon S \to T$ preserving units and multiplications. Diagrammatically, it must make the following diagram commute.

$$
\begin{tikzcd}
S^2 \arrow[r, "\mu"] \arrow[d, "\alpha^2"'] & S \arrow[d, "\alpha"] & \mathrm{Id} \arrow[l, "\eta"'] \arrow[d, "\mathrm{id}"] \\
T^2 \arrow[r, "\mu'"'] & T & \mathrm{Id} \arrow[l, "\eta'"]
\end{tikzcd}
$$

In other words, it is a morphism of monoids in the monoidal category of endofunctors. Dually, a morphism of comonads is a morphism of comonoids in the category of endofunctors.

Every monad morphism $\alpha\colon S \to T$ induces a functor T-Alg $\to$ S-Alg between the Eilenberg-Moore categories of the monads. A morphism of T-algebras $f\colon (a,\rho) \to (b,\rho')$, can be reinterpreted as a morphism of S-algebras as follows.

$$\begin{tikzcd}
Sa \arrow[r, "Sf"] \arrow[d, "\alpha"'] & Sb \arrow[d, "\alpha"] \\
Ta \arrow[r, "Tf"] \arrow[d, "\rho"'] & Tb \arrow[d, "\rho'"] \\
a \arrow[r, "f"] & b
\end{tikzcd}$$

In this diagram, the upper square commutes because of the definition of monad morphism. The lower square commutes because f is a morphism of T-algebras.

Proposition 2.2.12. *Let (S,μ,η) and (T,μ',η') be two monads on the same category $\mathbf{C}$. Let $A\colon T\text{-}Alg \to S\text{-}Alg$ be a functor preserving the forgetful functor.*

$$\begin{tikzcd}
T\text{-}Alg \arrow[rr, "A"] \arrow[dr, "U"'] & & S\text{-}Alg \arrow[dl, "U"] \\
& \mathbf{C} &
\end{tikzcd}$$

Then A is induced by some monad morphism in the way described earlier.

Proof. We know that, for any $x \in \mathbf{C}$, the functor A takes the free T-algebra $(\mu'_x\colon T^2x \to Tx)$ into some S-algebra that we call $(\overline{\mu}_x\colon STx \to Tx)$. We will show that this defines a natural transformation $\overline{\mu}\colon ST \Rightarrow T$. In fact, given any $f\colon x \to y$, the morphism $Tf\colon Tx \to Ty$ is a morphism between the free algebras because of naturality of μ'. Given that A preserves the forgetful functor, Tf must also be a morphism between the algebras given by $\overline{\mu_x}$ and $\overline{\mu_y}$. This is precisely naturality.

$$\begin{tikzcd}
STx \arrow[r, "STf"] \arrow[d, "\overline{\mu}_x"'] & STy \arrow[d, "\overline{\mu}_y"] \\
Tx \arrow[r, "Tf"] & Ty \\
T^2x \arrow[r, "T^2f"] \arrow[u, "\mu'"] & T^2y \arrow[u, "\mu'"']
\end{tikzcd}$$

Now we define the natural transformation $\alpha = \overline{\mu} \circ S\eta\colon S \Rightarrow T$. We will show it is a monad morphism. It preserves the unit because of commutativity of the following diagram, that uses that η is a natural transformation and $\overline{\mu}$ is a monad algebra.

$$\begin{tikzcd}
x \arrow[r, "\eta'_x"] \arrow[d, "\eta_x"] & Tx \arrow[d, "\eta_{Tx}"] \arrow[dr, "\mathrm{id}"] & \\
Sx \arrow[r, "S\eta'_x"'] & STx \arrow[r, "\overline{\mu}_x"'] & Tx
\end{tikzcd}$$

It also preserves the multiplication because of commutativity of the following diagram. Here we use that μ is a natural transformation, that $\overline{\mu}$ is a monad algebra, the unitality of the monad T, and the fact that μ' must be a morphism between the corresponding S-algebras thanks to the action of the functor.

$$\begin{tikzcd}
S^2x \arrow[r, "S^2\eta_x"] \arrow[d, "\mu_x"] & S^2Tx \arrow[r, "S\overline{\mu}_x"] \arrow[d, "\mu_{Tx}"] & STx \arrow[r, "S\eta'_{Tx}"] \arrow[d, "\overline{\mu}_x"] & ST^2x \arrow[l, bend left, "S\mu'_x"] \arrow[d, "\overline{\mu}_{Tx}"] \\
Sx \arrow[r, "S\eta'_x"] & STx \arrow[r, "\overline{\mu}_x"] & Tx & T^2x \arrow[l, "\mu'_x"'] \\
 & & T^2x \arrow[u, "\mu_x"] & T^3x \arrow[l, "T\mu'_x"] \arrow[u, "\mu_{Tx}"']
\end{tikzcd}$$

We can finally check that this α induces the original functor A. In fact, for any T-algebra $(l\colon Tx \to x)$, we have an algebra morphism from the free algebra $(\mu'\colon T^2x \to Tx)$ given precisely by $l\colon Tx \to x$. This must be sent by the functor to the following S-algebra morphism, whose commutativity determines that A is induced by α. □

$$\begin{tikzcd}
STx \arrow[r, "Sl"] \arrow[d, "\overline{\mu}_x"] & Sx \arrow[l, bend left, "S\eta"] \arrow[d] \\
Tx \arrow[r, "l"] & x
\end{tikzcd}$$

2.2.4 Monoidal actions

A monoidal category can be seen as a pseudomonoid in **Cat**, and we can consider strong monoidal functors between them that preserve the action up to isomorphism.

Definition 2.2.13. A **strong monoidal functor** between two monoidal categories $(\mathbf{M}, \otimes, i)$ and $(\mathbf{N}, \boxtimes, j)$ is given by a functor $F\colon \mathbf{M} \to \mathbf{N}$ together with *structure isomorphisms* $\phi_i\colon j \cong F(i)$ and $\phi_{m,n}\colon F(m) \boxtimes F(n) \cong F(m \otimes n)$ that interplay nicely with the associators and unitors of the monoidal category in the sense that they make the following diagrams commute.

$$\begin{tikzcd}
(Fx \boxtimes Fy) \boxtimes Fz \arrow[r, "\alpha_{\mathbf{N}}"] \arrow[d, "\phi_{x,y}"'] & Fx \boxtimes (Fy \boxtimes Fz) \arrow[d, "\phi_{y,z}"] \\
F(x \otimes y) \boxtimes Fz \arrow[d, "\phi_{(x\otimes y),z}"'] & Fx \boxtimes F(y \otimes z) \arrow[d, "\phi_{x,(y\otimes z)}"] \\
F((x \otimes y) \otimes z) \arrow[r, "\alpha_M"] & F(x \otimes (y \otimes z))
\end{tikzcd}$$

$$\begin{tikzcd}
Fi \boxtimes Fx \arrow[d, "\phi_{i,x}"'] & j \boxtimes Fx \arrow[l, "\phi_i"'] \arrow[d, "\lambda_{\mathbf{N}}"] \\
F(i \otimes x) \arrow[r, "F\lambda_{\mathbf{M}}"'] & Fx
\end{tikzcd}
\qquad
\begin{tikzcd}
Fx \boxtimes j \arrow[r, "\phi_i"] \arrow[d, "\rho_{\mathbf{N}}"'] & Fx \boxtimes Fi \arrow[d, "\phi_{x,i}"] \\
Fx & F(x \otimes i) \arrow[l, "F\rho_{\mathbf{M}}"]
\end{tikzcd}$$

Let us denote by **MonCat** the category of monoidal categories with strong monoidal functors between them. Note that the identity and the composition of two strong monoidal functors are again strong monoidal functors.

Definition 2.2.14. A **monoidal action** from a monoidal category $\mathbf{M}$ into a category $\mathbf{C}$ is a strong monoidal functor $F \colon \mathbf{M} \to [\mathbf{C}, \mathbf{C}]$ that has as target the monoidal category of endofunctors of $\mathbf{C}$ endowed with composition as the monoidal product. We can consider the slice category $\mathbf{MonCat}/[\mathbf{C}, \mathbf{C}]$ as the category of monoidal actions.

It can be shown that these are precisely pseudomonid modules $\mathbf{M} \times \mathbf{C} \to \mathbf{C}$, whose laws are true up to some isomorphism. In the case of strong monoidal actions, because of the strictness of the monoidal category of endofunctors, the coherence diagrams become simplified.

2.3 The bicategory of profunctors

Definition 2.3.1. The bicategory **Prof** has categories as 0-cells. 1-cells between two categories $\mathbf{C}$ and $\mathbf{D}$ are profunctors $\mathbf{C} \nrightarrow \mathbf{D}$, and 2-cells between two profunctors are natural transformations. The composition of two profunctors $p \colon \mathbf{C} \nrightarrow \mathbf{D}$ and $q \colon \mathbf{D} \nrightarrow \mathbf{E}$ is written as $(q \diamond p) \colon \mathbf{C} \nrightarrow \mathbf{E}$, and is given by the following coend.

$$(q \diamond p)(c, e) = \int^{d \in \mathbf{D}} p(c, d) \times q(d, e).$$

The identity for this composition in a category $\mathbf{C}$ is the hom-profunctor $\mathbf{C}(-,-) \colon \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{Sets}$. A detailed description of this bicategory, together with proofs for unitality and associativity can be found for instance in [Lor15, §5], where profunctors are called *relators*.

2.3.1 Promonads

Definition 2.3.2. We give the name **promonads** to the monoids in the 2-category **Prof**.

Let $\mathbf{C}$ be a category and $p \colon \mathbf{C} \nrightarrow \mathbf{C}$ an endoprofunctor. A promonad structure on this endoprofunctor is given by some *unit*, a family of functions $\eta_{a,b} \colon \mathbf{C}(a, b) \to p(a, b)$ natural in $a, b \in \mathbf{C}$; and some *multiplication*, a family of functions

$$\mu_{a,c} \colon \left(\int^{b} p(a, b) \otimes p(b, c) \right) \to p(a, c)$$

natural in $a, b \in \mathbf{C}$. Using continuity, we can rewrite the multiplication as an element of the following end, that resembles function composition.

$$\mu_{a,b} \colon \int_{b} p(a, b) \times p(b, c) \to p(a, c)$$

It can be shown that unitality for the promonad is the fact that the identity $\mathrm{id} \in \mathbf{C}(a, a)$ is sent to some element in $p(a, a)$ that acts as the identity for this composition. Associativity for the promonad is the fact that this composition is associative. In this sense, a promonad could be seen as embedding the category $\mathbf{C}$ into a category with the same objects but new morphisms. This intuition can be made precise by the following proposition.

Proposition 2.3.3 ([Str72])**.** *A promonad induces an identity-on-objects functor to the Kleisli category of the promonad. Every identity-on-objects functor arises in this way for some promonad.*

2.3.2 Promonad modules

Lemma 2.3.4. *If* $\psi\colon \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{Sets}$ *is a promonad, then* $(\psi \diamond -)\colon [\mathbf{C}^{op} \times \mathbf{C}, \mathbf{Sets}] \to [\mathbf{C}^{op} \times \mathbf{C}, \mathbf{Sets}]$ *is a monad. Modules over the promonad* ψ *are precisely the algebras over the monad.*

Proof. The unit of the monad is given by the unit of the promonad $\eta_p\colon p \cong \mathbf{C}(-,-) \diamond p \Rightarrow \psi \diamond p$. The multiplication is in turn given by the multiplication of the promonad as $\mu_p\colon \psi \diamond (\psi \diamond p) \cong (\psi \diamond \psi) \diamond p \Rightarrow \psi \diamond p$. It can be seen that the axioms for the monad are precisely the axioms for the promonad.

Now a module over the promonad is given by $\psi \diamond p \Rightarrow p$, which is precisely the data for an algebra over the monad $(\psi \diamond -)$. Because of the definition of the unit and multiplication of the promonad, the axioms of the monad algebra are precisely the axioms for the module. □

2.4 Pseudomonic functors and replete subcategories

We will follow [nLa18] into getting the necessary definitions to construct a well-behaved notion of 2-categorical image.

Definition 2.4.1. A functor $F\colon \mathbf{C} \to \mathbf{D}$ is **pseudomonic** if it is faithful and full on isomorphisms. That is, given two $c, c' \in \mathbf{C}$, every isomorphism $Fc \cong Fc'$ is the image under the functor of an isomorphism $c \cong c'$.

An alternative definition of pseudomonic functor is that they are precisely the functors $F\colon \mathbf{C} \to \mathbf{D}$ such that the following square is a pullback. It can be seen that pseudomonic functors are stable under pullback.

$$\begin{tikzcd} \mathbf{C} \arrow[r, "\mathrm{id}"] \arrow[d, "\mathrm{id}"'] & \mathbf{C} \arrow[d, "F"] \\ \mathbf{C} \arrow[r, "F"] & \mathbf{D} \end{tikzcd}$$

Definition 2.4.2. A subcategory $\mathbf{C} \subseteq \mathbf{D}$ is **replete** if, for all $c \in \mathbf{C}$, the existence of an isomorphism $f\colon c \cong d$ in $\mathbf{D}$ implies that $d \in \mathbf{C}$ and $f \in \mathbf{C}(c, d)$. That is, the subcategory respects the isomorphisms of the original category.

The smallest replete subcategory containing a subcategory $\mathbf{C} \subseteq \mathbf{D}$ is called its **repletion**, $\mathrm{repl}(\mathbf{C})$. It can be explicitly constructed taking all objects of $\mathbf{D}$ that admit an isomorphism to an object in $\mathbf{C}$ and taking all morphisms that can be written as composites of morphisms in $\mathbf{C}$ and isomorphisms in $\mathbf{D}$.

Proposition 2.4.3. *Let* $\mathbf{C} \subseteq \mathbf{D}$ *be a subcategory. The inclusion* $\mathbf{C} \to \mathrm{repl}(\mathbf{C})$ *is an equivalence if and only if the inclusion* $\mathbf{C} \to \mathbf{D}$ *is pseudomonic.*

Proof. Note that the inclusion of a replete subcategory, $\mathrm{repl}(\mathbf{C}) \to \mathbf{D}$, is pseudomonic by definition. If $\mathbf{C} \to \mathrm{repl}(\mathbf{C})$ is an equivalence, then it is fully faithful and pseudomonic in particular. These two facts make the composite $\mathbf{C} \to \mathrm{repl}(\mathbf{C}) \to \mathbf{D}$ pseudomonic.

Let $\mathbf{C} \to \mathbf{D}$ be pseudomonic. Morphisms in $\mathrm{repl}(\mathbf{C})$ can be written as the composition of morphisms of the form $g \circ f \circ h$ where f is a morphism in $\mathbf{C}$ and g, h are isomorphisms in $\mathbf{D}$. We will show that for any $c, c' \in \mathbf{C}$, $\mathrm{repl}(\mathbf{C})(c, c') \subseteq \mathbf{C}(c, c')$ by induction on the minimum number of pieces $(g \circ f \circ h)$ required to form the morphism. In the case of a morphism $(g \circ f \circ h) \in \mathrm{repl}(\mathbf{C})(c, c')$, the isomorphisms g and h have their source and target in $\mathbf{C}$, and they must be morphisms of $\mathbf{C}$ because of the pseudomonic condition. In the case of a morphism $(g_1 \circ f_1 \circ h_1) \circ \cdots \circ (g_k \circ f_k \circ h_k)$, we can apply the inductive hypothesis to show that $f_1 \circ h_1 \circ \cdots \circ g_k \circ f_k$ is a morphism in $\mathbf{C}$ and then apply the same reasoning as in the base case to conclude that g_1 and h_k are morphisms in $\mathbf{C}$.

This shows that the inclusion $\mathbf{C} \to \mathrm{repl}(\mathbf{C})$ is full. It is faithful and essentially surjective by definition, which shows it is an equivalence. $\square$

Given an arbitrary functor $F \colon \mathbf{C} \to \mathbf{D}$, that does not need to be pseudomonic, we can define its **image**, $\mathrm{img}(F)$, as the least subcategory containing all objects and morphisms that are images of the functor. Note that this does not mean that every morphism in $\mathrm{img}(F)$ will be the image of some morphism in $\mathbf{C}$ under the functor F; there could exist morphisms that become composable only after applying the functor. The image of any functor is a subcategory, and we can define the **replete image** of a functor as the repletion of its image, $\mathrm{repl}(\mathrm{img}(F))$.

Chapter 3

Existential optics

3.1 Existential optics

The structure that is common to all optics is that they split a bigger structure s into the focus a and some context m acting on it. In some sense, we cannot access or act on that context, only on its shape. The definition will capture this fact using the dinaturality condition of a coend. However, we can still use this context to update the original data structure, replacing the current focus by a new element.

Definition 3.1.1. Consider a monoidal category $\mathbf{M}$ and two arbitrary categories $\mathbf{C}$ and $\mathbf{D}$. Let $\underline{(\_)}\colon \mathbf{M} \to [\mathbf{C},\mathbf{C}]$ and $\underline{(\_)}\colon \mathbf{M} \to [\mathbf{D},\mathbf{D}]$ be two strong monoidal functors and let $s, a \in \mathbf{C}$ and $t, b \in \mathbf{D}$. An **optic** from (s,t) with focus on (a,b) is an element of the following set described as a coend.

$$\mathbf{Optic}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} s \\ t \end{pmatrix}\right) = \int^{m \in \mathbf{M}} \mathbf{C}(s, \underline{m}a) \times \mathbf{D}(\underline{m}b, t)$$

Remark 3.1.2. The definition of *optic* given by [Ril18], or the one considered in [BG18], deal only with the particular case in which $\mathbf{D}$ and $\mathbf{C}$ are the same category and both actions are identified. We are actually defining what [Ril18] calls *mixed optics*, that do not have this limitation. The following construction of the category of optics is similar to that of Proposition 2.0.3 in [Ril18] but it provides a more general result, as it is considering arbitrary monoidal actions instead of monoidal products and *mixed* optics instead of assuming $\mathbf{C} = \mathbf{D}$.

It can be shown directly that **Optic** can be given the structure of a category; but note that we could also wait for the profunctor representation theorem (Theorem 5.3.1) to describe **Optic** as a Kleisli category. The main idea is that we can compose $s \to \underline{m}a$ and $a \to \underline{n}x$ into $s \to \underline{(m \otimes n)}x$; and in the same way, we can compose $\underline{m}b \to t$ and $\underline{n}y \to b$ into $\underline{(m \otimes n)}y \to t$. The following definition and proofs are just a formality around the structure morphisms of the monoidal category and the monoidal functor that ensures that everything works nicely.

In order to show that **Optic** is a category we use a notation directly taken from [Ril18]. We write elements of the end $\int^{m \in \mathbf{M}} \mathbf{C}(s, \underline{m}a) \times \mathbf{D}(\underline{m}b, t)$ as pairs of functions $\langle l \mid r \rangle$ for $l \in \mathbf{C}(s, \underline{m}a)$ and $r \in \mathbf{D}(\underline{m}b, t)$. These pairs are quotiented by a relation that equates $\langle \alpha \circ l \mid r \rangle \sim \langle l \mid r \circ \alpha \rangle$ for any $\alpha \in \mathbf{M}(m,n)$, any $l \in \mathbf{C}(s, \underline{m}a)$ and any $r \in \mathbf{D}(\underline{n}b, t)$.

The *identity* of **Optic** is defined as $\mathrm{id} = \langle \phi_i^{-1} \mid \phi_i \rangle$, where $\phi_i \colon \underline{i}a \to a$ is the structure map of the monoidal action. Consider two optics $\langle l \mid r \rangle \in \mathbf{Optic}((a,b),(s,t))$ and $\langle l' \mid r' \rangle \in$

$\mathbf{Optic}((a,b),(x,y))$ given by $l \in \mathbf{C}(s, \underline{m}a)$, $l' \in \mathbf{C}(a, \underline{n}x)$, $r' \in \mathbf{D}(\underline{n}y, b)$ and $r \in \mathbf{D}(\underline{m}b, t)$; their *composition* is defined as $\langle \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \phi_{m,n} \rangle$, where $\phi_{m,n} \colon \underline{(m \otimes n)}a \to \underline{mn}a$ is the structure map of the monoidal action. We show now that composition is well-defined with respect to the equivalence relation. In fact, for any $\alpha \colon \mathbf{M}(m,m')$ and $\beta \colon \mathbf{M}(n,n')$, the relations $\langle \alpha \circ l \mid r \rangle \sim \langle l \mid r \circ \alpha \rangle$ and $\langle \beta \circ l' \mid r' \rangle \sim \langle l' \mid r' \circ \beta \rangle$ translate into the following relation.

$$
\begin{aligned}
& \langle \phi_{m,n}^{-1} \circ \underline{m}'(\beta \circ l') \circ (\alpha \circ l) \mid r \circ \underline{m}r' \circ \phi_{m,n} \rangle \\
= \quad & \text{(Functoriality of } \underline{m}'\text{)} \\
& \langle \phi_{m,n}^{-1} \circ \underline{m}'\beta \circ \underline{m}'l' \circ \alpha \circ l \mid r \circ \underline{m}r' \circ \phi_{m,n} \rangle \\
= \quad & \text{(Functoriality of the action makes } \alpha \text{ natural)} \\
& \langle \phi_{m,n}^{-1} \circ \underline{m}'\beta \circ \alpha \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \phi_{m,n} \rangle \\
= \quad & \text{(Naturality of } \phi\text{)} \\
& \langle (\beta \otimes \alpha) \circ \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \phi_{m,n} \rangle \\
\sim \quad & \text{(Equivalence relation)} \\
& \langle \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \phi \circ (\beta \otimes \alpha) \rangle \\
= \quad & \text{(Naturality of } \phi\text{)} \\
& \langle \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \underline{m}\beta \circ \alpha \circ \phi_{m,n} \rangle \\
= \quad & \text{(Functoriality of } \underline{m}\text{)} \\
& \langle \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid r \circ \underline{m}(r' \circ \beta) \circ \alpha \circ \phi_{m,n} \rangle \\
= \quad & \text{(Functoriality of the action makes } \alpha \text{ natural)} \\
& \langle \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid (r \circ \alpha) \circ \underline{m}'(r' \circ \beta) \circ \phi_{m,n} \rangle
\end{aligned}
$$

We proceed now to check that the identity is neutral with respect to composition and that composition is associative. For the first, we are going to use that the definition of strong monoidal functor imposes $\phi_{m,i}^{-1} \circ \underline{m}\phi_i = \underline{\lambda}$. Composition with the identity on the left goes as follows.

$$
\begin{aligned}
& \langle l \mid r \rangle \circ \langle \phi_i^{-1} \mid \phi_i \rangle \\
= \quad & \text{(Definition of composition)} \\
& \langle \phi_{m,i}^{-1} \circ m\phi_i^{-1} \circ l \mid r \circ m\phi_i \circ \phi_{m,i} \rangle \\
= \quad & \text{(Conditions on a strong monoidal functor)} \\
& \langle \lambda^{-1} \circ l \mid r \circ \lambda \rangle \\
\sim \quad & \text{(Equivalence relation)} \\
& \langle l \mid r \rangle
\end{aligned}
$$

And composition on the right follows a similar reasoning.

$$
\begin{aligned}
& \langle \phi_i^{-1} \mid \phi_i \rangle \circ \langle l \mid r \rangle \\
= \quad & \text{(Definition of composition)} \\
& \langle \phi_{m,i}^{-1} \circ \underline{i}l \circ \phi_i^{-1} \mid \phi_i \circ \underline{i}r \circ \phi_{m,i} \rangle \\
= \quad & \text{(Naturality of } \phi\text{)} \\
& \langle \phi_{m,i}^{-1} \circ \phi_i^{-1} \circ l \mid r \circ \phi_i \circ \phi_{m,i} \rangle \\
= \quad & \text{(Conditions on a strong monoidal functor)}
\end{aligned}
$$

$$
\begin{aligned}
& \langle \lambda^{-1} \circ l \mid r \circ \lambda \rangle \\
\sim \quad & \text{(Equivalence relation)} \\
& \langle l \mid r \rangle
\end{aligned}
$$

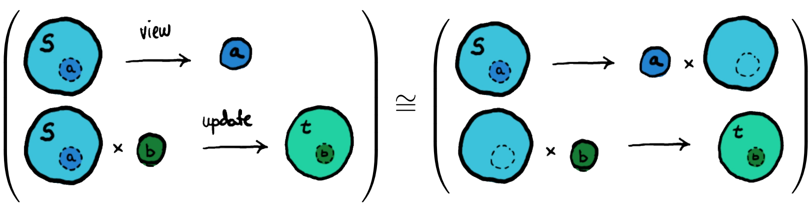
Finally, associativity holds because of the following chain of equations. Let $s, a, x, u \in \mathbf{C}$ and $t, b, y, v \in \mathbf{D}$. Let, on one side, $l \colon s \to \underline{m}a$, $l' \colon a \to \underline{n}x$ and $l'' \colon x \to \underline{k}u$ be morphisms in $\mathbf{C}$; let, on the other side, $r \colon \underline{m}b \to t$, $r' \colon \underline{n}y \to b$ and $r'' \colon \underline{k}v \to y$ be morphisms in $\mathbf{D}$.

$$
\begin{aligned}
& \langle l \mid r \rangle \circ (\langle l' \mid r' \rangle \circ \langle l'' \mid r'' \rangle) \\
= \quad & \text{(Definition of composition)} \\
& \langle l \mid r \rangle \circ \langle \phi_{n,k}^{-1} \circ \underline{n}l'' \circ l' \mid r' \circ \underline{n}r'' \circ \phi_{n,k} \rangle \\
= \quad & \text{(Definition of composition)} \\
& \langle \phi_{m,n\otimes k}^{-1} \circ \underline{m}(\phi_{n,k}^{-1} \circ \underline{n}l'' \circ l') \circ l \mid r \circ \underline{m}(r' \circ \underline{n}r'' \circ \phi_{n,k}) \circ \phi_{m,n\otimes k} \rangle \\
= \quad & \text{(Functoriality of } \underline{m}\text{)} \\
& \langle \phi_{m,n\otimes k}^{-1} \circ \underline{m}\phi_{n,k}^{-1} \circ \underline{mn}l'' \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \underline{mn}r'' \circ \underline{m}\phi_{n,k} \circ \phi_{m,n\otimes k} \rangle \\
\sim \quad & \text{(Equivalence relation)} \\
& \langle \alpha_{m,n,k} \circ \phi_{m,n\otimes k}^{-1} \circ \underline{m}\phi_{n,k}^{-1} \circ \underline{mn}l'' \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \underline{mn}r'' \circ \underline{m}\phi_{n,k} \circ \phi_{m,n\otimes k} \circ \alpha_{m,n,k}^{-1} \rangle \\
= \quad & \text{(Axioms of a strong monoidal functor)} \\
& \langle \phi_{m\otimes n,k}^{-1} \circ \phi_{m,n}^{-1} \circ \underline{mn}l'' \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \underline{mn}r'' \circ \phi_{m,n} \circ \phi_{m\otimes n,k} \rangle \\
= \quad & \text{(Naturality of } \phi\text{)} \\
& \langle \phi_{m\otimes n,k}^{-1} \circ \underline{(m \otimes n)}l'' \circ \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \phi_{m,n} \circ \underline{(m \otimes n)}r'' \circ \phi_{m\otimes n,k} \rangle \\
= \quad & \text{(Definition of composition)} \\
& \langle \phi_{m,n}^{-1} \circ \underline{m}l' \circ l \mid r \circ \underline{m}r' \circ \phi_{m,n} \rangle \circ \langle l'' \mid r'' \rangle \\
= \quad & \text{(Definition of composition)} \\
& \langle l \mid r \rangle \circ \langle l' \mid r' \rangle \circ \langle l'' \mid r'' \rangle
\end{aligned}
$$

3.2 Lenses and prisms

The next step is to show that Definition 3.1.1 actually captures our motivating examples. We will need to make use of the Yoneda lemma to translate between the two forms of the optic. We go from the form of the optic that the definition prescribes (the *existential optic*) to their description after eliminating the coend (the *concrete optic*).

Proposition 3.2.1 ([Mil17])**.** *Lenses are optics for the cartesian product.*

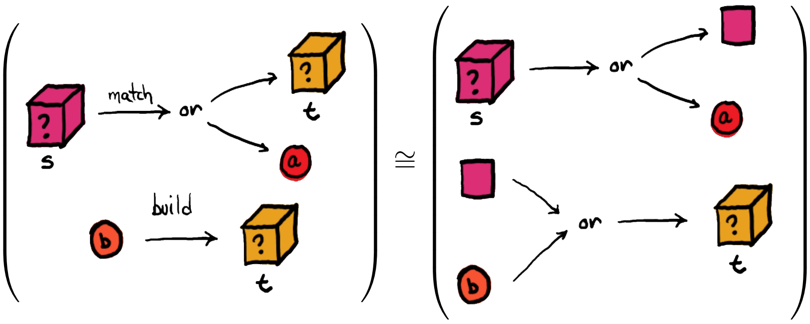


Proof.

$$
\begin{aligned}
&\int^{c \in \mathbf{C}} \mathbf{C}(s, c \times a) \times \mathbf{C}(c \times b, t) \\
\cong \quad & \text{(Product)} \\
&\int^{c \in \mathbf{C}} \mathbf{C}(s, c) \times \mathbf{C}(s, a) \times \mathbf{C}(c \times b, t) \\
\cong \quad & \text{(Yoneda lemma)} \\
&\mathbf{C}(s, a) \times \mathbf{C}(s \times b, t).
\end{aligned}
$$

□

Proposition 3.2.2 ([Mil17])**.** *Dually, prisms are optics for the coproduct.*



Proof.

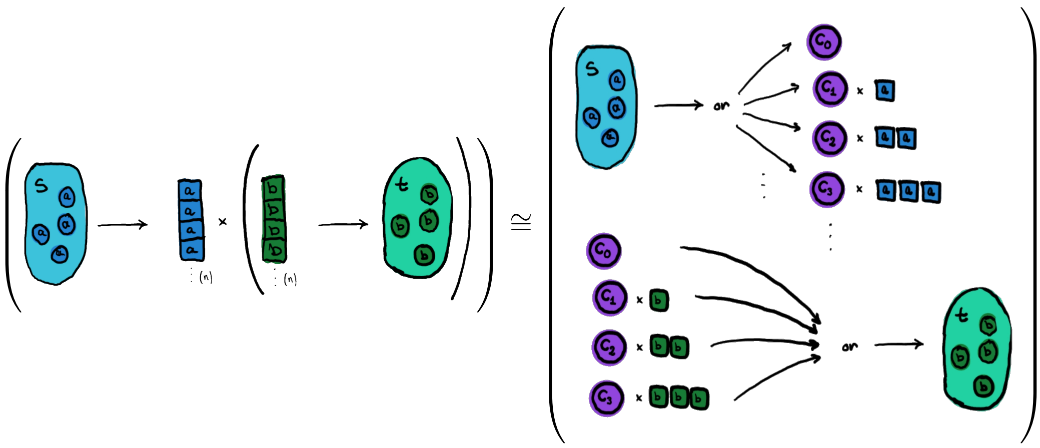
$$
\begin{aligned}
&\int^{m \in \mathbf{C}} \mathbf{C}(s, m + a) \times \mathbf{C}(m + b, t) \\
\cong \quad & \text{(Coproduct)} \\
&\int^{m \in \mathbf{C}} \mathbf{C}(s, m + a) \times \mathbf{C}(m, t) \times \mathbf{C}(m + b, t) \\
\cong \quad & \text{(Yoneda lemma)} \\
&\mathbf{C}(s, t + a) \times \mathbf{C}(b, t).
\end{aligned}
$$

□

3.3 Traversals

Given some functor $c \in [\mathbb{N}, \mathbf{C}]$ from the discrete category of the natural numbers, we can define a *power series* functor $F \colon \mathbf{C} \to \mathbf{C}$ given by $F(a) = \sum_{n \in \mathbb{N}} c_n \times a^n$. This induces a monoidal action that we call $\mathrm{Series} \colon [\mathbb{N}, \mathbf{C}] \to [\mathbf{C}, \mathbf{C}]$. The monoidal product for this action is given by substitution of morphisms, and it corresponds to the fact that two power series $F(a) = \sum_{n \in \mathbb{N}} c_n \times a^n$ and $G(a) = \sum_{n \in \mathbb{N}} d_n \times a^n$ can be composed into $GF(a) = \sum_{n \in \mathbb{N}} d_n \times \left(\sum_{m \in \mathbb{N}} c_m \times a^m\right)^n$, which can be seen again as a power series functor. A detailed description of this substitution can be found in [Koc09] or [Yor14] for the case of general combinatorial species.

Proposition 3.3.1. *Traversals are optics for the action* $\mathrm{Series} \colon [\mathbb{N}, \mathbf{C}] \to [\mathbf{C}, \mathbf{C}]$ *given by* *evaluation of the power series functor.*



Proof. Unfolding the definitions, the formula we want to prove is the following one.

$$\int^{c\in[\mathbb{N},\mathbf{C}]} \mathbf{C}\left(s, \sum\nolimits_n c_n \times a^n\right) \times \mathbf{C}\left(\sum\nolimits_n c_n \times b^n, t\right) \cong \mathbf{C}\left(s, \sum\nolimits_n a^n \times (b^n \to t)\right).$$

This is Yoneda, this time for functors $c\colon \mathbb{N} \to \mathbf{C}$. Note that, because we are taking the discrete category of the natural numbers, ends over this category are products, and a natural transformation $[\mathbb{N},\mathbf{C}](F,G)$ can be written as $\prod_{n\in\mathbb{N}} \mathbf{C}(F(n),G(n))$.

$$\begin{aligned}
&\int^{c} \mathbf{C}\left(s, \sum_{n\in\mathbb{N}} c_n \times a^n\right) \times \mathbf{C}\left(\sum_{n\in\mathbb{N}} c_n \times b^n, t\right)\\
\cong\quad &\text{(Cocontinuity)}\\
&\int^{c} \mathbf{C}\left(s, \sum_{n\in\mathbb{N}} c_n \times a^n\right) \times \prod_{n\in\mathbb{N}} \mathbf{C}\left(c_n \times b^n, t\right)\\
\cong\quad &\text{(Exponential)}\\
&\int^{c} \mathbf{C}\left(s, \sum_{n\in\mathbb{N}} c_n \times a^n\right) \times \prod_{n\in\mathbb{N}} \mathbf{C}\left(c_n, b^n \to t\right)\\
\cong\quad &\text{(Natural transformation as an end)}\\
&\int^{c} \mathbf{C}\left(s, \sum_{n\in\mathbb{N}} c_n \times a^n\right) \times [\mathbb{N},\mathbf{C}]\left(c_{(-)}, b^{(-)} \to t\right)\\
\cong\quad &\text{(Yoneda lemma)}\\
&\mathbf{C}\left(s, \sum_{n\in\mathbb{N}} a^n \times (b^n \to t)\right).
\end{aligned}$$

□

This derivation solves the problem posed in [Mil17] of finding a derivation of the Traversal fitting the same elementary pattern as the other optics described there. It should be noted, however, that derivations of the traversal as the optic for a certain kind of functors called **Traversables** (which should not be confused with traversals themselves) have been previously described by [BG18] and [Ril18]. For a derivation using Yoneda, [Ril18] recalls a parameterised adjunction that has an equational proof in [JO15]. The *Traversable* characterization is the one that was known and commonly used in programming libraries.

This characterization in terms of *power series polynomials* could be considered a more elementary description, although the profunctor description can give problems when implementing it in languages such as Haskell with only partial support for families of types indexed by natural numbers. The concrete description can be implemented as a *nested datatype* [BM98].

3.4 More examples of optics

3.4.1 Grates

In this section, we assume we are working with a bicartesian closed category $\mathbf{C}$ instead of detailing the precise requisites that we would need to make each one of these optics definable in the category.

Proposition 3.4.1 ([Mil17])**.** *Grates are optics for the action of the exponential* $(\to) \colon \mathbf{C}^{op} \to [\mathbf{C}, \mathbf{C}]$.

Proof.

$$
\begin{aligned}
& \int^{c \in \mathbf{C}^{op}} \mathbf{C}(s, c \to a) \times \mathbf{C}(c \to b, t) \\
\cong \quad & \text{(Exponential)} \\
& \int^{c \in \mathbf{C}^{op}} \mathbf{C}(c, s \to a) \times \mathbf{C}(c \to b, t) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \mathbf{C}((s \to a) \to b, t).
\end{aligned}
$$

□

3.4.2 Achromatic lenses

Proposition 3.4.2. ***Achromatic lenses*** *(described by [Boi17]) are optics for the action* $(1 + (-)) \times (-) \colon \mathbf{C} \times \mathbf{C} \to \mathbf{C}$. *They have a concrete description*

$$
\mathbf{AchrLens}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} s \\ t \end{pmatrix} \right) = \mathbf{C}(s, (b \to t) + 1) \times \mathbf{C}(s, a) \times \mathbf{C}(b, t).
$$

Proof. Again applying the Yoneda lemma.

$$
\begin{aligned}
& \int^{c \in \mathbf{C}} \mathbf{C}(s, (c+1) \times a) \times \mathbf{C}((c+1) \times b, t) \\
\cong \quad & \text{(Product)} \\
& \int^{c \in \mathbf{C}} \mathbf{C}(s, c+1) \times \mathbf{C}(s, a) \times \mathbf{C}((c+1) \times b, t) \\
\cong \quad & \text{(Distributivity)} \\
& \int^{c \in \mathbf{C}} \mathbf{C}(s, c+1) \times \mathbf{C}(s, a) \times \mathbf{C}(c \times b + b, t) \\
\cong \quad & \text{(Coproduct)} \\
& \int^{c \in \mathbf{C}} \mathbf{C}(s, c+1) \times \mathbf{C}(s, a) \times \mathbf{C}(b, t) \times \mathbf{C}(c \times b, t)
\end{aligned}
$$

$$\cong \quad \text{(Exponential)}$$

$$\int^{c \in \mathbf{C}} \mathbf{C}(s, c+1) \times \mathbf{C}(s, a) \times \mathbf{C}(b, t) \times \mathbf{C}(c, b \to t)$$

$$\cong \quad \text{(Yoneda lemma)}$$

$$\mathbf{C}(s, (b \to t) + 1) \times (s \to a) \times (b \to t). \qquad \square$$

3.4.3 Kaleidoscopes

Proposition 3.4.3. ***Kaleidoscopes*** *are optics for the evaluation of applicative functors (Definition 2.2.7),* $\mathbf{App} \to [\mathbf{Sets}, \mathbf{Sets}]$*. They have a concrete description*

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

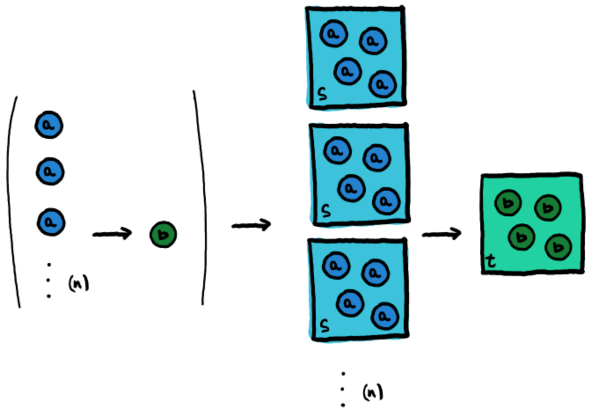


Figure 3.1: A big data structure s contains many substructures of type a; a way of folding them $(a^n \to b)$ gives a way of folding the big data structure $(s^n \to t)$.

Proof. We will make use of the construction of free applicatives given in Corollary 2.2.10. Note that we are implicitly applying a forgetful functor over the applicative F.

$$\int^{F \in \mathbf{App}} \mathbf{Sets}(s, Fa) \times \mathbf{Sets}(Fb, t)$$

$$\cong \quad \text{(Yoneda lemma)}$$

$$\int^{F \in \mathbf{App}} \mathbf{Sets}\left(s, \int_c (a \to c) \to Fc\right) \times \mathbf{Sets}(Fb, t)$$

$$\cong \quad \text{(Continuity)}$$

$$\int^{F \in \mathbf{App}} \left(\int_c \mathbf{Sets}\left(s, (a \to c) \to Fc\right)\right) \times \mathbf{Sets}(Fb, t)$$

$$\cong \quad \text{(Exponential)}$$

$$\int^{F \in \mathbf{App}} \left(\int_c \mathbf{Sets}\left(s \times (a \to c), Fc\right)\right) \times \mathbf{Sets}(Fb, t)$$

$$\cong \quad \text{(Natural transformations as ends)}$$

$$
\begin{aligned}
&\int^{F \in \mathbf{App}} \mathrm{Nat}\left(s \times (a \to (-)), F\right) \times \mathbf{Sets}(Fb, t) \\
\cong\quad & \text{(Free-forgetful adjunction for applicative functors)} \\
&\int^{F \in \mathbf{App}} \mathbf{App}\left(\sum_n s^n \times (a^n \to (-)), F\right) \times \mathbf{Sets}(Fb, t) \\
\cong\quad & \text{(Yoneda lemma)} \\
&\mathbf{Sets}\left(\sum_n s^n \times (a^n \to b), t\right).
\end{aligned}
$$

In **Sets**, we write this as $\prod_n (a^n \to b) \to (s^n \to t)$. □

3.4.4 Algebraic lenses

Remark 3.4.4. The action that defines lenses is not included in the action that gives kaleidoscopes because not every product by a set gives a lax monoidal functor. As we will see in §6.4, this implies that not every lens induces a kaleidoscope. However, when the object c is a monoid, the unit and multiplication induce functions $1 \to c \times 1$ and $(c \times a) \times (c \times b) \to c \times (a \times b)$, making products a particular example of applicative functor. This observation inspires the following optic.

Proposition 3.4.5. *Let ψ be a monad in a category $\mathbf{C}$. We consider the action of its algebras ψ-Alg given by forgetting about the algebra structure and taking the cartesian product. We know that the product of two algebras has again algebra structure and that the terminal object has an algebra structure: the forgetful functor from the Eilenberg-Moore category $U \colon \psi\text{-}Alg \to \mathbf{C}$ creates all limits that exist in $\mathbf{C}$.*

Proof. This gives a concrete optic we can call **algebraic lens**. Note that these are different from the *monadic lenses* studied in [ASCG+16].

$$
\begin{aligned}
&\int^{c \in \psi\text{-Alg}} \mathbf{C}(s, c \times a) \times \mathbf{C}(c \times b, t) \\
\cong\quad & \text{(Product)} \\
&\int^{c \in \psi\text{-Alg}} \mathbf{C}(s, c) \times \mathbf{C}(s, a) \times \mathbf{C}(c \times b, t) \\
\cong\quad & \text{(Free-forgetful adjunction for the algebras)} \\
&\int^{c \in \psi\text{-Alg}} \psi\text{-Alg}(\psi s, c) \times \mathbf{C}(s, a) \times \mathbf{C}(c \times b, t) \\
\cong\quad & \text{(Yoneda lemma)} \\
&\mathbf{C}(s, a) \times \mathbf{C}(\psi s \times b, t).
\end{aligned}
$$

□

In particular, taking ψ to be the *list* monad makes Remark 3.4.4 work appropiately. We call **ListLens** to this particular case of algebraic lens. *Coalgebraic prisms* work in exactly the same way.

3.4.5 Setters and adapters

Proposition 3.4.6. *For the case of setters, we will work in the category of* **Sets***.* ***Setters*** *are optics for the action given by evaluation of any endofunctor* $\mathrm{ev}\colon [\mathbf{Sets},\mathbf{Sets}] \to [\mathbf{Sets},\mathbf{Sets}]$*. They have a concrete description*

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

Proof.

$$\begin{aligned}
& \int^{F\in[\mathbf{Sets},\mathbf{Sets}]} \mathbf{Sets}(s, F(a)) \times \mathbf{Sets}(F(b), t) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int^{F\in[\mathbf{Sets},\mathbf{Sets}]} \mathbf{Sets}\left(s, \int_c ((a \to c) \to Fc\right) \times \mathbf{Sets}(F(b), t) \\
\cong \quad & \text{(Continuity)} \\
& \int^{F\in[\mathbf{Sets},\mathbf{Sets}]} \int_c \mathbf{Sets}\,(s \times (a \to c), Fc) \times \mathbf{Sets}(F(b), t) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int^{F\in[\mathbf{Sets},\mathbf{Sets}]} \mathrm{Nat}\,(s \times (a \to \Box), F) \times \mathbf{Sets}(F(b), t) \\
\cong \quad & \text{(Natural transformation as an end)} \\
& \mathbf{Sets}(s \times (a \to b), t).
\end{aligned}$$

□

In [Ril18], a similar derivation is given but in the more general case where we only ask our category to be powered and copowered over **Sets**.

Proposition 3.4.7. ***Adapters*** *are optics for the single action of the identity functor. By definition, they have a concrete description*

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

3.4.6 Generalized lens

The reader will notice that even when we were allowing the two parts of the optic to live in different categories, we are not using the construction in that generality. There are not many examples where *mixed* optics have an application, but we will show one example. The following generalization of the concept of lens was described to the author by David J. Myers [Mye], and similar generalizations have been described in [Spi19, §2.2]. They were described for the study of autopoietic systems and more generally, as a definition of lens that can work in a huge variety of different categories.

Definition 3.4.8. Let **C** be a symmetric monoidal category. A **generalized lens** in this monoidal category is a mixed optic for the action of the tensor product of cocommutative comonoids both on the (symmetric monoidal) category of cocommutative comonoids in

$\mathbf{C}$, that we call **Comon**; and in $\mathbf{C}$ itself. They have a concrete description given by the following formula, where we call $\mathcal{U} \colon \mathbf{Comon} \to \mathbf{C}$ to the forgetful functor.

$$\mathbf{GeneralizedLens}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} s \\ t \end{pmatrix}\right) = \mathbf{Comon}(s, a) \times \mathbf{C}(\mathcal{U}s \otimes b, t)$$

Proof. The main result we need to use is that a comonoid homomorphism between cocommutative comonoids $s \to c \otimes a$ can be split uniquely as the monoidal product of two comonoid homomorphisms $s \to c$ and $s \to a$.

$$\begin{aligned}
&\int_{c \in \mathbf{Comon}} \mathbf{Comon}(s, c \otimes a) \times \mathbf{C}(\mathcal{U}c \otimes b, t) \\
\cong &\qquad \text{(Split of the comonoid morphism)} \\
&\int_{c \in \mathbf{Comon}} \mathbf{Comon}(s, c) \times \mathbf{Comon}(s, a) \times \mathbf{C}(\mathcal{U}c \otimes b, t) \\
\cong &\qquad \text{(Yoneda lemma)} \\
&\mathbf{Comon}(s, a) \times \mathbf{C}(\mathcal{U}s \otimes b, t).
\end{aligned}$$

□

3.4.7 Optics for (co)free

Remark 3.4.9. In the derivation of the concrete Kaleidoscope (Proposition 3.4.3) we have only used the fact that we can generate free applicative functors. On the other hand, in the derivation of the Traversal on the next chapter (Proposition 4.2.1) we only use the fact that we can generate cofree traversable functors. These two observations can be generalized into a class of concrete optics. For a different but similar class of optics and their laws, see [Ril18, §4.4].

We start by considering the following two functors for some fixed $s, a \in \mathbf{Sets}$ and some fixed $b, t \in \mathbf{Sets}$.

$$L_{s,a} = s \times (a \to (-)), \qquad R_{t,b} = ((-) \to b) \to t.$$

The names come from their similarity to left and right Kan extensions, which will be justified from the fact that they arise from (co)Yoneda reductions. In fact, the property that is interesting to us is that for any given functor $H \colon \mathbf{Sets} \to \mathbf{Sets}$, the following isomorphisms hold.

$$\mathbf{Sets}(s, Ha) \cong [\mathbf{Sets}, \mathbf{Sets}](L_{s,a}, H), \qquad \mathbf{Sets}(Hb, t) \cong [\mathbf{Sets}, \mathbf{Sets}](H, R_{t,b}).$$

We can prove this again using the Yoneda lemma.

$$\begin{aligned}
&\mathbf{Sets}(s, Ha) \\
\cong &\qquad \text{(Yoneda lemma)} \\
&\mathbf{Sets}\left(s, \int_{c \in \mathbf{Sets}} (a \to c) \to Hc\right) \\
\cong &\qquad \text{(Continuity)} \\
&\int_{c \in \mathbf{Sets}} \mathbf{Sets}\left(s, (a \to c) \to Hc\right) \\
\cong &\qquad \text{(Exponential)}
\end{aligned}$$

$$\begin{aligned}
&\mathbf{Sets}(Hb, t) \\
\cong &\qquad \text{(Coyoneda lemma)} \\
&\mathbf{Sets}\left(\int^{c \in \mathbf{Sets}} Hc \times (c \to b), t\right) \\
\cong &\qquad \text{(Continuity)} \\
&\int_{c \in \mathbf{Sets}} \mathbf{Sets}\left(Hc \times (c \to b), t\right) \\
\cong &\qquad \text{(Exponential)}
\end{aligned}$$

$$\int_{c\in\mathbf{Sets}} \mathbf{Sets}\left(s \times (a \to c), Hc\right) \qquad \int_{c\in\mathbf{Sets}} \mathbf{Sets}\left(Hc, (c \to b) \to t\right)$$

$$\cong \quad \text{(Natural transformations as ends)} \qquad \cong \quad \text{(Natural transformations as ends)}$$

$$[\mathbf{Sets}, \mathbf{Sets}]\left(L_{s,a}, H\right). \qquad [\mathbf{Sets}, \mathbf{Sets}]\left(H, R_{b,t}\right).$$

Proposition 3.4.10. *Let a monoidal action* $U\colon \mathbf{M} \to [\mathbf{Sets}, \mathbf{Sets}]$ *have a left adjoint given by some* $F\colon [\mathbf{Sets}, \mathbf{Sets}] \to \mathbf{M}$*, that is,* $[\mathbf{Sets}, \mathbf{Sets}](f, Ug) \cong \mathbf{M}(Fg, f)$*. The optic determined by that monoidal action has a concrete form given by* $\mathbf{Sets}\left(UFL_{s,a}(b), t\right)$*. Dually, let it have a right adjoint given by some* $G\colon [\mathbf{Sets}, \mathbf{Sets}] \to \mathbf{M}$*, that is,* $[\mathbf{Sets}, \mathbf{Sets}](Uf, g) \cong \mathbf{M}(g, Gf)$*. The optic determined by that monoidal action has then a concrete form given by* $\mathbf{Sets}(s, UGR_{b,t}(a))$*.*

Proof.

$$\int^{f\in\mathbf{M}} \mathbf{Sets}(s, Uf(a)) \times \mathbf{Sets}(Uf(b), t) \qquad \int^{g\in\mathbf{M}} \mathbf{Sets}(s, Ug(a)) \times \mathbf{Sets}(Ug(b), t)$$

$$\cong \quad \text{(Definition of } L_{s,a}) \qquad \cong \quad \text{(Definition of } R_{b,t})$$

$$\int^{f\in\mathbf{M}} \mathrm{Nat}\left(L_{s,a}, Uf\right) \times \mathbf{Sets}(Uf(b), t) \qquad \int^{f\in\mathbf{M}} \mathbf{Sets}(s, Ug(a)) \times \mathrm{Nat}\left(Ug, R_{b,t}\right)$$

$$\cong \quad \text{(Adjunction)} \qquad \cong \quad \text{(Adjunction)}$$

$$\int^{f\in\mathbf{M}} \mathbf{M}\left(FL_{s,a}, f\right) \times \mathbf{Sets}(Uf(b), t) \qquad \int^{f\in\mathbf{M}} \mathbf{Sets}(s, Ug(a)) \times \mathbf{M}\left(g, GR_{b,t}\right)$$

$$\cong \quad \text{(Yoneda lemma)} \qquad \cong \quad \text{(Yoneda lemma)}$$

$$\mathbf{Sets}\left(UFL_{s,a}(b), t\right). \qquad \mathbf{Sets}\left(s, UGR_{b,t}a\right). \qquad \square$$

Chapter 4

Traversals

We have characterized traversals as the optic for *power series* functors in Proposition 3.3.1. However, the result that is usually presented and used in programming libraries is that traversals are the optic for *traversable functors*. Recall that we are taking power series functors to be these that can be written as $T(a) = \sum_n c_n \times a^n$ for some $c \colon \mathbb{N} \to \mathbf{Sets}$; while a functor T will be traversable if it has a distributive law $TF \Rightarrow FT$ for every applicative functor $F \in \mathbf{App}$ (Definition 2.2.7). We will show that both give rise to the optic called *traversal*.

4.1 Traversables as coalgebras

4.1.1 Traversable functors

Definition 4.1.1 ([JR12])**.** A **traversable** structure on a functor $T \colon \mathbf{C} \to \mathbf{C}$ is a family of transformations $\mathrm{trv}_F \colon TF \Rightarrow FT$ satisfying three additional rules called *naturality*, *unitarity* and *linearity;* which can be expressed respectively as the commutativity of the following diagrams, where $\alpha \colon F \Rightarrow G$ is a morphism of applicative functors.

$$\begin{tikzcd} TF \arrow[r, "\mathrm{trv}_F"] \arrow[d, "T\alpha"'] & FT \arrow[d, "\alpha"] \\ TG \arrow[r, "\mathrm{trv}_G"] & GT \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, bend left, "\mathrm{trv}_1"] \arrow[r, bend right, "\mathrm{id}"'] & T \end{tikzcd} \qquad \begin{tikzcd} TFG \arrow[r, "\mathrm{trv}_F"] \arrow[dr, "\mathrm{trv}_{FG}"'] & FTG \arrow[d, "F\,\mathrm{trv}_G"] \\ & FGT \end{tikzcd}$$

Remark 4.1.2*.* The first of the three rules is equivalent to a dinaturality condition over an end. With this in mind, we can define the traversable structure to be given by $\int_{F \in \mathbf{App}} (TF \Rightarrow FT)$ instead. Because of the definition of right Kan extensions, we can rewrite this as $T \Rightarrow \int_{F \in \mathbf{App}} \mathsf{Ran}_F FT$. This motivates our study of this particular end.

We will characterize traversable functors as coalgebras; the first step will be to simplify the end $\int_{F \in \mathbf{App}} \mathsf{Ran}_F FT$. The following lemma relies on the construction of a particular free applicative functor from Corollary 2.2.10. Recall that the free applicative functor over $(a \times (b \to -))$ is precisely

$$(a \times (b \to -))^* \cong \sum_{n \in \mathbb{N}} a^n \times (b^n \to -).$$

Lemma 4.1.3. *There exists an isomorphism with the following signature, natural on* $T\colon \mathbf{Sets} \to \mathbf{Sets}$*.*

$$\int_{F\in\mathbf{App}} \mathsf{Ran}_F FT(a) \cong \sum_{n\in\mathbb{N}} a^n \times T(n).$$

Proof. Note for this derivation that coproducts commute with connected limits (see for instance [nLa18]). Intuitively, if we need to choose a number for each set and it has to be preserved by morphisms, it needs to be constant.

$$
\begin{aligned}
& \int_{F\in\mathbf{App}} \mathsf{Ran}_F FT(a) \\
\cong \quad & \text{(Formula for a right Kan extension)} \\
& \int_{F\in\mathbf{App}} \int_{b\in\mathbf{Sets}} (a \to Fb) \to FT(b) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int_{F\in\mathbf{App}} \int_{b\in\mathbf{Sets}} \left(a \to \int_{c\in\mathbf{Sets}} (b \to c) \to Fc\right) \to FT(b) \\
\cong \quad & \text{(Continuity)} \\
& \int_{F\in\mathbf{App}} \int_{b\in\mathbf{Sets}} \left(\int_{c\in\mathbf{Sets}} a \to (b \to c) \to Fc\right) \to FT(b) \\
\cong \quad & \text{(Currying)} \\
& \int_{F\in\mathbf{App}} \int_{b\in\mathbf{Sets}} \left(\int_{c\in\mathbf{Sets}} a \times (b \to c) \to Fc\right) \to FT(b) \\
\cong \quad & \text{(Natural transformation as an end)} \\
& \int_{F\in\mathbf{App}} \int_{b\in\mathbf{Sets}} \mathrm{Nat}\left(a \times (b \to -), F\right) \to FT(b) \\
\cong \quad & \text{(Free-forgetful adjunction for applicative functors)} \\
& \int_{F\in\mathbf{App}} \int_{b\in\mathbf{Sets}} \mathbf{App}\left((a \times (b \to -))^*, F\right) \to FT(b) \\
\cong \quad & \text{(Free applicative functor)} \\
& \int_{F\in\mathbf{App}} \int_{b\in\mathbf{Sets}} \mathbf{App}\left(\sum_{n\in\mathbf{N}} a^n \times (b^n \to -), F\right) \to FT(b) \\
\cong \quad & \text{(Fubini)} \\
& \int_{b\in\mathbf{Sets}} \int_{F\in\mathbf{App}} \mathbf{App}\left(\sum_{n\in\mathbf{N}} a^n \times (b^n \to -), F\right) \to FT(b) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int_{b\in\mathbf{Sets}} \sum_{n\in\mathbf{N}} a^n \times (b^n \to T(b)) \\
\cong \quad & \text{(Ends distribute over discrete colimits)} \\
& \sum_{n\in\mathbf{N}} \int_{b\in\mathbf{Sets}} a^n \times (b^n \to T(b)) \\
\cong \quad & \text{(Fubini, as in Remark 2.1.6)}
\end{aligned}
$$

$$\sum_{n\in\mathbf{N}}\left(\int_{b\in\mathbf{Sets}} a\right)^n \times \left(\int_{b\in\mathbf{Sets}} b^n \to T(b)\right)$$

$\cong$ (Connected end over a constant functor)

$$\sum_{n\in\mathbf{N}} a^n \times \left(\int_{b\in\mathbf{Sets}} b^n \to T(b)\right)$$

$\cong$ (Exponential as function from a finite set)

$$\sum_{n\in\mathbf{N}} a^n \times \left(\int_{b\in\mathbf{Sets}} (n \to b) \to T(b)\right)$$

$\cong$ (Yoneda lemma)

$$\sum_{n\in\mathbf{N}} a^n \times T(n).$$

□

4.1.2 The shape-contents comonad

We will be studying the following higher-order functor $K\colon [\mathbf{Sets},\mathbf{Sets}] \to [\mathbf{Sets},\mathbf{Sets}]$ defined as $KT(a) = \sum_{n\in\mathbb{N}} T(n) \times a^n$. It is meant to represent a split between the shape and the contents of T, regarded as a container. Because of this, we write the elements of $KT(a)$ as $(n; s, c)$ with $n \in \mathbb{N}$ the *length*, $s \in T(n)$ the *shape*, and $c \in a^n$ the *contents*. The inspiration comes from [GdSO09], which mentions how traversables provide this kind of shape-contents split, studied in [JC94]; our goal is to show that it is precisely what characterizes them. One can see that, actually, the elements of $T(n)$ are more than the valid shapes; indexes could be repeated or not even present at all. We claim, however, that the coalgebra axioms are enough to ensure a valid *shape-contents* split.

Proposition 4.1.4. *There exists a functor $K\colon [\mathbf{Sets},\mathbf{Sets}] \to [\mathbf{Sets},\mathbf{Sets}]$ defined on objects by $KT = \sum_{n\in\mathbb{N}} T(n) \times a^n$ that can be given a comonad structure.*

Proof. First, we check that it is indeed a well-defined functor. If we want it to be well-defined on objects, we need to check that $\sum_{n\in\mathbb{N}} T(n) \times a^n$ is a functor for any $T \in [\mathbf{Sets},\mathbf{Sets}]$. Given $f\colon a \to b$, the corresponding $KT(f)\colon KT(a) \to KT(b)$ is defined by $KT(f)(n; s, c) = (n; s, f \circ c)$. We can see that this is functorial. Now we need to define its action on morphisms. Given any natural transformation $h\colon T \Rightarrow R$, we can define $Kh(n; s, c) = (n; h(s), c)$, which is also functorial.

Now we define the *counit* $\varepsilon\colon KT \Rightarrow T$ as the morphism given by evaluation $T(n)\times a^n \to T(a)$ in every possible $n \in \mathbb{N}$. In other words, $\varepsilon(n; s, c) = T(c)(s)$. The *comultiplication* $\delta\colon KT \Rightarrow K^2T$ is given by the universal property of the coproduct on the following diagram.

$$\begin{tikzcd}
\sum_n T(n) \times a^n \arrow[r, dashed, "\exists!"] & \sum_m \left(\sum_l T(l) \times m^l\right) \times a^m \\
T(n) \times a^n \arrow[u, "i_n"] \arrow[r, "{(\mathrm{id},(\mathrm{id}),\mathrm{id})}"] & T(n) \times n^n \times a^n \arrow[u, "i_{n,n}"']
\end{tikzcd}$$

That is, we choose both m and l to be n and then we use the identity. In other words, $\delta(n; s, c) = (n; (n; s, \mathrm{id}), c)$.

We will check now counitality and coassociativity. For *counitality*, we need the following two diagrams to commute.

$$\begin{tikzcd}
\sum_n T(n) \times a^n & \sum_m \left(\sum_l T(l) \times m^l\right) \times a^m \arrow[l, "K\varepsilon_T"'] \arrow[r, "\varepsilon_{KT}"] & \sum_n T(n) \times a^n \\
& \sum_n T(n) \times a^n \arrow[ul, "\mathrm{id}"] \arrow[u, "\delta_T"'] \arrow[ur, "\mathrm{id}"'] &
\end{tikzcd}$$

They do commute because of the following chains of equations that arise from unfolding the definitions.

$$\begin{aligned}
& K\varepsilon_T(\delta(n;s,c)) \\
= & \quad \text{(Definition of } \delta) \\
& K\varepsilon_T(n;(n,s,\mathrm{id}),c) \\
= & \quad \text{(Definition of } K) \\
& (n;\varepsilon_T(n,s,\mathrm{id}),c) \\
= & \quad \text{(Definition of } \varepsilon) \\
& (n;T(\mathrm{id})(s),c) \\
= & \quad \text{(Identity)} \\
& (n;s,c).
\end{aligned}
\qquad\qquad
\begin{aligned}
& \varepsilon_{KT}(\delta(n;s,c)) \\
= & \quad \text{(Definition of } \delta) \\
& \varepsilon_{KT}(n;(n;s,\mathrm{id}),c) \\
= & \quad \text{(Definition of } \varepsilon) \\
& KTc(n;s,\mathrm{id}) \\
= & \quad \text{(Definition of } KT) \\
& (n;s,c \circ \mathrm{id}) \\
= & \quad \text{(Identity)} \\
& (n;s,c).
\end{aligned}$$

Coassociativity is the fact that the following diagram commutes.

$$\begin{tikzcd}
\sum_n \left(\sum_m \left(\sum_l T(l) \times m^l\right) \times n^m\right) \times a^n & \sum_n \left(\sum_l T(l) \times n^l\right) \times a^n \arrow[l, "K\delta_T"'] \\
\sum_n \left(\sum_m T(m) \times n^m\right) \times a^n \arrow[u, "\delta_{KT}"] & \sum_n T(n) \times a^n \arrow[l, "\delta_T"] \arrow[u, "\delta_T"']
\end{tikzcd}$$

It does because of the following chain of equations, again following the definitions.

$$\begin{aligned}
& \delta_{KT}(\delta_T(n;s,c)) \\
= & \quad \text{(Definition of } \delta) \\
& \delta_{KT}(n;(n;s,\mathrm{id}),c) \\
= & \quad \text{(Definition of } \delta) \\
& (n;(n;(n;s,\mathrm{id}),\mathrm{id}),c) \\
= & \quad \text{(Definition of } \delta) \\
& (n;\delta_T(n;s,\mathrm{id}),c) \\
= & \quad \text{(Definition of } K) \\
& K\delta_T(n;(n;s,\mathrm{id}),c) \\
= & \quad \text{(Definition of } \delta) \\
& K\delta_T(\delta_T(n;s,\mathrm{id})).
\end{aligned}$$

This finishes the construction of a comonad over K. □

The coalgebra axioms follow from the structure, but we will write them explicitly and comment on them. Let $\sigma\colon Ta \to \sum_n Tn \times a^n$ be a coalgebra. The first axiom is *counitality*, and in our case, it says that the following diagram commutes.

$$\begin{tikzcd} Ta \arrow[r, "\sigma"] \arrow[dr, "\mathrm{id}"'] & \sum_n Tn \times a^n \arrow[d, "\varepsilon"] \\ & Ta \end{tikzcd}$$

When $\sigma(t) = (n; s, c)$, we have that $Tc(s) = t$. This is to say that, if we split into shape and contents and then we put back the contents onto the shape, we should get back our original structure. The second axiom is *coassociativity*, and in our case, it says that the following diagram commutes.

$$\begin{tikzcd} Ta \arrow[r, "\sigma"] \arrow[d, "\sigma"'] & \sum_n Tn \times a^n \arrow[d, "K\sigma"] \\ \sum_n Tn \times a^n \arrow[r, "\delta"] & \sum_n \left(\sum_m Tm \times n^m\right) \times a^n \end{tikzcd}$$

When $\sigma(t) = (n; c, s)$, we have that $\sigma(s) = (n; \mathrm{id}, s)$. This is to say that the shape of a shape s is again s. In this sense, taking the shape is idempotent.

4.1.3 Linearity and unitarity from coalgebra laws

We will show that traversables can be defined equivalently as coalgebras for the shape-contents comonad. This definition feels intuitive to us: traversables are precisely functors equipped with a split into shape and contents.

Theorem 4.1.5. *A coalgebra for the shape-contents comonad is a traversal.*

Proof. Because of Lemma 4.1.3, we already have a bijection between natural transformations $T \Rightarrow \sum_n Tn \times (-)^n$ and natural transformations $T \Rightarrow \int_{F \in \mathbf{App}} \mathsf{Ran}_F FT$. We have already shown that $KT = \sum_n T(n) \times (-)^n$ acts as a comonad; we will show that linearity and unity follow from the coalgebra laws.

We first show that there exists a family $n_{T,F} \colon KT \circ F \Rightarrow F \circ KT$ natural in both $T \in [\mathbf{Sets}, \mathbf{Sets}]$ and $F \in \mathbf{App}$. In fact, we can use the multiplication of F and the fact that every functor is lax monoidal with respect to the coproduct to construct the following map and check that it is natural. Let $\alpha \colon F \Rightarrow G$ be a morphism of applicatives, which must preserve the multiplication and unit, and make the first and second squares commute. The third square commutes because of naturality of α.

$$\begin{tikzcd} \sum_n (Fa)^n \times Tn \arrow[r, "u_{F,Tn}"] \arrow[d, "\alpha"] \arrow[rrr, bend left, "n_{T,F}"] & \sum_n F(a^n) \times FTn \arrow[r, "w_F"] \arrow[d, "\alpha"] & \sum_n F(a^n \times Tn) \arrow[r] \arrow[d, "\alpha"] & F\left(\sum_n a^n \times Tn\right) \arrow[d, "\alpha"] \\ \sum_n (Ga)^n \times Tn \arrow[r, "u_{G,Tn}"] \arrow[rrr, bend right, "n_{T,G}"'] & \sum_n G(a^n) \times GTn \arrow[r, "w_G"] & \sum_n G(a^n \times Tn) \arrow[r] & G\left(\sum_n a^n \times Tn\right) \end{tikzcd}$$

We define now $m_{T,F} = F\varepsilon_T \circ n_{T,F} \colon KT \circ F \Rightarrow F \circ T$, and the traverse of our functor T will be $\mathrm{trv}_{T,F} = m_{T,F} \circ \sigma_T \colon T \circ F \Rightarrow F \circ T$.

We now prove *unitarity* from the counitality axiom $\varepsilon \circ \sigma = \mathrm{id}$. The following is the relevant diagram, showing how they both imply each other. Note that $n_{T,\mathrm{id}} = \mathrm{id}$.

$$\begin{tikzcd}
T \circ \mathrm{id} \arrow[r, "\sigma_T"] & KT \circ \mathrm{id} \arrow[rr, "m_{T,\mathrm{id}}"] \arrow[dr, "n_{T,\mathrm{id}}=\mathrm{id}"'] & & \mathrm{id} \circ T \\
& & \mathrm{id} \circ KT \arrow[ur, "\varepsilon_T"'] &
\end{tikzcd}$$

We now prove *linearity* from the coalgebra axiom $K\sigma \circ \sigma = \delta \circ \sigma$. In order to do that, we will simplify the two sides of the linearity equation to make them match coassociativity. The first side of the equation can be simplified as follows.

$$\begin{tikzcd}
T \circ F \circ G \arrow[r, "\sigma"] \arrow[rr, bend left, "\mathrm{trv}_{F\circ G}"] & KT \circ F \circ G \arrow[r, "m_F"] \arrow[d, "K\sigma"] & F \circ T \circ G \arrow[d, "F\sigma"] \arrow[dd, bend left, "F\,\mathrm{trv}_F"] \\
& K^2T \circ F \circ G \arrow[r, "m_F"] & F \circ KT \circ G \arrow[d, "m_G"] \\
& & F \circ G \circ T
\end{tikzcd}$$

The second side of the equation can be also simplified. In order to do this, we first write $m_{F\circ G}$ in terms of m_F and m_G. Applicative functors are composed composing their units and multiplications; this makes $n_{FG} = Fn_G \circ n_F$.

$$\begin{aligned}
& m_{F\circ G} \\
= \quad & \text{(Definition of } m\text{)} \\
& FG\varepsilon_T \circ n_{F\circ G} \\
= \quad & \text{(Using that } n_{FG} = Fn_G \circ n_F\text{)} \\
& FG\varepsilon_T \circ Fn_G \circ n_F \\
= \quad & \text{(Functoriality)} \\
& F(G\varepsilon_T \circ n_G) \circ n_F \\
= \quad & \text{(Definition of } m\text{)} \\
& Fm_G \circ n_F \\
= \quad & \text{(Counitality for the algebra)} \\
& Fm_G \circ F\varepsilon \circ F\delta \circ n_F \\
= \quad & \text{(Naturality of } n\text{)} \\
& Fm_G \circ F\varepsilon \circ n_F \circ \delta \\
= \quad & \text{(Definition of } m\text{)} \\
& Fm_G \circ m_F \circ \delta.
\end{aligned}$$

We now proceed to simplify the diagram with the other side of the linearity equation as follows.

$$\begin{tikzcd}
T \circ F \circ G \arrow[r, "\sigma"] \arrow[rr, bend left, "\mathrm{trv}_{F\circ G}"] & KT \circ F \circ G \arrow[r, "m_{FG}"] \arrow[d, "\delta"'] & F \circ G \circ T \\
& K^2T \circ F \circ G \arrow[r, "m_F"] & F \circ KT \circ G \arrow[u, "m_G"']
\end{tikzcd}$$

We can see now that coassociativity implies linearity. In fact, when we use $\delta \circ \sigma = K\sigma \circ \sigma$, we get the desired result.

Let us show now that linearity implies coassociativity. Let $a, b \in \mathbf{C}$ and consider the functor $L_{a,b} = a \times (b \to (-))$ we described in §3.4.7 and its free applicative $A_{x,y} = \sum_n x^n \times (y^n \to (-))$. Note that we have a trivial monomorphism $i \colon a \to A_{a,b}(b)$. It can be checked that $A_{a,b}T(b) \cong KT(a)$, but also that the following diagram commutes.

$$\begin{tikzcd}
KT(a) \arrow[r, "i"] \arrow[dr, "\cong"'] & KT(A_{a,b}(b)) \arrow[d, "n_{R_{a,b}}"] \\
 & R_{a,b}T(b)
\end{tikzcd}$$

By linearity, the following diagram commutes. Note that the internal square does not necessarily commute yet, but we will show that coassociativity follows from this.

$$\begin{tikzcd}
TA_{a,b}(b) \arrow[r, "i"] & TA_{a,b}A_{b,c}(c) \arrow[r, "\sigma"] \arrow[d, "\sigma"] & KTA_{a,b}A_{b,c}(c) \arrow[d, "\delta"] & A_{a,b}A_{b,c}T(c) \\
T(a) \arrow[u, "i"] & KTA_{a,b}A_{b,c}(c) \arrow[r, "\sigma"] & K^2TA_{a,b}A_{b,c}(c) \arrow[r, "n_{A_{a,b}}"] & A_{a,b}KTA_{b,c}(c) \arrow[u, "n_{A_{b,c}}"']
\end{tikzcd}$$

Naturality allows us to rewrite this into the following diagram. Because the last part of the diagram is an isomorphism, the internal square commutes.

$$\begin{tikzcd}
T(a) \arrow[r, "\sigma"] \arrow[d, "\sigma"] & KT(a) \arrow[d, "\delta"] & & \\
KT(a) \arrow[r, "\sigma"] & K^2T(a) \arrow[d, "i"] & & \\
 & \int_b K^2TA_{a,b}(b) \arrow[r, "n"] & \int_b A_{a,b}KT(b) \arrow[d, "i"] & \\
 & & \int_{b,c} A_{a,b}KTA_{b,c}(c) \arrow[r, "n"] & \int_{b,c} A_{a,b}A_{b,c}T(c)
\end{tikzcd}$$

□

Remark 4.1.6. At the moment it is not clear to us that the isomorphism constructed in Lemma 4.1.3 is precisely the one used to construct the traversal from the coalgebra in Theorem 4.1.5. The main idea here is that Lemma 4.1.3 should be a morphism of comonads, and linearity and unitarity should correspond (in general) to the comonad axioms of the left hand side. We assume this for §4.2; see also the Appendix 8.2.

Relating traversables and coalgebras makes us consider *cofree traversables* given by KH for an arbitrary functor H. If we consider coalgebra morphisms between traversables to define a category $\mathbf{Trv}$, we have the adjunction $\mathbf{Trv}(T, KH) \cong [\mathbf{Sets}, \mathbf{Sets}](T, H)$, see Appendix 8.2.

4.2 Traversals as the optic for traversables

The definition of traversables as coalgebras and the construction of cofree traversables also provides a new description of the traversals as optics for the evaluation of traversable functors. This is the result widely used by optic libraries to provide a profunctor description of the traversal.

Proposition 4.2.1. *In* **Sets**, *the traversal is the optic for traversable functors.*

Proof.

$$
\begin{aligned}
& \int^{T \in \mathbf{Trv}} \mathbf{Sets}(s, Ta) \times \mathbf{Sets}(Tb, t) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int^{T \in \mathbf{Trv}} \mathbf{Sets}(s, Ta) \times \mathbf{Sets}\left(\int^{c} Tc \times (c \to b), t \right) \\
\cong \quad & \text{(Continuity)} \\
& \int^{T \in \mathbf{Trv}} \mathbf{Sets}(s, Ta) \times \int_{c} \mathbf{Sets}\left(Tc \times (c \to b), t \right) \\
\cong \quad & \text{(Exponential)} \\
& \int^{T \in \mathbf{Trv}} \mathbf{Sets}(s, Ta) \times \int_{c} \mathbf{Sets}\left(Tc, (c \to b) \to t \right) \\
\cong \quad & \text{(Natural transformation as an end)} \\
& \int^{T \in \mathbf{Trv}} \mathbf{Sets}(s, Ta) \times \mathrm{Nat}\left(T, ((-) \to b) \to t \right) \\
\cong \quad & \text{(Cofree traversable)} \\
& \int^{T \in \mathbf{Trv}} \mathbf{Sets}(s, Ta) \times \mathbf{Trv}\left(T, \sum_{n} (-)^n \times (b^n \to t) \right) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \mathbf{Sets}\left(s, \textstyle\sum_{n} a^n \times (b^n \to t) \right)
\end{aligned}
$$

□

4.3 Species

A final observation on power series functors P is that they are precisely the left Kan extensions of some family of sets indexed by the natural numbers $S \colon \mathbb{N} \to \mathbf{Sets}$ over the inclusion $i \colon \mathbb{N} \to \mathbf{Sets}$ of every natural number as a set with that cardinality.

$$
\begin{tikzcd}
\mathbb{N} \arrow[r, "S"] \arrow[d, hook, "i"'] & \mathbf{Sets} \\
\mathbf{Sets} \arrow[ur, "P"'] &
\end{tikzcd}
$$

The formula for left Kan extensions (Proposition 2.1.12) gives us $P(a) = \sum_{n \in \mathbb{N}} a^n \times S(n)$, where the fact that the category is discrete turns the end into a sum. *Linear species*, as in [BLL97], are described on this way, just substituting natural numbers for an equivalent category given by linearly ordered sets with monotone bijections. *Combinatorial species* (or just *species*) generalize these. This motivates the idea of extending our discussion of traversals to arbitrary species.

We follow [Yor14] in discussing combinatorial species. A **combinatorial species** is a copresheaf on the groupoid of finite sets with bijections, $F \colon \mathbf{B} \to \mathbf{Sets}$. In other words, we are assigning a set of shapes to any finite set of labels, in such a way that for any bijection that we apply to the labels we get back a bijection for the shapes.

Definition 4.3.1 (Joyal, 1986)**. Analytic functors** are those that arise as the left Kan extension of a combinatorial species $F\colon \mathbf{B} \to \mathbf{Sets}$ along the inclusion $\mathbf{B} \to \mathbf{Sets}$.

$$\begin{tikzcd}
\mathbf{B} \arrow[r, "F"] \arrow[d, hook, "i"'] & \mathbf{Sets} \\
\mathbf{Sets} \arrow[ur, "\hat{F}"'] &
\end{tikzcd}$$

The formula for left Kan extensions gives us $\hat{F}(a) = \int^{l \in \mathbf{B}} (il \to a) \times F(l)$. Note that taking a coend over the category $\mathbf{B}$ instead of over a discrete category means we need to quotient by the group of permutations.

$$\hat{F}(a) = \int^{n \in \mathbf{B}} F(n) \times a^n$$

Following the analogy with traversals, we can define the **unsorted traversal** as the optic associated with the evaluation of analytic functors. Note that we have an operation that composes species described in [Yor14] and that the inclusion is unital with respect to this operation.

Proposition 4.3.2. *The unsorted traversal has a concrete form given by*

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

Proof. We will make use of the adjunction $\mathbf{Sets}(\hat{F}b, t) \cong \mathbf{Species}(F, b^{(-)} \to t)$ described by [Yor14, §6.4]. Note also that the analytic functor for the species $b^{(-)} \to t$ is given by $\int^n (-)^n \times (b^n \to t)$.

$$\begin{aligned}
& \int^{F \in \mathbf{Species}} \mathbf{Sets}(s, \hat{F}a) \times \mathbf{Sets}(\hat{F}b, t) \\
\cong & \quad \text{(Adjunction in [Yor14])} \\
& \int^{F \in \mathbf{Species}} \mathbf{Sets}(s, \hat{F}a) \times \mathbf{Species}(F, b^{(-)} \to t) \\
\cong & \quad \text{(Yoneda, and analytic functor for the species)} \\
& \mathbf{Sets}\left(s, \int^{n \in \mathbf{B}} a^n \times (b^n \to t) \right).
\end{aligned}$$

□

Chapter 5

Profunctor optics

Tambara modules were first described in [Tam06] for the case of monoidal categories. Fixing a monoidal category $\mathbf{C}$, Tambara modules are structures on top of the endoprofunctors $\mathbf{Prof}(\mathbf{C})$. It was shown in [PS08] that a category of this structures with morphisms preserving them in some suitable sense is equivalent to the copresheaves of some category called there the *double* of a monoidal category.

Our interest in Tambara modules and their characterization comes from the fact that profunctor optics are functions parametric precisely over Tambara modules. In our case, these *doubles* are categories of optics. We first provide a proof that these structures are the coalgebras for a comonad also described in [PS08]. From this result, we directly get the profunctor representation theorem, which relates optics in existential form to their profunctor form.

5.1 Tambara modules

During this section we fix a monoidal category $\mathbf{M}$ that acts both on two arbitrary categories $\mathbf{C}$ and $\mathbf{D}$. We write $\underline{m}$ for the image of $m \in \mathbf{M}$ both in $[\mathbf{C}, \mathbf{C}]$ and $[\mathbf{D}, \mathbf{D}]$. We write ϕ for the structure isomorphisms of these strong monoidal actions.

Definition 5.1.1. A **Tambara module** consists of a profunctor $p \colon \mathbf{C}^{op} \times \mathbf{D} \to \mathbf{Sets}$ endowed with a family of morphisms $\alpha_m \colon p(a, b) \to p(\underline{m}a, \underline{m}b)$ natural in both $a \in \mathbf{C}$ and $b \in \mathbf{D}$, and dinatural in $m \in \mathbf{M}$; which additionally satisfy the two equations $\alpha_i = p(\phi_i^{-1}, \phi_i)$ and $\alpha_{m \otimes n} = p(\phi_{m,n}^{-1}, \phi_{m,n}) \circ \alpha_m \circ \alpha_n$.

$$\begin{tikzcd} p(a,b) \arrow[r, "\alpha_i"] \arrow[dr, "\mathrm{id}"'] & p(\underline{i}a, \underline{i}b) \arrow[d, "p(\phi^{-1},\phi)"] \\ & p(a,b) \end{tikzcd} \qquad \begin{tikzcd} p(a,b) \arrow[r, "\alpha_n"] \arrow[drr, "\alpha_{n\otimes m}"'] & p(\underline{n}a, \underline{n}b) \arrow[r, "\alpha_m"] & p(\underline{mn}a, \underline{mn}b) \arrow[d, "p(\phi^{-1},\phi)"] \\ & & p(\underline{m \otimes n}a, \underline{m \otimes n}b) \end{tikzcd}$$

Remark 5.1.2. The definition of Tambara module in [PS08] deals only with actions that arise from a monoidal product $\otimes \colon \mathbf{C} \to [\mathbf{C}, \mathbf{C}]$. We have decided to use the term *Tambara module* also for the more general concept, instead of introducing new nomenclature. This generalization also includes *mixed optics*, that were proposed by [Ril18] as further work.

Definition 5.1.3. In the same way that we introduced よ to represent the Yoneda embedding, we will write the hiragana "ta", た, to refer to Tambara modules. Let た be the category of Tambara modules with morphisms $(p,\alpha) \to (q,\alpha')$ given natural transformations $\eta\colon p \Rightarrow q$ that satisfy $\eta_{a,b} \circ \alpha_{m,a,b} = \alpha'_{m,a,b} \circ \eta_{\underline{m}a,\underline{m}b}$. Diagrammatically, these are transformations such that the following diagram commutes.

$$\begin{tikzcd}
p(a,b) \arrow[r, "\alpha_m"] \arrow[d, "\eta_{a,b}"'] & p(\underline{m}a,\underline{m}b) \arrow[d, "\eta_{\underline{m}a,\underline{m}b}"] \\
q(a,b) \arrow[r, "\alpha'_m"] & q(\underline{m}a,\underline{m}b)
\end{tikzcd}$$

5.2 The Pastro-Street comonad

Tambara modules are equivalently coalgebras for a comonad studied in [PS08]. That comonad has a left adjoint that must therefore be a monad, and Tambara modules can be also characterized as algebras for that monad. We will get the category of Tambara modules た as an Eilenberg-Moore category, and this will be the main lemma towards the profunctor representation theorem.

Definition 5.2.1. We define $\Theta\colon \mathbf{Prof}(\mathbf{C},\mathbf{D}) \to \mathbf{Prof}(\mathbf{C},\mathbf{D})$ as

$$\Theta(p)(a,b) = \int_{m\in\mathbf{M}} p(\underline{m}a,\underline{m}b).$$

This is a comonad.

Proof. We start by showing that it is indeed a functor. We have already defined its action on objects, so we proceed to define its action on morphisms. Given two profunctors $p,q \in \mathbf{Prof}(\mathbf{C},\mathbf{D})$, let $\eta\colon p \Rightarrow q$ be a natural transformation. The following diagram constructs a wedge for some $h\colon m \to n$ that in turns defines a unique map $\int_m p(ma,mb) \to \int_m q(ma,mb)$. The map is natural in $a \in \mathbf{C}$ and $b \in \mathbf{D}$, as it is composed of natural maps. That gives a natural transformation $\Theta p \Rightarrow \Theta q$. Squares commute because of naturality of η and dinaturality of the coend in p.

$$\begin{tikzcd}
 & & \int_m p(\underline{m}a,\underline{m}b) \arrow[dll, "\pi_m"'] \arrow[dr, "\pi_n"] \arrow[ddd, dashed, "\Theta\eta_{a,b}"] & \\
p(\underline{m}a,\underline{m}b) \arrow[dr, "p(\mathrm{id},\underline{h})"] \arrow[ddd, "\eta_{ma,mb}"] & & & p(\underline{n}a,\underline{n}b) \arrow[dll, "p(h,\mathrm{id})"'] \arrow[ddd, "\eta_{na,nb}"] \\
 & p(\underline{m}a,\underline{n}b) \arrow[ddd, "\eta_{ma,nb}"] & & \\
 & & \int_m q(\underline{m}a,\underline{m}b) \arrow[dll, "\pi_m"'] \arrow[dr, "\pi_n"] & \\
q(\underline{m}a,\underline{m}b) \arrow[dr, "q(\mathrm{id},h)"'] & & & q(\underline{n}a,\underline{n}b) \arrow[dll, "q(h,\mathrm{id})"] \\
 & q(\underline{m}a,\underline{n}b) & &
\end{tikzcd}$$

Functoriality follows from the fact that, when $\eta = \mathrm{id}$, the identity map makes the diagram commute; and from the fact that, when $\eta = \eta_2 \circ \eta_1$, the composite $\Theta\eta_2 \circ \Theta\eta_1$ makes the diagram commute. More abstractly, we are using the functoriality of the coends and the naturality to lift the natural transformation.

We proceed to describe the components of the comonad. The *counit* is $\varepsilon_p = p(\phi^{-1}, \phi) \circ \pi_i$, defined by projecting on the monoidal unit component.

$$\int_{m\in\mathbf{M}} p(\underline{m}a, \underline{m}b) \xrightarrow{\pi_i} p(\underline{i}a, \underline{i}b) \xrightarrow{p(\phi^{-1},\phi)} p(a, b)$$

The *comultiplication* δ_p is obtained by the universal property of the end as the unique morphism making the following diagram commute.

$$\begin{tikzcd}
p(\underline{m \otimes n}a, \underline{m \otimes n}b) \arrow[d, "p(\phi^{-1},\phi)"'] & \int_{m\in\mathbf{M}} p(\underline{m}a, \underline{m}b) \arrow[l, "\pi_{m\otimes n}"'] \arrow[d, dashed, "\exists!\delta_p"] \\
p(\underline{mn}a, \underline{mn}b) & \int_{n\in\mathbf{M}} \int_{m\in\mathbf{M}} p(\underline{mn}a, \underline{mn}b) \arrow[l, "\pi_m \circ \pi_n"']
\end{tikzcd}$$

We will show now that it is indeed a comonad, proving counitality and coassociativity. *Counitality*, $\Theta\varepsilon \circ \delta = \mathrm{id}$, follows from commutativity of the following diagram. We use our definitions and the coherence of the end. The other side of counitality, $\varepsilon \circ \delta = \mathrm{id}$, is similar.

$$\begin{tikzcd}
p(\underline{i \otimes u}a, \underline{i \otimes u}a) \arrow[d] & & \int_m p(\underline{m}a, \underline{m}b) \arrow[ll, "\pi_{i\otimes u}"'] \arrow[d, "\delta"] \\
p(\underline{iu}a, \underline{iu}b) \arrow[d] & \int_m p(\underline{m}ua, \underline{m}ub) \arrow[l, "\pi_i"] & \int_n \int_m p(\underline{mn}a, \underline{mn}b) \arrow[l, "\pi_u"] \arrow[d, "\Theta\varepsilon"] \\
p(ua, ub) & & \int_n p(\underline{n}a, \underline{n}b) \arrow[ll, "\pi_u"]
\end{tikzcd}$$

Coassociativity, $\Theta\delta \circ \delta = \delta \circ \delta$, follows from commutativity of the following diagram. The internal squares commute by definition and coherence of the action. Finally, the two outermost morphisms are the same because of coherence of the first coend.

$$\begin{tikzcd}
& \int_m p(\underline{m}a, \underline{m}b) \arrow[dl, "\delta"] \arrow[dr, "\delta"] & \\
\int_n \int_m p(\underline{mn}a, \underline{mn}b) \arrow[r, "\Theta\delta"'] \arrow[d, "\pi"] & \int_o \int_n \int_m p(\underline{mno}a, \underline{mno}b) \arrow[d, "\pi"] & \int_n \int_m p(\underline{mn}a, \underline{mn}b) \arrow[l, "\delta"] \arrow[dd, "\pi"] \\
\int_m p(\underline{mw}a, \underline{mw}b) \arrow[r] \arrow[dd, "\pi"] & \int_n \int_m p(\underline{mnw}a, \underline{mnw}b) \arrow[d, "\pi"] & \\
& \int_m p(\underline{mvw}a, \underline{mvw}b) \arrow[d, "\pi"] & \int_m p(\underline{m(v \otimes w)}a, \underline{m(v \otimes w)}b) \arrow[l] \arrow[d, "\pi"] \\
p(\underline{(u \otimes v)w}a, \underline{(u \otimes v)w}b) \arrow[r] \arrow[dr] & p(\underline{uvw}a, \underline{uvw}b) \arrow[d, "\pi"] & p(\underline{u(v \otimes w)}a, \underline{u(v \otimes w)}b) \arrow[l, leftrightarrow] \arrow[dl] \\
& p(\underline{u \otimes v \otimes w}a, \underline{u \otimes v \otimes w}b) &
\end{tikzcd}$$

$\square$

Proposition 5.2.2. *Tambara modules are equivalently coalgebras for this comonad. The category た is equivalent, with a bijective-on-objects functor, to the Eilenberg-Moore category of* Θ.

Proof. Note that the data for a coalgebra is a natural transformation $\alpha\colon p \Rightarrow \Theta p$ whose projections are components of the Tambara module, $\alpha_{m,a,b} = \pi_m \circ \alpha_{a,b}$. The naturality of α is exactly the naturality of the components of the Tambara module; the coherence conditions of the end are precisely the dinaturality of the components. The only thing we need to show is that the coalgebra axioms correspond with the Tambara axioms.

For the counit, we know that $\mathrm{id} = \varepsilon \circ \alpha = p(\phi^{-1}, \phi) \circ \pi_i \circ \alpha = p(\phi^{-1}, \phi) \circ \alpha_i$, giving the first axiom. Diagrammatically, we have the following.

$$\begin{tikzcd}
p(a,b) \arrow[r, "\alpha"] \arrow[dr, "\mathrm{id}"'] \arrow[rr, bend left, "\alpha_i"] & \int_m p(\underline{m}a, \underline{m}b) \arrow[r, "\pi_i"] \arrow[d, "\varepsilon"] & p(\underline{i}a, \underline{i}b) \arrow[dl, "p(\phi^{-1},\phi)"] \\
 & p(a,b) &
\end{tikzcd}$$

For the comultiplication, we note that the following two diagrams are giving the same morphism if and only if the Tambara condition holds. Because of the universal property of the end, this is the same as to say that $\delta \circ \alpha = \Theta\alpha \circ \alpha$.

$$\begin{tikzcd}
p(a,b) \arrow[r, "\alpha"] \arrow[dr, "\alpha_{u\otimes v}"'] & \int_m p(\underline{m}a, \underline{m}b) \arrow[r, "\delta"] \arrow[d, "\pi"] & \int_m \int_n p(\underline{mn}a, \underline{mn}b) \arrow[d, "\pi"] \\
 & p(\underline{u \otimes v}a, \underline{u \otimes v}b) \arrow[r, "p(\phi^{-1},\phi)"] & p(\underline{uv}a, \underline{uv}b)
\end{tikzcd}$$

$$\begin{tikzcd}
p(a,b) \arrow[r, "\alpha"] \arrow[dr, "\alpha_v"'] & \int_m p(\underline{m}a, \underline{m}b) \arrow[r, "\Theta\alpha"] \arrow[d, "\pi_v"] & \int_n \int_m p(\underline{mn}a, \underline{mn}b) \arrow[d, "\pi_v"] \\
 & p(\underline{v}a, \underline{v}b) \arrow[r, "\alpha"] \arrow[dr, "\alpha_u"'] & \int_m p(\underline{mv}a, \underline{mv}b) \arrow[d, "\pi_u"] \\
 & & p(\underline{uv}a, \underline{uv}b)
\end{tikzcd}$$

Finally, we will show that morphisms of Tambara modules and coalgebra morphisms are the same thing. This gives a bijective-on-objects and fully faithful functor between た and the Eilenberg-Moore category of Θ. Given a natural transformation $\eta\colon p \Rightarrow q$ between Tambara modules endowed with α and α', the exterior part of this diagram commutes when η is a morphism of Tambara modules and the interior part commutes when η is a coalgebra morphism. By the universal property of the end and the definition of Θ, they both imply each other.

$$\begin{tikzcd}
p(a,b) \arrow[r, "\alpha_m"] \arrow[d, "\eta_{a,b}"'] \arrow[rr, bend left, "\alpha"] & \int_m p(\underline{m}a, \underline{m}b) \arrow[r, "\pi_m"] \arrow[d, "\Theta\eta_{a,b}"] & p(\underline{m}a, \underline{m}b) \arrow[d, "\eta_{\underline{m}a,\underline{m}b}"] \\
q(a,b) \arrow[r, "\alpha'"] \arrow[rr, bend right, "\alpha'_m"'] & \int_m q(\underline{m}a, \underline{m}b) \arrow[r, "\pi_m"] & q(\underline{m}a, \underline{m}b)
\end{tikzcd}$$

Proposition 5.2.3 ([PS08])**.** *The Θ comonad has a left adjoint, which must therefore be a monad. On objects, it is given by the following formula.*

$$\Psi q(x,y) = \int^{m\in\mathbf{M}} \int^{a\in\mathbf{C},b\in\mathbf{D}} q(a,b) \times \mathbf{C}(\underline{m}a, x) \times \mathbf{D}(y, \underline{m}b).$$

That is, there exist a natural isomorphism $\mathrm{Nat}(\Phi q, p) \cong \mathrm{Nat}(q, \Theta p)$.

Proof. This comonad can also be written as $\Theta(p) = \int_{m\in\mathbf{M}} p \circ (\underline{m}, \underline{m})$. From this definition we can see that it has a left adjoint $\Psi(p) = \int^{m\in\mathbf{M}} \mathsf{Lan}_{(\underline{m},\underline{m})} p$, which must be a monad. More explicitly, the adjunction can be computed as follows for any given $p, q \colon \mathbf{C}^{op} \times \mathbf{D} \to \mathbf{Sets}$.

$$\begin{aligned}
& q \Rightarrow \Theta p \\
\cong \quad & \text{(Natural transformation as an end)} \\
& \int_{(a,b)\in\mathbf{C}^{op}\times\mathbf{D}} q(a,b) \to \Theta p(a,b) \\
\cong \quad & \text{(Fubini rule)} \\
& \int_{a\in\mathbf{C},b\in\mathbf{D}} q(a,b) \to \Theta p(a,b) \\
\cong \quad & \text{(Definition of } \Theta\text{)} \\
& \int_{a\in\mathbf{C},b\in\mathbf{D}} q(a,b) \to \int_{m\in\mathbf{M}} p(\underline{m}a, \underline{m}b) \\
\cong \quad & \text{(Continuity of the end)} \\
& \int_{a\in\mathbf{C},b\in\mathbf{D}} \int_{m\in\mathbf{M}} q(a,b) \to p(\underline{m}a, \underline{m}b) \\
\cong \quad & \text{(Fubini rule)} \\
& \int_{m\in\mathbf{M}} \int_{a\in\mathbf{C},b\in\mathbf{D}} q(a,b) \to p(\underline{m}a, \underline{m}b) \\
\cong \quad & \text{(Ninja Yoneda lemma)} \\
& \int_{m\in\mathbf{M}} \int_{a\in\mathbf{C},b\in\mathbf{D}} q(a,b) \to \int_{x\in\mathbf{C},y\in\mathbf{D}} \mathbf{C}(\underline{m}a, x) \times \mathbf{D}(y, \underline{m}b) \to p(x,y) \\
\cong \quad & \text{(Continuity of the end)} \\
& \int_{m\in\mathbf{M}} \int_{a\in\mathbf{C},b\in\mathbf{D}} \int_{x\in\mathbf{C},y\in\mathbf{D}} q(a,b) \to (\mathbf{C}(\underline{m}a, x) \times \mathbf{D}(y, \underline{m}b) \to p(x,y)) \\
\cong \quad & \text{(Currying)}
\end{aligned}$$

$$
\begin{aligned}
& \int_{m\in\mathbf{M}} \int_{a\in\mathbf{C},b\in\mathbf{D}} \int_{x\in\mathbf{C},y\in\mathbf{D}} q(a,b)\times\mathbf{C}(\underline{m}a,x)\times\mathbf{D}(y,\underline{m}b)\to p(x,y) \\
\cong \quad & \text{(Fubini rule)} \\
& \int_{x\in\mathbf{C},y\in\mathbf{D}} \int_{a\in\mathbf{C},b\in\mathbf{D}} \int_{m\in\mathbf{M}} q(a,b)\times\mathbf{C}(\underline{m}a,x)\times\mathbf{D}(y,\underline{m}b)\to p(x,y) \\
\cong \quad & \text{(Cocontinuity of the coend)} \\
& \int_{x\in\mathbf{C},y\in\mathbf{D}} \left(\int^{m\in\mathbf{M}} \int^{a\in\mathbf{C},b\in\mathbf{D}} q(a,b)\times\mathbf{C}(\underline{m}a,x)\times\mathbf{D}(y,\underline{m}b)\right)\to p(x,y) \\
\cong \quad & \text{(Definition of } \Psi) \\
& \int_{x\in\mathbf{C},y\in\mathbf{D}} \Psi q(x,y)\to p(x,y) \\
\cong \quad & \text{(Natural transformation as an end)} \\
& \Psi q\Rightarrow p.
\end{aligned}
$$
□

Remark 5.2.4. Because of Lemma 5.2.3, Tambara modules are algebras for the monad Ψ. In particular, knowing that the category た of Tambara modules is equivalently the category of coalgebras for Θ or the category of algebras for Φ, we can construct free and cofree Tambara modules. Given a Tambara module $q \in$ た and some arbitrary functor $p \in \mathbf{Prof}(\mathbf{C})$, we have the following adjuctions $\Phi \dashv U \dashv \Theta$.

$$
\text{た}(\Phi p, q) \cong \mathbf{Prof}(p, Uq), \qquad \text{た}(q, \Theta p) \cong \mathbf{Prof}(Uq, p).
$$

5.3 The profunctor representation theorem

Theorem 5.3.1 ([Ril18, BG18])**.**

$$
\int_{p\in \text{た}} \mathbf{Sets}(p(a,b),p(s,t)) \cong \mathbf{Optic}((a,b),(s,t)).
$$

Proof. The proof here is different from the ones in [Ril18, BG18], although it follows the same basic idea.

$$
\begin{aligned}
& \int_{p\in \text{た}} \mathbf{Sets}(p(a,b),p(s,t)) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int_{p\in \text{た}} \mathbf{Sets}\left(\mathrm{Nat}(\text{よ}_{(a,b)},p),p(s,t)\right) \\
\cong \quad & \text{(Free-forgetful adjunction for Tambara modules)} \\
& \int_{p\in \text{た}} \mathbf{Sets}\left(\text{た}(\Psi_{\mathbf{M}}\text{よ}_{(a,b)},p),p(s,t)\right) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \Psi\,\text{よ}_{(a,b)}(s,t) \\
\cong \quad & \text{(Definition of } \Psi) \\
& \int^{m\in\mathbf{M}} \int^{x\in\mathbf{C},y\in\mathbf{D}} \mathbf{C}(s,\underline{m}x)\times\mathbf{D}(\underline{m}y,t)\times \text{よ}_{(a,b)}(x,y)
\end{aligned}
$$

$$\cong \quad \text{(Yoneda lemma)}$$

$$\int^{m\in\mathbf{M}} \mathbf{C}(s, \underline{m}a) \times \mathbf{D}(\underline{m}b, t). \qquad \square$$

Remark 5.3.2. In fact, we can stop midway there and say that an optic is an element of $\Psi よ_{(a,b)}(s,t)$, which is, again by Yoneda lemma, the same as a natural transformation $よ_{(s,t)} \Rightarrow \Psi よ_{(a,b)}$. We could have defined our category of optics to be the opposite of the full subcategory of representable functors of the Kleisli category for Ψ.

5.4 Examples of profunctor optics

We can apply the profunctor representation theorem 5.3.1 to each one of our optics and get their profunctor versions.

- **Lenses** are described in Proposition 3.2.1 as optics for the product. Tambara modules for the cartesian product are called *strong profunctors* [Kme18] or *cartesian profunctors* [BG18]. The following is the consequence of the profunctor representation theorem.

$$\mathbf{C}(s,a) \times \mathbf{C}(s \times b, t) \cong \int_{p \in た(\times)} \mathbf{Sets}(p(a,b), p(s,t)).$$

- **Prisms** are described in Proposition 3.2.2 as optics for the coproduct. Tambara modules for the coproduct are called *choice profunctors* [Kme18] or *cocartesian profunctors* [BG18].

$$\mathbf{C}(s, t+a) \times \mathbf{C}(b,t) \cong \int_{p \in た(+)} \mathbf{Sets}(p(a,b), p(s,t)).$$

- **Traversals** are described in Proposition 3.3.1 as optics for the evaluation of power series functors. We can write them as functions parametric over Tambara modules for power series functors

$$\mathbf{C}\left(s, \sum_{n\in\mathbb{N}} a^n \times (b^n \to t)\right) \cong \int_{p \in た(\mathrm{Series})} \mathbf{Sets}(p(a,b), p(s,t)).$$

  The commonly used characterization follows from Proposition 4.2.1, which describes traversals as the optic for traversable functors.

$$\mathbf{Sets}\left(s, \sum\nolimits_n a^n \times (b^n \to t)\right) \cong \int_{p \in た(\mathbf{Trv})} \mathbf{Sets}(p(a,b), p(s,t)).$$

A relevant question here is what happens when we compose two optics of two different kinds: for instance a lens with a prism. If we follow what Haskell implementations do, we compose pointwise on profunctors that have Tambara structures for *both* monoidal actions. Lenses and prisms would compose into something of the following form.

$$\int_{(p,p)\in た(\times)\times た(+)} \mathbf{Sets}(p(a,b), p(s,t)).$$

In our example, we would be taking an end over the full subcategory of た$(\times) \times$ た$(+)$ given by structures with the same underlying functor. Categorically, this would be the following pullback of categories.

$$
\begin{tikzcd}
た(\times) \times_{\mathbf{Prof}} た(+) \arrow[r, "\pi"] \arrow[d, "\pi"'] & た(\times) \arrow[d, "U"] \\
た(+) \arrow[r, "U"] & \mathbf{Prof}
\end{tikzcd}
$$

We discuss this composition in §6.3.

5.5 The category of optics as a Kleisli object

We have tried to give a geodesic to the profunctor representation theorem, but the proof outlined in [PS08] for the case of monoidal products can be repeated in full generality. The following lemma is a consequence of Proposition 2.3.3.

Lemma 5.5.1. *There exists an identity on objects functor $\mathbf{C}^{op} \times \mathbf{D} \to \mathbf{Optic}$, which means that the following $\psi \colon (\mathbf{C}^{op} \times \mathbf{D}) \times (\mathbf{C}^{op} \times \mathbf{D})$, given by the set of morphisms of* **Optic***, is a promonad.*

$$\psi((a,b),(s,t)) = \int^{m \in \mathbf{M}} \mathbf{C}(s, \underline{m}a) \times \mathbf{D}(s, \underline{m}b).$$

Moreover, **Optic** *is the Kleisli object for this promonad.*

With this characterization, we can show that copresheaves over **Optic** are precisely modules over the promonad; and use that to show that these are Tambara modules.

Proposition 5.5.2. *Copresheaves over* **Optic** *are equivalent to Tambara modules.*

Proof. We start by showing that $\mathbf{Cat}(\mathbf{Optic}, \mathbf{Sets}) \cong \mathbf{Mod}(\psi)$ By the universal property of the Kleisli object, we know that profunctors from the terminal category $\mathbf{1} \to \mathbf{Optic}$ correspond to right promodules over the promonad, $\mathbf{Prof}(\mathbf{1}, \mathbf{Optic}) \cong \mathbf{Mod}(\psi)$. We note that profunctors of that form are functors $\mathbf{1} \times \mathbf{Optic} \to \mathbf{Sets}$.

Now, because of Lemma 2.3.4, modules over the promonad ψ are equivalent to algebras over Ψ, which are precisely Tambara modules. □

We can use that to reprove the profunctor representation theorem with a technique that is called *double Yoneda* in [Mil17].

Lemma 5.5.3 ("Double Yoneda" from [Mil17])**.** *For any category $\mathbf{A}$, morphisms between x and y are naturally isomorphic to natural transformations between the functors that evaluate copresheaves in x and y.*

$$\mathbf{A}(x,y) \cong [[\mathbf{A}, \mathbf{Sets}], \mathbf{Sets}](-(x), -(y)).$$

Proof. In a category of functors $[\mathbf{A}, \mathbf{Sets}]$, we can apply the Yoneda embedding to two representable functors $\mathbf{A}(y,-)$ and $\mathbf{A}(x,-)$ to get the following.

$$\mathrm{Nat}(\mathbf{A}(y,-), \mathbf{A}(x,-)) \cong \int_{f \in [\mathbf{A}, \mathbf{Sets}]} \mathbf{Sets}\Big(\mathrm{Nat}(\mathbf{A}(x,-), f), \mathrm{Nat}(\mathbf{A}(y,-), f)\Big).$$

Here reducing by Yoneda lemma on both the left hand side and the two arguments of the right hand side, we get the desired result. □

Theorem 5.5.4 (Profunctor representation theorem)**.**

$$\int_{p \in \text{た}} \mathbf{Sets}(p(a,b), p(s,t)) \cong \mathbf{Optic}((a,b),(s,t)).$$

Proof. We apply the double Yoneda lemma (Lemma 5.5.3) to the category **Optic** and then use that copresheaves over it are precisely Tambara modules (Lemma 5.5.2). □

Chapter 6

Composing optics

6.1 Change of action

The motivation of this section is to create functors between categories of optics that are induced by morphisms of actions.

Lemma 6.1.1. *Let $\alpha \colon \mathbf{M} \to [\mathbf{C}, \mathbf{C}]$ and $\beta \colon \mathbf{N} \to [\mathbf{C}, \mathbf{C}]$ be two actions. A morphism of actions $f \colon \mathbf{M} \to \mathbf{N}$ induces a comonad morphism $f_* \colon \Theta_{\mathbf{N}} \to \Theta_{\mathbf{M}}$. Moreover, this assignment is contravariantly functorial, in the sense that for any $f \colon \mathbf{M} \to \mathbf{N}$ and $g \colon \mathbf{N} \to \mathbf{L}$, we have $(g \circ f)_* = f_* \circ g_*$ and $\mathrm{id}_* = \mathrm{id}$.*

Proof. Recall that a comonad morphism is a natural transformation that commutes with counits and comultiplications. We start by defining the natural transformation, using the universal property of the end, as the only morphism making the following diagram commute for any profunctor p and any morphism $h \colon m \to m'$ in $\mathbf{M}$. Here the back of the diagram commutes because of dinaturality of the first end and naturality of the structure morphisms. Here ϕ_f is the natural isomorphism $\underline{fm}a \cong \underline{m}a$ giving the morphism of actions.

$$\begin{tikzcd}
 & \int_{n\in\mathbf{N}} p(\underline{n}a, \underline{n}b) \arrow[dl, "\pi_{fm}"'] \arrow[d, dashed, "f_*"] \arrow[dr, "\pi_{fm'}"] & \\
p(\underline{fm}a, \underline{fm}b) \arrow[d, "\phi_f"'] \arrow[dr] & \int_{m\in\mathbf{M}} p(\underline{m}a, \underline{m}b) \arrow[dl] \arrow[dr] & p(\underline{fm'}a, \underline{fm'}b) \arrow[dl] \arrow[d, "\phi_f"] \\
p(\underline{m}a, \underline{m}b) \arrow[dr, "p(\mathrm{id},h)"'] & p(\underline{fm}a, \underline{fm'}b) \arrow[d] & p(\underline{m'}a, \underline{m'}b) \arrow[dl, "p(h,\mathrm{id})"] \\
 & p(\underline{m}a, \underline{m'}b) &
\end{tikzcd}$$

We can see here that $(g \circ f)_* = f_* \circ g_*$ follows from the fact that composition of monoidal actions is defined to have $\phi_{g\circ f} = \phi_f \circ \phi_g$.

We show now that this natural transformation f_* is in fact a comonad morphism. Counitality follows from commutativity of the following diagram. Here the upper square commutes by definition of f_*, the triangle commutes because of the conditions of the end, and the pentagon commutes by coherence.

$$\begin{tikzcd}
 & \int_n p(\underline{n}a, \underline{n}b) \arrow[r, "f_*"] \arrow[dl, "\pi_i"'] \arrow[d, "\pi_{fi}"] & \int_m p(\underline{m}a, \underline{m}b) \arrow[d, "\pi_i"] \\
p(\underline{i}a, \underline{i}b) \arrow[r, "\phi"] \arrow[dr, "\phi"] & p(\underline{fi}a, \underline{fi}b) \arrow[r, "\phi"] & p(\underline{i}a, \underline{i}b) \arrow[d, "\phi"] \\
 & p(a, b) \arrow[r, "\mathrm{id}"] & p(a, b)
\end{tikzcd}$$

Comultiplicativity follows from the commutativity of the following diagram. The squares of this diagram commute because of the definitions of δ, Θ and f_* and because of coherence.

$$\begin{tikzcd}
\int_n p(\underline{n}a, \underline{n}b) \arrow[rr, "f_*"] \arrow[dr, "\pi_{f(m\otimes m')}"] \arrow[ddr, bend right, "\pi_{fm\otimes fm'}"] \arrow[ddd, "\delta"'] & & \int_m p(\underline{m}a, \underline{m}b) \arrow[dr, "\pi_{m\otimes m'}"] \arrow[ddd, "\delta"] & \\
 & p(f(m\otimes m')a, f(m\otimes m')b) \arrow[rr] \arrow[d, "\phi"] & & p(\underline{m\otimes m'}a, \underline{m\otimes m'}b) \arrow[ddddd, "\phi"] \\
 & p(\underline{fm\otimes fm'}a, \underline{fm'\otimes fm}b) \arrow[dddd] & & \\
\int_n \int_{n'} p(nn'a, nn'b) \arrow[rr, "\Theta f_*"'] \arrow[d, "\pi_n"] & & \int_m \int_{m'} p(mm'a, mm'b) \arrow[d, "\pi_m"] & \\
\int_{n'} p(fmn'a, fmn'b) \arrow[rr, "f_*"] \arrow[dr, "\pi_{fm'}"] & & \int_{m'} p(mm'a, mm'b) \arrow[dr, "\pi_{m'}"] & \\
 & p(fmfm'a, fmfm'b \arrow[rr, "\phi"] & & p(mm'a, mm'b)
\end{tikzcd}$$

□

A comonad morphism induces a functor between the Eilenberg-Moore categories of the comonad, see Definition 2.2.11. In this case, the coalgebras of the Pastro-Street comonad are Tambara modules and we get a functor $\text{た}_{\mathbf{N}} \to \text{た}_{\mathbf{M}}$. These are also the Eilenberg-Moore categories of the adjoint monads; and the functor between Eilenberg-Moore categories induces in turn (see Proposition 2.2.12) a morphism of monads $\Psi_{\mathbf{M}} \to \Psi_{\mathbf{N}}$.

Proposition 6.1.2. *A morphism of actions $f\colon \mathbf{M} \to \mathbf{N}$ induces a functor between the categories of optics $\mathbf{Optic}_{\mathbf{M}} \to \mathbf{Optic}_{\mathbf{N}}$.*

Proof. We have seen that the morphism of actions induces a morphism of monads. The morphism of monads gives an identity-on-objects functor between the Kleisli categories of the monads. The category of optics is the full subcategory on representable functors of that Kleisli category; and because the functor is the identity on objects, it is sent again to a subcategory of representable functors. □

6.2 Distributive actions

6.2.1 Distributive laws for Pastro-Street comonads

If we have a pair of comonads $\Theta_{\mathbf{M}}$ and $\Theta_{\mathbf{N}}$ with left adjoint monads $\Phi_{\mathbf{M}} \dashv \Theta_{\mathbf{M}}$ and $\Phi_{\mathbf{N}} \dashv \Theta_{\mathbf{N}}$, then a distributive law $\Theta_{\mathbf{M}} \circ \Theta_{\mathbf{N}} \Rightarrow \Theta_{\mathbf{N}} \circ \Theta_{\mathbf{M}}$ makes $\Theta_{\mathbf{M}} \circ \Theta_{\mathbf{N}}$ a comonad with a left adjoint given by the composition of the two adjunctions, $\Phi_{\mathbf{N}} \circ \Phi_{\mathbf{M}} \dashv \Theta_{\mathbf{M}} \circ \Theta_{\mathbf{N}}$. This makes $\Phi_{\mathbf{N}} \circ \Phi_{\mathbf{M}}$ a monad. This is particularly useful because we can work in the Kleisli category of this monad and it will contain in particular the Kleisli categories for each one of the monads. That is, we are creating a kind of optic that contains the two kinds we started with.

With this in mind, let us study what is a distributive law between two Pastro-Street comonads as in Definition 5.2.1. Given pairs of monoidal actions $\mathbf{M} \to [\mathbf{C}, \mathbf{C}]$, $\mathbf{M} \to [\mathbf{D}, \mathbf{D}]$, and $\mathbf{N} \to [\mathbf{C}, \mathbf{C}]$, $\mathbf{N} \to [\mathbf{D}, \mathbf{D}]$, we are looking for a natural transformation $\Theta_{\mathbf{M}} \circ \Theta_{\mathbf{N}} \Rightarrow \Theta_{\mathbf{N}} \circ \Theta_{\mathbf{M}}$. Let us apply the Yoneda lemma to reduce this; note that the category of presheaves is cocomplete and coends exist in it.

$$
\begin{aligned}
& \Theta_{\mathbf{M}}\Theta_{\mathbf{N}} \Rightarrow \Theta_{\mathbf{N}}\Theta_{\mathbf{M}} \\
\cong \quad & \text{(Natural transformation as an end)} \\
& \int_{c \in \mathbf{C}} \int_{d \in \mathbf{D}} \int_{p \in \mathbf{Prof}(\mathbf{C},\mathbf{D})} \Theta_{\mathbf{M}}\Theta_{\mathbf{N}} p(c,d) \to \Theta_{\mathbf{N}}\Theta_{\mathbf{M}} p(c,d) \\
\cong \quad & \text{(Fubini, Definition of } \Theta\text{)} \\
& \int_{c,d,p} \left(\int_{m,n} p(\underline{nm}c, \underline{nm}d) \right) \to \left(\int_{n',m'} p(\underline{m'n'}c, \underline{m'n'}d) \right) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int_{c,d,p} \left(\int_{m,n} \mathrm{Nat}\left(\text{よ}_{\underline{nm}c,\underline{nm}d}, p \right) \right) \to \left(\int_{n',m'} p(\underline{m'n'}c, \underline{m'n'}d) \right) \\
\cong \quad & \text{(Continuity)} \\
& \int_{c,d,p} \mathrm{Nat} \left(\int^{m,n} \text{よ}_{\underline{nm}c,\underline{nm}d}, p \right) \to \left(\int_{n',m'} p(\underline{m'n'}c, \underline{m'n'}d) \right) \\
\cong \quad & \text{(Yoneda lemma, Fubini)} \\
& \int_{c,d,n',m'} \int^{m,n} \text{よ}_{\underline{nm}c,\underline{nm}d}(\underline{m'n'}c, \underline{m'n'}d) \\
\cong \quad & \text{(Definition of the representable functor in } \mathbf{C}^{op} \times \mathbf{D}\text{)} \\
& \int_{c,d,n',m'} \int^{m,n} \mathbf{C}(\underline{m'n'}c, \underline{nm}c) \times \mathbf{D}(\underline{nm}d, \underline{m'n'}d).
\end{aligned}
$$

The resulting reduction is slightly more general, but we will be interested in the case where, for all m', n', we give a natural isomorphism $\underline{nm} \cong \underline{m'n'}$ for some m, n. This induces the required $\mathbf{C}(\underline{mn}c, \underline{m'n'}c) \times \mathbf{D}(\underline{m'n'}d, \underline{mn}d)$ that define a distributive law. In that case, we will prove that the distributive law allows us to create a monad that defines a new optic.

Theorem 6.2.1. *Let $\mathbf{M} \to [\mathbf{C}, \mathbf{C}]$, $\mathbf{M} \to [\mathbf{D}, \mathbf{D}]$ and $\mathbf{N} \to [\mathbf{C}, \mathbf{C}]$, $\mathbf{N} \to [\mathbf{D}, \mathbf{D}]$ be two pairs monoidal actions. If for all $m' \in \mathbf{M}$ and $n' \in \mathbf{N}$ we have a natural isomorphism $\underline{nm} \cong \underline{m'n'}$ for some $m \in \mathbf{M}$ and $n \in \mathbf{N}$, chosen dinaturally, then $\Phi_{\mathbf{N}} \circ \Phi_{\mathbf{M}}$ is again a* *monad. Moreover,* $\Phi_{\mathbf{N}} \circ \Phi_{\mathbf{M}} = \Phi_{\mathbf{H}}$ *for a pair of actions* $\mathbf{H} \to [\mathbf{C}, \mathbf{C}]$ *and* $\mathbf{H} \to [\mathbf{D}, \mathbf{D}]$ *that we will construct.*

Proof. We have already shown that the first part of the theorem is a valid way of constructing the distributive law that makes $\Phi_{\mathbf{N}} \circ \Phi_{\mathbf{M}}$ a monad. It is still left to show that this new monad is the Pastro-Street monad for some action.

We will construct a pair of actions $\mathbf{H} \to [\mathbf{C}, \mathbf{C}]$ and $\mathbf{H} \to [\mathbf{D}, \mathbf{D}]$ such that $\Theta_{\mathbf{M}} \circ \Theta_{\mathbf{N}}$ is precisely $\Theta_{\mathbf{H}}$. We take $\mathbf{H} = \mathbf{N} \times \mathbf{M}$ to be a product category, but we endow it with a monoidal product given by $(n_2, m_2) \otimes (n_1, m_1) = (n_2 \otimes n_0, m_0 \otimes m_1)$, where $\underline{n_0 m_0} \cong \underline{m_2 n_1}$. The action is given by $(n, m) \mapsto \underline{n} \circ \underline{m}$. Because of Fubini, we have that $\Theta_{\mathbf{H}} \cong \Theta_{\mathbf{M}} \circ \Theta_{\mathbf{N}}$ as functors.

$$\Theta_{\mathbf{H}} p(c, d) = \int_{(n,m) \in \mathbf{H}} p(\underline{nm}c, \underline{nm}d) \cong \int_{m \in \mathbf{M}} \int_{n \in \mathbf{N}} p(\underline{nm}c, \underline{nm}d).$$

Note that the comultiplication of the composition is given by the distributive law and thus we can check that it is the same one $\mathbf{H}$ uses. □

This is particularly interesting because we can compose pairs of optics of different kinds whose monads have a distributive law $d \colon \Phi_{\mathbf{N}} \circ \Phi_{\mathbf{M}} \Rightarrow \Phi_{\mathbf{M}} \circ \Phi_{\mathbf{N}}$. If we have a pair of optics of different kinds written as Kleisli morphisms, $f \colon よ_{s,t} \Rightarrow \Phi_{\mathbf{M}} よ_{a,b}$ and $g \colon よ_{a,b} \Rightarrow \Phi_{\mathbf{N}} よ_{x,y}$, regard both as Kleisli morphisms for the composite comonad $\Phi_{\mathbf{N}} \circ \Phi_{\mathbf{M}} = \Phi_{\mathbf{H}}$ as follows.

$$l^* \colon よ_{s,t} \xrightarrow{f} \Phi_{\mathbf{M}} よ_{a,b} \xrightarrow{\eta} \Phi_{\mathbf{N}} \Phi_{\mathbf{M}} よ_{x,y},$$

$$g^* \colon よ_{a,b} \xrightarrow{\eta} \Phi_{\mathbf{M}} よ_{a,b} \xrightarrow{g} \Phi_{\mathbf{N}} \Phi_{\mathbf{M}} よ_{x,y} \xrightarrow{d} \Phi_{\mathbf{M}} \Phi_{\mathbf{N}} よ_{x,y}.$$

We will now discuss concrete examples of this construction. In §6.2.2 we construct two optics from the fact that products distribute over sums and exponentials distribute over products.

6.2.2 Affine traversals and glasses

Let $\mathbf{C}$ be a bicartesian closed category, which implies that products distribute over coproducts. For any $a, b \in \mathbf{C}$, we have that $a \times (b + (-)) \cong (a \times b) + a \times (-)$. The optic given by this action is called an *affine traversal*, has a concrete representation and was first conjectured to exist in [PGW17] and considered in [BG18]. We can use Theorem 6.2.1 on the actions of lenses and prisms to see that $c + d \times (-)$ is an action describing this new optic that contains both lenses and prisms.

Proposition 6.2.2. ***Affine traversals*** *are optics for the action* $\mathbf{C}^2 \to [\mathbf{C}, \mathbf{C}]$ *that sends* $c, d \in \mathbf{C}$ *to the functor* $F(a) = c + d \times a$*. They have a concrete description*

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

Proof.

$$\int^{c,d} \mathbf{C}(s, c + d \times a) \times \mathbf{C}((c + d \times b), t)$$

$$\cong \qquad \text{(Coproduct)}$$

$$
\begin{aligned}
& \int^{c,d} \mathbf{C}(s, c + d \times a) \times \mathbf{C}(c, t) \times \mathbf{C}(d \times b, t) \\
\cong \quad & \text{(Exponential)} \\
& \int^{d} \mathbf{C}(s, t + d \times a) \times \mathbf{C}(d, (b \to t)) \\
\cong \quad & \text{(Yoneda)} \\
& \mathbf{C}(s, t + a \times (b \to t)).
\end{aligned}
$$

□

The affine traversal motivates us to look for other distributive laws. Using the theory previously developed, we will use that exponentials distribute over products to create a new concrete optic. Let $\mathbf{C}$ be a cartesian closed category. For any $a, b \in \mathbf{C}$, we can distribute exponentials over products as $a \to (b \times (-)) \cong (a \to b) \times (a \to (-))$. We will call *glass* to the kind of optic generated by this action; lenses and grates are, in particular, glasses. To the best of our knowledge, neither a description nor a derivation of this optic were present in the literature; it remains to be seen what are its potential applications.

Proposition 6.2.3. ***Glasses*** *are optics for the action* $\mathbf{C}^2 \to [\mathbf{C}, \mathbf{C}]$ *that sends* $c, d \in \mathbf{C}$ *to the functor* $F(a) = d \times (c \to a)$. *They have a concrete description*

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

Proof.

$$
\begin{aligned}
& \int^{c,d} \mathbf{C}(s, c \times (d \to a)) \times \mathbf{C}((d \to b) \times c, t) \\
\cong \quad & \text{(Product)} \\
& \int^{c,d} \mathbf{C}(s, c) \times \mathbf{C}(s, d \to a) \times \mathbf{C}((d \to b) \times c, t) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \int^{d} \mathbf{C}(s, d \to a) \times \mathbf{C}((d \to b) \times s, t) \\
\cong \quad & \text{(Product-Exponential)} \\
& \int^{d} \mathbf{C}(d, s \to a) \times \mathbf{C}((d \to b) \times s, t) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \mathbf{C}(((s \to a) \to b) \times s, t).
\end{aligned}
$$

□

6.3 Joining Tambara modules

In Haskell, the composition of two optics of two different kinds is done as follows. Assume we have monoidal actions $\alpha \colon \mathbf{M} \to [\mathbf{C}, \mathbf{C}]$ and $\beta \colon \mathbf{N} \to [\mathbf{C}, \mathbf{C}]$, determining two different kinds of optics. By virtue of Theorem 5.3.1, the optics can be written as functions that will be polymorphic over Tambara modules for $\mathbf{M}$ and $\mathbf{N}$; that is, over coalgebras for $\Theta_{\mathbf{M}}$ and $\Theta_{\mathbf{N}}$.

$$
\int_{p \in \text{た}(\alpha)} \mathbf{Sets}(p(a,b), p(s,t)), \qquad \int_{q \in \text{た}(\beta)} \mathbf{Sets}(q(x,y), q(a,b)).
$$

When we compose them, Haskell outputs a function that is polymorphic over profunctors with Tambara module structure for both actions. These are profunctors with structure of *bicoalgebra*: coalgebra structure for both $\Theta_{\mathbf{M}}$ and $\Theta_{\mathbf{N}}$.

$$\int_{(p,p)\in \text{た}(\alpha)\times_{\mathbf{Prof}} \text{た}(\beta)} \mathbf{Sets}(p(a,b), p(s,t))$$

If we want to see this new function as an optic, we need to describe these bicoalgebras as coalgebras for another Pastro-Street comonad. The action that will generate this new comonad will be the coproduct of the two actions we had, $\alpha+\beta\colon \mathbf{M}+\mathbf{N}\to[\mathbf{C},\mathbf{C}]$.

$$\begin{tikzcd}
\mathbf{M} \arrow[r, "i"] \arrow[dr, bend right, "\alpha"'] & \mathbf{M}+\mathbf{N} \arrow[d, dashed, "\exists!\alpha+\beta"] & \mathbf{N} \arrow[l, "i"'] \arrow[dl, bend left, "\beta"] \\
 & {[\mathbf{C},\mathbf{C}]} &
\end{tikzcd}$$

That is, we are claiming that, for a fixed profunctor p, a pair of coalgebra structures over $\Theta_{\mathbf{M}}$ and $\Theta_{\mathbf{N}}$ is exactly the same as a coalgebra structure over $\Theta_{\mathbf{M}+\mathbf{N}}$. If we call $\text{た}_{\mathbf{M}}p$ to the set of Tambara structures over p for the monoidal action of $\mathbf{M}$, we are claiming that $\text{た}_{\mathbf{M}+\mathbf{N}}p \cong \text{た}_{\mathbf{M}}p \times \text{た}_{\mathbf{N}}p$. In order to show this, we will prove that $\text{た}_{(-)}p\colon \mathbf{MonCat}/[\mathbf{C},\mathbf{C}]\to\mathbf{Sets}$ is representable.

Theorem 6.3.1. *Let $p\in\mathbf{Prof}(\mathbf{C},\mathbf{D})$ be a profunctor. The associated functor*

$$\text{た}_{(-)}p\colon \mathbf{MonCat}/[\mathbf{C},\mathbf{C}]\to\mathbf{Sets}$$

is representable.

Proof. We will show that $\text{た}_{\mathbf{M}}p\cong\mathbf{MonCat}/[\mathbf{C},\mathbf{C}](\mathbf{M},\mathbf{D}_p)$ for a monoidal action $\gamma\colon\mathbf{D}_p\to[\mathbf{C},\mathbf{C}]$ from a category $\mathbf{D}_p$ we are going to construct. We define the objects of $\mathbf{D}_p$ to be pairs (f,η) with $f\in[\mathbf{C},\mathbf{C}]$ and $\eta\in\int_{a,b}p(a,b)\to p(fa,fb)$. Morphisms of $\mathbf{D}_p$ between (f,η) and (g,η') are natural transformations $\alpha\colon f\Rightarrow g$ such that the following triangle commutes.

$$\begin{tikzcd}
 & p(fa,fb) \arrow[d, "p(\alpha,\alpha)"] \\
p(a,b) \arrow[ur, "\eta"] \arrow[r, "\eta'"'] & p(ga,gb)
\end{tikzcd}$$

Finally, the monoidal product of $\mathbf{D}_p$ is given by $(f,\eta)\otimes(g,\eta')=(f\circ g,\eta\circ\eta')$. The action $\mathbf{D}_p\to[\mathbf{C},\mathbf{C}]$ simply projects the first component.

A Tambara module over p is precisely choosing some $p(a,b)\to p(fa,fb)$ for every $m\in\mathbf{M}$ such that $\underline{m}=f$. This needs to be done in a natural way, which is precisely the dinaturality of the Tambara module, and preserving the monoidal structure, which precisely gives the Tambara axioms. $\square$

The fact that it is representable implies $\text{た}_{\mathbf{M}+\mathbf{N}}p\cong\text{た}_{\mathbf{M}}p\times\text{た}_{\mathbf{N}}p$. The conclusion is that, when Haskell joins the constraints of two profunctor optics, it is creating a profunctor optic for the coproduct action. By analogy with the coproduct monoid, the coproduct action from $\mathbf{M}+\mathbf{N}$ can be thought as a string of objects of $\mathbf{M}$ and $\mathbf{N}$ acting consecutively. For instance, if we consider the actions given by the cartesian product and the disjoint union of sets, the coproduct of these actions is an action taking a word of sets $x_1,y_1,\dots,x_n,y_n$ and acting on a set a as in the following formula.

$$x_1+y_1\times(x_2+y_2\times(\dots(x_n+y_n\times a)\dots))$$

Remark 6.3.2. Algebras for the coproduct monad are precisely pairs of algebra structures for each one of the components [Kel80] [AMBL12]. Coalgebras for the product comonad are precisely pairs of coalgebra structures for each one of the components. This suggests that the previous result is just trying to say $\Theta_{\mathbf{M}+\mathbf{N}} \cong \Theta_{\mathbf{M}} \times \Theta_{\mathbf{N}}$, where the product must be understood as the product in the category of comonads.

6.4 Clear optics and the lattice of optics

The two previous sections pose a problem: Haskell seems to be joining lenses and prisms into an optic for a complicated action of the form $x_1 + y_1 \times (\dots(x_n + y_n \times a)\dots)$, but we actually only care about the action that determined affine traversals, $x + y \times a$, which is much simpler. The relation between these two actions is that any action with the form of the first one is always isomorphic in the category $[\mathbf{C}, \mathbf{C}]$ to some action of the second form. It is folklore that lenses and prisms *compose* into affine traversals but, to make that precise in our description of optics, we are implicitly quotienting by some isomorphisms. We would like to know which quotienting is being done and have a formal description of this construction.

A problem that looks vaguely related is that we can artificially include irrelevant information in our optics. Consider for instance an adapter, described by the action of the functor from the terminal category $\mathbf{1} \to [\mathbf{C}, \mathbf{C}]$ picking the identity functor. Consider now the action given by the monoid (the discrete monoidal category) of natural numbers $\mathbb{N} \to [\mathbf{C}, \mathbf{C}]$ sending $1 \in \mathbb{N}$ to the functor $(1 \times (-)) \in [\mathbf{C}, \mathbf{C}]$. Both are essentially describing adapters, but the second one will *"remember"* the number that was used to construct it. An element of the first optic is just an adapter, while an element of the second optic is an adapter together with a natural number. The problem is that, inside $\mathbb{N}$, the functors $1 \times (1 \times (-))$ and $1 \times (-)$ are regarded as non-isomorphic; that is,

$$\int^{1} \mathbf{C}(s, a) \times \mathbf{C}(b, t) \ncong \int^{n \in \mathbb{N}} \mathbf{C}(s, a) \times \mathbf{C}(b, t).$$

One would think that these problems would dissapear when, instead of considering arbitrary monoidal actions as in [Ril18], we limit the possible actions to be given by subcategories of endofunctors as in [BG18] and moreover we require them to be full. However, that comes with its own set of problems: the category where we are applying Yoneda to get the concrete representations is *not* always a full subcategory of endofunctors. We propose an intermediate way between these two definitions. We will consider just the optics defined by *pseudomonic* actions (as in Definition 2.4.1): actions that determine subcategories that are full on isomorphisms. The second problem is solved because the artificial description of adapters is not pseudomonic; the first problem can be solved quotienting to the smallest pseudomonic action, which is precisely the one that describes affine traversals. Let us detail this approach to what we will call **clear optics**.

6.4.1 Clear optics

Definition 6.4.1. An kind of optic is **clear** if it is given by a monoidal action $\mathbf{M} \to [\mathbf{C}, \mathbf{C}]$ that is a pseudomonic functor as in Definition 2.4.1.

In other words, if we consider only replete subcategories, clear optics are these given by the subobjects of $[\mathbf{C}, \mathbf{C}]$ in the category **MonCat**. Let us show that some common optics are clear at least in the case of $\mathbf{C} = \mathbf{Sets}$.

Lemma 6.4.2. *Lenses are a clear optic.*

Proof. We want to show that the action of the product is pseudomonic. For some pair of sets $C, D \in \mathbf{Sets}$, assume an isomorphism $u\colon (C \times A) \cong (D \times A)$ natural in A. We need to show that it is the image of some isomorphism $C \cong D$ under the action. We take $v\colon C \cong C \times 1 \cong D \times 1 \cong D$. The following diagram commutes by naturality.

$$\begin{tikzcd}
C \times A \arrow[d, "u_A"] & C \times 1 \arrow[l, "\mathrm{id}\times a"'] \arrow[d, "u_1"] & C \arrow[l, "\cong"'] \arrow[d, "v"] \\
D \times A & D \times 1 \arrow[l, "\mathrm{id}\times a"] & D \arrow[l, "\cong"]
\end{tikzcd}$$

For every $a\colon 1 \to A$, we have that $u_A(c, a) = (v(c), a)$. That shows that $u = (v \times -)$, and thus the action of the product is pseudomonic. □

Lemma 6.4.3. *Prisms are a clear optic.*

Proof. We want to show that the action of the coproduct is pseudomonic. For some pair of sets $C, D \in \mathbf{Sets}$, assume an isomorphism $u\colon (C + A) \cong (D + A)$ natural in A. We need to show that it is the image of some isomorphism $C \cong D$ under the action. We take $v\colon C \cong C + 0 \cong D + 0 \cong D$ and because of naturality, the isomorphism $C + A \cong D + A$ must behave like v on elements of C.

$$\begin{tikzcd}
C + A \arrow[d, "u_A"] & C + 0 \arrow[l, "\mathrm{id}\times a"'] \arrow[d, "u_1"] & C \arrow[l, "\cong"'] \arrow[d, "v"] \\
D + A & D + 0 \arrow[l, "\mathrm{id}\times a"] & D \arrow[l, "\cong"]
\end{tikzcd}$$

An isomorphism between sets $C + A \cong D + A$ that behaves like an isomorphism $C \cong D$ in elements of C must split into this isomorphism and an isomorphism $A \cong A$. Being natural in A, this needs to be the identity because of Yoneda lemma. □

Lemma 6.4.4. *Affine traversals are a clear optic.*

Proof. For some sets $C, D, C', D' \in \mathbf{Sets}$, consider an isomorphism $D + C \times A \cong D' + C' \times A$. Taking $A = 0$ and following the same reasoning as in Lemma 6.4.1, ew get an isomorphism $C \times A \cong C' \times A$ natural in A. Following 6.4.1, we get an isomorphism $C \cong C'$ that together with $D \cong D'$ shows the original isomorphism as image under the action of these. □

Remark 6.4.5. Not every optic we have considered so far is clear. Glasses, for instance, are not a clear optic because $0 \times (1 \to a) \cong 0 \times (2 \to a)$ while $1 \ncong 2$. We could, however, take the replete image of the action determining glasses and define a clear version of the same optic. Given an action $\alpha\colon \mathbf{M} \to [\mathbf{C}, \mathbf{C}]$ determining a category of optics $\mathbf{Optic}_{\mathbf{M}}$, we can consider the replete image (as in §2.4) of $\mathbf{M}$ and take the action $\mathrm{repl}(\mathrm{img}(\alpha))\colon \mathrm{repl}(\mathrm{img}(\mathbf{M})) \to [\mathbf{C}, \mathbf{C}]$. This is a clear optic because of Proposition 2.4.3. The inclusion $\mathbf{M} \to \mathrm{repl}(\mathrm{img}(\mathbf{M}))$ is determining a morphism of actions that, by virtue of Lemma 6.1.1, induces a functor $\mathbf{Optic}_{\mathbf{M}} \to \mathbf{Optic}_{\mathrm{repl}(\mathrm{img}(\mathbf{M}))}$. This functor is full, and this can be checked realizing that the hom-sets of the second category are just coends with extra conditions and that the map with the following signature is a surjection.

$$\left(\int^{m \in \mathbf{M}} \mathbf{C}(s, \underline{m}a) \times \mathbf{C}(\underline{m}b, t) \right) \to \left(\int^{m \in \mathrm{repl}(\mathrm{img}(\mathbf{M}))} \mathbf{C}(s, \underline{m}a) \times \mathbf{C}(\underline{m}b, t) \right)$$

In other words, there exists a clear version of every optic. Even when we have derived concrete forms only for the original version, the same concrete forms work for the clear version, up to some identifications.

6.4.2 Composing clear optics

We want to reconcile here how Haskell joins Tambara modules to compose profunctor optics, as in §6.3, with the intuition that lenses and prisms should compose into affine traversals, as in 6.2.2.

We have shown that, after joining the Tambara structures of a profunctor lens with a profunctor prism, we should get a profunctor optic for the coproduct action. The coproduct action takes a word of sets $x_1, y_1, \dots, x_n, y_n$ and acts on a set a as in the following formula.

$$x_1 + y_1 \times (x_2 + y_2 \times (\dots (x_n + y_n \times a) \dots))$$

The category giving this action is *not* equivalent to the one given by affine traversals $x + y \times (-)$. The morphisms on the first category are componentwise morphisms of the words, whereas a morphism $x_1, y_1, \dots, x_n, y_n \to x'_1, y'_1, \dots, x'_n, y'_n$ is given by a family of morphisms $x_i \to x'_i$ and $y_i \to y'_i$. This means that, contrary to intuition, $1+1\times(1+\times(-)) \ncong 2+(-)$ in this category. The problem is solved if we consider the corresponding clear optic: the repletion of this category is the same as the repletion of the category giving affine traversals. Moreover, because affine traversals are a clear optic, that means that they are equivalent as categories. To sum up, when we join Tambara modules, lenses and prisms compose into some complicated optic; however, the clear version of this optic is equivalent (in the categorical sense) to the one describing affine traversals. If we are willing to accept that we are working with clear optics, we can say that Haskell composes lenses and prisms into affine traversals in a sound way that is compatible with the Tambara structure.

6.4.3 Lattice of optics

If clear optics are given by replete subcategories of $[\mathbf{C}, \mathbf{C}]$, we can consider the lattice they form and translate it via Lemma 6.1.1 to a lattice of optics. We aim to recover and expand the lattice of optics that was first described in [PGW17].

We start by constructing the lattice of replete subcategories. We will be working in the full subcategory of $\mathbf{MonCat}/[\mathbf{C}, \mathbf{C}]$ where objects are pseudomonic. Given two objects $\alpha \colon \mathbf{M} \to [\mathbf{C}, \mathbf{C}]$ and $\beta \colon \mathbf{N} \to [\mathbf{C}, \mathbf{C}]$, let $\alpha \wedge \beta$ be their product on this category, which is the following pullback.

$$\begin{tikzcd} & \mathbf{M} \wedge \mathbf{N} \arrow[dl, "\pi"'] \arrow[dr, "\pi"] & \\ \mathbf{M} \arrow[dr, "\alpha"'] & & \mathbf{N} \arrow[dl, "\beta"] \\ & {[\mathbf{C}, \mathbf{C}]} & \end{tikzcd}$$

This is again pseudomonic because pseudomonic functors are stable under pullback. In terms of replete subcategories, we are taking an intersection of the two categories. Let $\alpha \vee \beta$ be the coproduct on this category; it is given by the smallest replete subcategory containing both. Because of the construction of the replete image, this is precisely the replete image of the coproduct, $\mathrm{repl}(\mathrm{img}(\alpha + \beta))$, as in the following diagram.

$$
\begin{tikzcd}
\mathbf{M} \arrow[r, "i"] \arrow[ddr, bend right, "\alpha"'] & \mathbf{M}+\mathbf{N} \arrow[d, "\mathrm{replimg}"] & \mathbf{N} \arrow[l, "i"'] \arrow[ddl, bend left, "\beta"] \\
& \mathbf{M}\vee\mathbf{N} \arrow[d] & \\
& {[\mathbf{C},\mathbf{C}]} &
\end{tikzcd}
$$

We will actually construct a bounded lattice. The top element is given by the identity action $\top \colon [\mathbf{C}, \mathbf{C}] \to [\mathbf{C}, \mathbf{C}]$, and the bottom element is given by the action of the trivial monoidal category $\bot \colon \mathbf{1} \to [\mathbf{C}, \mathbf{C}]$.

Lemma 6.4.6. *The previously defined operations* $(\wedge, \vee, \top, \bot)$ *form a bounded lattice up to isomorphisms.*

Proof. First, we note that $(\wedge, \vee)$ are commutative and associative because they are products and coproducts. The meet $(\wedge)$ is idempotent, $\alpha \wedge \alpha = \alpha$, because pseudomonic functors are precisely those whose pullback with themselves is again themselves, see Definition 2.4.1. The join $(\vee)$ is idempotent, $\alpha \vee \alpha$, because of the construction of the repleted image. As subcategories, the image of $\alpha \wedge \beta$ is necessarily contained in α, and that gives $\alpha \vee (\alpha \wedge \beta) = \alpha$. On the other hand, α is necessarily contained in $\alpha \vee \beta$, and that gives $\alpha \wedge (\alpha \vee \beta) = \alpha$.

Every monoidal action must send the unit of the monoidal category to the identity, this ensures $\alpha \vee \bot = \alpha$. Because they are all subcategories of endofunctors, $\alpha \wedge \top = \alpha$. $\square$

Theorem 6.4.7. *The lattice induces functors between categories of clear optics. We have the following hierarchy of optics.*

Proof. We can consider many evident inclusions between the different actions that define the optics in the lattice defined in Lemma 6.4.6. For instance, affine maps can be written as power series functors, so the action defining affine traversals is contained in the one defining traversals. Because of Lemma 6.1.1, that lattice is transported to functors between categories of optics. Note that the bottom and the top correspond to **Adapter** and **Setter**, respectively.

$$
\begin{tikzcd}
& & \mathbf{Setter} & & \\
& & \mathbf{Traversal} \arrow[u] & & \mathbf{Kaleidoscope} \arrow[ull] \\
& \mathbf{Glass} \arrow[uur] & \mathbf{Affine} \arrow[u] & & \\
\mathbf{Grate} \arrow[ur] & \mathbf{Prism} \arrow[ur] & & \mathbf{Lens} \arrow[ul] \arrow[ull] & \\
& & & \mathbf{ListLens} \arrow[u] \arrow[uuur] & \\
& & \mathbf{Adapter} \arrow[uuull] \arrow[uul] \arrow[ur] & &
\end{tikzcd}
$$

Note that these are not the only optics we have considered. It is interesting to compare how this matches the original lattice of optics in [PGW17] and the one in [Boi17], described for Haskell constraints. □

Chapter 7

Applications

7.1 Library of optics

As an application, we have implemented a Haskell library that makes use of the general form of the profunctor representation theorem (Theorem 5.3.1) to provide a translation between the existential and profunctor representations, parametric in the type of optic. This development follows closely the definitions in [BG18], in that it considers optics as given by submonoids of endofunctors instead of monoidal actions.

In order to achieve this, we consider the submonoid of endofunctors described by the monoidal action as given by some Haskell *constraint*. Constraints are then passed to the relevant data constructors (in GADT style) using the `ConstraintKinds` extension of the Glasgow Haskell Compiler. The complete code for the full description and translation between the two representations of optics is surprisingly concise and it is included here with references to the relevant theorems.

- Existential optic as in Definition 3.1.1.

```
data ExOptic mon a b s t where
  ExOptic :: (mon f) => (s -> f a) -> (f b -> t)
                     -> ExOptic mon a b s t
```

- A class that witnesses that the given constraint declares a submonoid of endofunctors. Note that the following morphisms can be constructed when the constraint is defining a class of endofunctors closed under identity and composition, see the construction of the category of optics in §3.1.

```
class MonoidalAction mon where
  idOptic   :: ExOptic mon a b a b
  multOptic :: (mon f) => ExOptic mon a b s t
                       -> ExOptic mon a b (f s) (f t)
```

- Definition of a Tambara module (Defintion 5.1.1) and the profunctor representation.

```
class (Profunctor p) => Tambara mon p where
  action :: forall f a b . (mon f) => p a b -> p (f a) (f b)

type ProfOptic mon a b s t =
  forall p . Tambara mon p => p a b -> p s t
```

- The equivalence given by the profunctor representation theorem (Theorem 5.3.1) is constructed explicitly here.

```
instance Profunctor (ExOptic mon a b) where
  dimap u v (ExOptic l r) = ExOptic (l . u) (v . r)

instance (MonoidalAction mon) => Tambara mon (ExOptic mon a b) where
  action = multOptic

crOptic :: (MonoidalAction mon) =>
  ProfOptic mon a b s t -> ExOptic mon a b s t
crOptic p = p idOptic

mkOptic :: forall mon a b s t .
  ExOptic mon a b s t -> ProfOptic mon a b s t
mkOptic (ExOptic l r) = dimap l r . (action @mon)
```

Note that the code included here is enough for implementing the translation between representations of optics in general. We believe this can lead to more concise optic libraries in the future. Completing the library, including an implementation of the full range of the combinators that optics libraries such as Kmett's [Kme18] provide, is left for further work.

7.1.1 Example usage

The code, together with implementations for the most common optics and examples of usage, can be found at the following HTML address.

- `https://github.com/mroman42/vitrea/`

We take seriously Examples 1.1.2, 1.1.5 and 1.1.8 to give an idea of how this library works. The exact code for these examples can be found in the library.

Example 7.1.1 (Lenses and prisms). Examples 1.1.2, 1.1.5 show how we can compose a *lens* (`street`) with a *prism* (`postal`) and still be able to access the contents. In the first code line we declare a string called `address`. In the second line, we match it into a postal address using the prism `postal`; this step could have failed. Then we compose a prism with a lens to get the affine traversal `postal.street`, that accesses the street inside a string. We can update that internal field and have the changes propagate upwards. The same prism can be reused with a different lens (`city`) to access a different part of the address. This solution is modular.

```
let address = "45 Banbury Rd, OX1 3QD, Oxford"
address^.postal
  -- Street:  45 Banbury Rd
  -- Code:    OX2 6LH
  -- City:    Oxford
address^.postal.street
  -- "45 Banbury Rd"
address^.postal.street %~ "7 Banbury Rd"
  -- "7 Banbury Rd, OX1 3QD, Oxford"
```

```
address^.postal.city <>~ " (UK)"
  -- "45 Banbury Rd, OX1 3QD, Oxford (UK)"
```

Example 7.1.2 (Traversals). The traversal (`mail`) from Example 1.1.8 can be used how to compose traversals with lenses. In the first line of code, we have a mailing list. In the second line of code, we extract the emails using the traversal. In the third line of code we compose it with two lenses (`email`, `domain`) to access particular subfields. Note that lenses and traversals compose again to traversals (Lemma 6.4.6). In the last line of code, we apply an `uppercase` function the contents of these subfields; the changes propagate to the initial mailing list.

```
someMailingList
  -- | Name           | Email                    | Frequency |
  -- |----------------+--------------------------+-----------|
  -- | Turing, Alan   | turing@manchester.ac.uk | Daily     |
  -- | Noether, Emily | emmynoether@fau.eu       | Monthly   |
  -- | Gauss, Carl F. | gauss@goettingen.de      | Weekly    |
someMailingList^..mails
  -- ["turing@manchester.ac.uk", "emmynoether@fau.eu", "gauss@goettingen.de"]
someMailingList^..mails.email.domain
  -- ["manchester.ac.uk", "fau.eu", "goettingen.de"]
someMailingList^..mails.email.domain <~~~ uppercase
  -- | Name           | Email                    | Frequency |
  -- |----------------+--------------------------+-----------|
  -- | Turing, Alan   | turing@MANCHESTER.AC.UK | Daily     |
  -- | Noether, Emily | emmynoether@FAU.EU       | Monthly   |
  -- | Gauss, Carl F. | gauss@GOETTINGEN.DE      | Weekly    |
```

7.1.2 A case study

The code presented at the beginning of this section makes the profunctor representation theorem works for any arbitrary action. We can go beyond the usual optics and elaborate an example that starts from the observation of Remark 3.4.4: a *list lens* can be composed with a *kaleidoscope* to create a new *kaleidoscope*.

- We start by defining *kaleidoscopes* as the optic for applicative functors. The definition of a new optic is simple under this framework: we only need to witness the fact that applicative functors are closed under composition and that they contain the identity functor.

  ```
  type ExKaleidoscope   s t a b = ExOptic   Applicative a b s t
  type ProfKaleidoscope s t a b = ProfOptic Applicative a b s t

  instance MonoidalAction Applicative where
    idOptic = ExOptic Identity runIdentity
    multOptic (ExOptic l r) = ExOptic
      (Compose . fmap l) (fmap r . getCompose)
  ```

  List lenses are defined in a similar fashion as the optic for the product of a monoid.

Now assume we have a large dataset where, instead of reading all the entries, we want to learn about the aggregate data. In our case we pick the standard *iris* dataset [Fis36]. The entries of the dataset represent a **Flower**, characterized by its **Species** (which can be *Iris setosa*, *Iris versicolor* and *Iris virginica*) and four numeric **Measurements** on the length and width of its sepal and petal. One could take $\mathbf{Species} = \mathbf{3}$, $\mathbf{Measurements} = \mathbb{R}^4_+$ and $\mathbf{Flower} = \mathbf{Species} \times \mathbf{Measurements}$.

We consider a type-invariant *list lens* of the following type.

$$\text{measureNearest} \in \mathbf{ListLens}((\mathbf{Flower}, \mathbf{Flower}), (\mathbf{Measurements}, \mathbf{Measurements})),$$

It is given by a *view* function $\mathbf{Flower} \to \mathbf{Measurements}$, that projects the measurements of a flower; and a *classify* function $\mathbf{Flower}^* \times \mathbf{Measurements} \to \mathbf{Flower}$, that takes a list of flowers and uses it to classify the input measurements into some species, outputting a new flower. Internally, our classify function will use the *1-nearest neighbour algorithm* [CH+67]. We also consider a type-invariant *kaleidoscope* of the following type.

$$\text{aggregate} \in \mathbf{ListLens}((\mathbf{Measurements}, \mathbf{Measurements}), (\mathbb{R}_+, \mathbb{R}_+)),$$

It is given by a function that takes as an input some monoid structure in the positive real numbers, given as a map $\mathbb{R}^*_+ \to \mathbb{R}$ from the free monoid, and induces a componentwise monoid structure on **Measurements**.

- List lenses are, in particular, lenses; we can use them to *view* the measurements of the first element of our dataset.

```
(iris !! 1)^.measureNearest
  --      Sepal length: 4.9
  --      Sepal width:  3.0
  --      Petal length: 1.4
  --      Petal width:  0.2
```

- They are more abstract than a lens in the sense that they can be used to classify some measurements into a new species taking into account all the examples of the dataset.

```
(iris ?. measureNearest) (Measurements 4.8 3.1 1.5 0.1)
  -- Flower:
  --      Sepal length: 4.8
  --      Sepal width:  3.1
  --      Petal length: 1.5
  --      Petal width:  0.1
  --      Species:      Iris setosa
```

- Now we compose the list lens with the kaleidoscope and we get back a new kaleidoscope that, from a monoid structure on the positive real numbers, induces first a monoid structure on the measurements, uses it to aggregate the measurements of the dataset, and finally classifies the aggregated measurements into a species.

```
(iris >- measureNearest.aggregate) mean
  -- Flower:
  --     Sepal length: 5.843
  --     Sepal width:  3.054
  --     Petal length: 3.758
  --     Petal width:  1.198
  --     Species:      Iris versicolor
```

- Any other monoidal structure will work the same. Here we use the maximum instead of the average.

```
(iris >- measureNearest.aggregate) maximum
  -- Flower:
  --      Sepal length: 7.9
  --      Sepal width:  4.4
  --      Petal length: 6.9
  --      Petal width:  2.5
  --      Species:      Iris virginica
```

7.2 Agda implementation

We have also developed an Agda implementation of the library. The Agda language [Nor] is dependently typed, and the enhanced expressivity of the type system justifies some changes from the Haskell version. As a major difference, optics are defined as arising from a monoidal action (as in [Ril18]) instead of a submonoid of endofunctors. The code can be found in the following link.

- https://github.com/mroman42/vitrea-agda

Foundations become relevant when writing a formalization of our reasoning, and there is an important aspect of this text that we have not discussed yet: even when we have been agnostic regarding foundations, the theory of optics is *constructive* in nature (in the sense of [TVD14]). The reader can check that we have avoided the use of the excluded middle or the axiom of choice (which implies excluded middle). There is a clear exception in Proposition 2.4.3, but we are not considering it part of the main theory and we think the use of the axiom of choice could be avoided in this case considering *anafunctors*, which are motivated by foundational concerns exactly like this one (see [Mak96]). This makes it possible to formalize parts of this text in some variant of Martin-Löf type theory using a proof assistant.

- The library is built in turn over a small library we have developed to deal automatically with trivial isomorphisms in the category **Sets**. The automation works on top of Agda's instance resolution algorithm, and given two types depending on some variables, it tries to find a isomorphism between them, natural on the variables.
- There is a partial formalization and a construction of the bijection of the profunctor representation theorem (Theorem 5.3.1). It provides a definition of monoidal action (Definition 2.2.14) and of Tambara module over an arbitrary action (Definition 5.1.1). From this partial formalization, the algorithm that translates between the existential and the profunctor form of an optic can be extracted.
- The formal coend derivations are explicitly written down in the code of the library. From these proof-relevant derivations, the algorithm that translates between the concrete and the existential form can be extracted. An example of the code for such a derivation can be seen in Figure 7.1 and Figure 7.2. Compare this to the proofs of Proposition 3.2.1 and Proposition 3.2.2. This makes the library extremely close to the theory and makes it possible for the code to justify its own correctness.

```
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 7.1: (Co)end derivation of the concrete form of a lens in Agda. Compare it to the proof of Proposition 3.2.1.

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

Figure 7.2: (Co)end derivation of the concrete form of a prism in Agda. Compare it to the proof of Proposition 3.2.2.

Chapter 8

Conclusions

8.1 Conclusions and further work

The definition of optics in terms of monoidal actions [Ril18] [BG18] captures in an elementary way all the basic optics, including traversals (Proposition 3.3.1). It is also a valuable tool for constructing new optics (§3.4), even if getting an interesting concrete form is not done in general. When studying the profunctor representation of optics, there seem to be reasons to prefer the more restrictive notion of *clear optics* (Definition 6.4.1). We think it is interesting to study how the different variations on the notion of *submonoidal category* of $[\mathbf{C}, \mathbf{C}]$ give rise to different variants of the definition of optic.

Among optics, the case of **traversals** was particularly interesting to us. Milewski [Mil17] posed the problem of finding an *as elementary as possible* description of the traversal. Our derivation in Proposition 3.3.1 tries to achieve precisely this goal. The derivation described in [Ril18] for *traversables* was using a parameterised comonad from [JO15], so it made sense to go back and try to simplify that approach with the *shape-contents* comonad we described in §4.1.2. Finally, describing power series functors as linear species is what guided us to the definition of *unsorted traversal* in Proposition 4.3.2. There is still work to be done for traversals that we summarize in the following questions.

- Can we explain the description of traversables in terms of monoidal, cartesian and cocartesian functors from [PGW17]? Note that *monoidal* profunctors do not fit into the usual pattern of Tambara modules we have been discussing so far.

Lawful optics are a topic that we have decided not to consider in this text. They are studied in detail in the brilliant work of Mitchell Riley [Ril18]. It would be particularly interesting to check what are sensible laws for the optics we have derived.

Combining different optics is one of the main motivations of the theory. With the description of the lattice in §6.4, they form a family of intercomposable bidirectional accessors. Going up in the lattice we forget the structure of our context, going down in the lattice we make it more explicit. Lenses, with their many applications ([GHWZ18], [FJ19]) seem to be a sweet spot in this hierarchy, but we believe the whole family of optics can have potential applications. Part of the work that remains to be done is to determine what are the practical applications of many optics we are describing, and to develop useful intuitions for them.

A final direction is to generalize the theory of optics. There is a particular way of doing so that seems to follow naturally from the proofs in this text: repeat the same theory in the context of *enriched* category theory. The reader can check that most of the text would work the same if we substitute **Sets** for an arbitrary cartesian Benabou cosmos $\mathcal{V}$. In some sense, this captures better its computational flavour, as it shows that the same theory could have been developed, for instance, over the category of directed-complete partial orders **Dcpo** (which is cartesian closed, complete and cocomplete [AJ94, Theorem 3.3.3]). Recall that we discussed in §7.2 how the theory was constructive in nature; we would expect to encounter no major problems when repeating our reasoning internally to topoi other than **Sets**.

8.2 A zoo of set-based optics

Name	Description (concrete form/monoidal action)	Proposition
Adapter	Identity functor $(s \to a) \times (b \to t)$	3.4.7
Lens	Product $(s \to a) \times (b \times s \to t)$	3.2.1
Prism	Coproduct $(s \to t + a) \times (b \to t)$	3.2.2
Grate	Exponential $((s \to a) \to b) \to t$	3.4.1
Glass	Product and Exponential $((s \to a) \to b) \to s \to t$	6.2.3
Affine Traversal	Product and Coproduct $s \to t + a \times (b \to t)$	6.2.2
Unsorted Traversal	Combinatorial species $s \to \sum_n a^n/n! \times (b^n/n! \to t)$	4.3.2
Traversal	Power series, Traversables $s \to \Sigma n.a^n \times (b^n \to t)$	3.3.1 and 4.2.1
Achromatic lens	Pointed product $(s \to (b \to t) + 1) \times (s \to a) \times (b \to t)$	3.4.2
Algebraic lens	Product by a monoid $(s \to a) \times (\psi s \times b \to t)$	3.4.5
Coalgebraic prism	Coproduct by a monoid $(s \to \psi t + a) \times (b \to t)$	3.4.5
Kaleidoscope	Applicative functors $\prod_n (a^n \to b) \to (s^n \to t)$	3.4.3
Setter	Any functor $(s \to a) \times (b \to t)$	3.4.6

Appendix: an alternative approach to traversables

Conjecture 8.2.1. *Let $\mathbf{S} \to [\mathbf{C}, \mathbf{C}]$ be the monoidal action of applicative functors such that the higher-order functor $\int_{F \in \mathbf{S}} \mathsf{Ran}_F F(-)$ is a comonad. Families of morphisms $\int_{F \in \mathbf{S}} TF \Rightarrow FT$ satisfying linearity and unitarity correspond to comonad algebras under the adjunction given by the right Kan extension.*

Assuming this result holds for the particular case where $\mathbf{S}$ is the category of traversable functors $\mathbf{App}$, we can again derive the traversal as the optic for traversables just by applying the construction of Proposition 3.4.10 to that comonad and then the Lemma 4.1.3.

In any case, the only thing we need to prove Proposition 4.2.1 is the construction of cofree traversables. We prove it here, to be able to claim that result even after considering Remark 4.1.6. The main idea on that proof would be that the comonad defined there preserves ends.

Proposition 8.2.2. *Let $T\colon \mathbf{Sets} \to \mathbf{Sets}$ be a traversable functor and let $H\colon \mathbf{Sets} \to \mathbf{Sets}$ be an arbitrary functor. There exists an adjunction $\mathbf{Trv}(T, KH) \cong [\mathbf{Sets}, \mathbf{Sets}](\mathcal{U}T, H)$, where $\mathcal{U}$ is the forgetful functor.*

Proof. We give KH traversable structure taking the morphisms $n_F\colon KH \circ F \to F \circ KH$ defined in Theorem 4.1.5, which satisfies unitarity and linearity. Note that $\mathrm{trv}\colon \int_F TF \to FT$ determines a natural transformation $\sigma\colon T \Rightarrow KT$ by Lemma 4.1.3. This is a morphism of traversables $\sigma \in \mathbf{Trv}(T, KT)$ because the two sides of the relevant commutative diagram have the same adjoint under the adjunction that defines right Kan extensions, as the following diagram shows.

$$
\begin{array}{c}
\begin{tikzcd} TF \arrow[r, "\sigma"] & (\mathsf{Ran}_F FT) \circ F \arrow[r, "\varepsilon"] \arrow[rr, bend left, "n_F"] & FT \arrow[r, "\sigma"] & F \circ (\mathsf{Ran}_F FT) \end{tikzcd} \\
\hline
T \xrightarrow{\sigma} \mathsf{Ran}_F FT \xrightarrow{\mathrm{id}} \mathsf{Ran}_F FT \xrightarrow{\sigma} \mathsf{Ran}_F F(\mathsf{Ran}_F FT) \\
\hline
TF \xrightarrow{\mathrm{trv}} FT \xrightarrow{\mathrm{id}} FT \xrightarrow{F\sigma} F(\mathsf{Ran}_F FT)
\end{array}
$$

Now, assume a morphism of traversables $f \in \mathbf{Trv}(T, KH)$; that is, a natural transformation between the underlying functors such that the following diagram commutes for every $F \in \mathbf{App}$.

$$
\begin{tikzcd}
T \circ F \arrow[r, "\mathrm{trv}"] \arrow[d, "f"] & F \circ T \arrow[d, "Ff"] \\
KH \circ F \arrow[r, "n_F"] & F \circ KH
\end{tikzcd}
$$

We want to show that there exists a unique natural transformation $g\colon \mathcal{U}T \Rightarrow H$ making f factor uniquely as $Kg \circ \sigma$. Note that the previous diagram becomes, under the adjunction, the following square, where δ and ε were defined in the comonad structure.

$$
\begin{tikzcd}
T \arrow[r, "\sigma"] \arrow[d, "f"] & KT \arrow[d, "Kf"] \\
KH \arrow[r, shift left, "\delta"] & K^2H \arrow[l, shift left, bend left, "K\varepsilon"]
\end{tikzcd}
$$

In the case where $f = Kg \circ \sigma$, the fact that $K\varepsilon \circ \delta = \mathrm{id}$ implies $K(\varepsilon \circ f) \circ \sigma = Kg \circ \sigma$. Note that $K(\varepsilon \circ f) \circ \sigma = f$. We will show that $g = \varepsilon \circ f$ from $K(\varepsilon \circ f) \circ \sigma = Kg \circ \sigma$, proving uniqueness. In fact, again because of the adjunction determining right Kan extensions, we have the following two adjunctions that must be equal for any $F \in \mathbf{App}$.

$$\frac{T \xrightarrow{\sigma} KT \underset{K(\varepsilon\circ f)}{\overset{Kg}{\rightrightarrows}} KH}{T \circ F \xrightarrow{\mathrm{trv}} F \circ T \underset{F(\varepsilon\circ f)}{\overset{Fg}{\rightrightarrows}} F \circ KH}$$

In the particular case where $F = \mathrm{id}$, we get $\mathrm{trv}_{\mathrm{id}} = \mathrm{id}$ because unitarity and then $g = \varepsilon \circ f$. This shows KH is the cofree traversal over H. $\square$

Bibliography

[AJ94] Samson Abramsky and Achim Jung. Handbook of logic in computer science (vol. 3). chapter Domain Theory, pages 1–168. Oxford University Press, Inc., New York, NY, USA, 1994.

[AMBL12] Jirí Adámek, Stefan Milius, Nathan Bowler, and Paul B Levy. Coproducts of monads on set. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, pages 45–54. IEEE Computer Society, 2012.

[ASCG+16] Faris Abou-Saleh, James Cheney, Jeremy Gibbons, James McKinna, and Perdita Stevens. Reflections on monadic lenses. In *A List of Successes that can Change the World*, pages 1–31. Springer, 2016.

[BG18] Guillaume Boisseau and Jeremy Gibbons. What you needa know about Yoneda: Profunctor optics and the Yoneda Lemma (functional pearl). *PACMPL*, 2(ICFP):84:1–84:27, 2018.

[BLL97] François Bergeron, Gilbert Labelle, and Pierre Leroux. *Combinatorial species and tree-like structures*, volume 67 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 1997.

[BM98] Richard Bird and Lambert Meertens. Nested datatypes. In *International Conference on Mathematics of Program Construction*, pages 52–67. Springer, 1998.

[Boi17] Guillaume Boisseau. Understanding profunctor optics: a representation theorem. Master's thesis, University of Oxford, 2017.

[CH+67] Thomas M Cover, Peter Hart, et al. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[CW01] Mario Cáccamo and Glynn Winskel. A higher-order calculus for categories. In *International Conference on Theorem Proving in Higher Order Logics*, pages 136–153. Springer, 2001.

[Fis36] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[FJ19] Brendan Fong and Michael Johnson. Lenses and learners. *CoRR*, abs/1903.03671, 2019.

[GdSO09] Jeremy Gibbons and Bruno C. d. S. Oliveira. The essence of the Iterator pattern. *The Journal of Functional Programming*, 19(3-4):377–402, 2009.

[GHWZ18] Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 472–481, 2018.

[Hed18] Jules Hedges. Limits of bimorphic lenses. *arXiv preprint arXiv:1808.05545*, 2018.

[HFM+19] Thomas Honeyman, Phil Freeman, Brian Marick, Lukas Heidemann, Christoph Hegemann, and Liam Goodacre. purescript-profunctor-lenses. GitHub, Purescript-contrib, `https://github.com/purescript-contrib/purescript-profunctor-lenses`, 2019.

[JC94] C Barry Jay and J Robin B Cockett. Shapely types and shape polymorphism. In *European Symposium on Programming*, pages 302–316. Springer, 1994.

[JO15] Mauro Jaskelioff and Russell O'Connor. A representation theorem for second-order functionals. *The Journal of Functional Programming*, 25, 2015.

[JR12] Mauro Jaskelioff and Ondrej Rypacek. An investigation of the laws of traversals. In *Proceedings Fourth Workshop on Mathematically Structured Functional Programming, MSFP@ETAPS 2012, Tallinn, Estonia, 25 March 2012.*, pages 40–49, 2012.

[Kel80] G Max Kelly. A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on. *Bulletin of the Australian Mathematical Society*, 22(1):1–83, 1980.

[Kme18] Edward Kmett. lens library, version 4.16. Hackage `https://hackage.haskell.org/package/lens-4.16`, 2012–2018.

[Koc09] Joachim Kock. Notes on polynomial functors. *Manuscript, version*, pages 08–05, 2009.

[Lor15] Fosco Loregian. This is the (co)end, my only (co)friend. *arXiv preprint arXiv:1501.02503*, 2015.

[Mak96] Michael Makkai. Avoiding the axiom of choice in general category theory. *Journal of Pure and Applied Algebra*, 108(2):109–173, 1996.

[Mar14] Dan Marsden. Category theory using string diagrams. *CoRR*, abs/1401.7220, 2014.

[Mil17] Bartosz Milewski. Profunctor optics: the categorical view. https://bartoszmilewski.com/2017/07/07/profunctor-optics-the-categorical-view/, 2017.

[ML71] Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 1971.

[MP08] Conor McBride and Ross Paterson. Applicative programming with effects. *The Journal of Functional Programming*, 18(1):1–13, 2008.

[Mye] David Jaz Myers. Personal communication, 25th July 2019.

[nLa18] nLab authors. HomePage. http://ncatlab.org/nlab/show/HomePage, May 2018. Revision 262.

[Nor] Ulf Norell. Dependently Typed Programming in Agda. In *Advanced Functional Programming, 6th International School, AFP 2008, Heijen, The Netherlands, May 2008, Revised Lectures*, pages 230–266.

[Ole82] Frank Joseph Oles. *A Category-theoretic Approach to the Semantics of Programming Languages*. PhD thesis, Syracuse, NY, USA, 1982. AAI8301650.

[PGW17] Matthew Pickering, Jeremy Gibbons, and Nicolas Wu. Profunctor optics: Modular data accessors. *Programming Journal*, 1(2):7, 2017.

[PS08] Craig Pastro and Ross Street. Doubles for monoidal categories. *Theory and applications of categories*, 21(4):61–75, 2008.

[Red95] Uday S Reddy. Monads and algebras - An introduction. 1995.

[Rie17] Emily Riehl. *Category theory in context*. Courier Dover Publications, 2017.

[Ril18] Mitchell Riley. Categories of optics. *arXiv preprint arXiv:1809.00738*, 2018.

[RJ17] Exequiel Rivas and Mauro Jaskelioff. Notions of computation as monoids. *Journal of functional programming*, 27, 2017.

[Spi19] David I Spivak. Generalized lens categories via functors $\mathcal{C}^{\mathrm{op}} \to \mathsf{Cat}$. *arXiv preprint arXiv:1908.02202*, 2019.

[Str72] Ross Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2(2):149–168, 1972.

[Tam06] Daisuke Tambara. Distributors on a tensor category. *Hokkaido mathematical journal*, 35(2):379–425, 2006.

[TLu] Tom Leinster (https://mathoverflow.net/users/586/tom-leinster). Coend computation. MathOverflow. https://mathoverflow.net/q/20451 (version: 2010-04-06).

[TVD14] Anne Sjerp Troelstra and Dirk Van Dalen. *Constructivism in mathematics*, volume 2. Elsevier, 2014.

[Yor14] Brent Yorgey. *Combinatorial species and labelled structures*. PhD thesis, University of Pennsylvania, 2014.