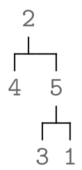
Traversables as Coalgebras

Mario Román

1 Introduction

Iterators are one of the most fundamental constructs in programming (GHJV93)[1]. They provide a way of sequentially accessing the elements of aggregated data structures, such as arrays, lists or trees. At the same time, they hide the details of their underlying implementations from the programmer. Most imperative languages support iteration in the form of a **foreach** construct. But in functional programming, iterators are encoded in *traversable* data structures (GdSO09), which provide a **traverse** method. For instance, we can *traverse* a binary tree while aggregating all its values.

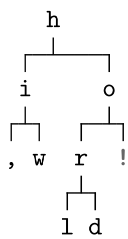



```
execWriterT $ traverse (tell . Sum) exampleTree
  >> Sum {getSum = 15}
```

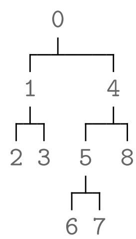
Traversables are usually described in terms of this **traverse** method. The method determines how the traversable data structure distributes over any effect that can be encoded into an *applicative* functor (MP08), see the following type signature. Moreover, it must satisfy certain axiomatic laws (JR12) that ensure that it actually represents a valid iterator.

```
traverse :: (Traversable t, Applicative f) => (a -> f b) -> t a -> f (t b)
```

The main idea of the present work is that traversables can be equivalently characterized as these data structures that can be split into their *shape* and their *contents*. For our purposes, the *contents* of the data structure are given by a list containing the elements inside the structure; whereas the *shape* will be a copy of the data structure labelled with increasing natural numbers that index the relative position of the elements. For instance,



gets split into



and `"hi,world!"`.

The advantage of this descripiton is twofold. On the one hand, valid *splits* turn out to be the coalgebras of a comonad that formally captures the idea of splitting into shape and contents. On the other hand, we recover a description of what it means to iterate a data structure that is much closer to our intuition, while still being equivalent to the original both in formulation and axioms.

[1] We will be talking about *internal* iterators, those where the programmer using them does not control the iterating behaviour explicitly. These stand in contrast to *external* iterators, which leave the responsability of actually iterating over the elements to the programmer.

2 Preliminaries (methods)

2.1 Coend calculus (procedure)

We shall employ during this text the techniques of coend calculus as described by Loregian (Lor15). Intuitively, ends are a form of universal quantification over the objects of a category that takes naturality conditions into account. Coends are their existential counterparts. When encoding ends and coends in functional programming languages, these are modelled using parametricity and existential types. Our goal is to prove the correspondence between the two descriptions of traversables in this generalized setting.

Definition 2.1. The **end** of a functor $P\colon \mathcal{C}^{op} \times \mathcal{C} \to \mathit{Set}$ is a set that we write as $\int_{X\in\mathcal{C}} P(X,X)$ endowed with a family of functions $\pi_A\colon \left(\int_{X\in\mathcal{C}} P(X,X)\right) \to P(A,A)$ for every $A \in \mathcal{C}$ such that, for any morphism $f\colon \mathcal{C}(A,B)$, they satisfy $P(\mathrm{id},f)\circ\pi_A = P(f,\mathrm{id})\circ\pi_B$. They are the universal such objects, in the sense that they are defined as the equalizer of the action of morphisms on both arguments of the functor.

$$\int_{X\in\mathcal{C}} P(X,X) \cong \mathrm{eq}\left(\textstyle\prod_{X\in\mathcal{C}} P(X,X) \rightrightarrows \prod_{f\colon A\to B} P(A,B) \right)$$

Dually, coends are coequalizers of a dual family of functions.

$$\int^{X\in\mathcal{C}} P(X,X) \cong \mathrm{coeq}\left(\textstyle\bigsqcup_{f\in\mathcal{C}(A,B)} P(A,B) \rightrightarrows \bigsqcup_{X\in\mathcal{C}} P(X,X) \right)$$

Ends and coends are thus particular cases of limits and colimits. They are unique up to isomorphism when they exist. They are functorial. Continuous functors preserve ends and cocontinuous functors preserve coends. In particular, we have isomorphisms that reflect the usual rules of quantification.

$$\int_{X\in\mathcal{C}} (P(X,X)\to D) \cong \left(\int^{X\in\mathcal{C}} P(X,X)\right) \to D$$
$$\int_{X\in\mathcal{C}} (D\to P(X,X)) \cong D \to \left(\int_{X\in\mathcal{C}} P(X,X)\right).$$

We have also a commutativity of quantifiers, that is sometimes called by analogy the *Fubini rule.*

$$\begin{aligned} &\int_{(X,Y)\in\mathcal{C}\times\mathcal{D}} P(X,X,Y,Y) \\ \cong &\int_{X\in\mathcal{C}}\int_{Y\in\mathcal{D}} P(X,X,Y,Y) \\ \cong &\int_{Y\in\mathcal{D}}\int_{X\in\mathcal{C}} P(X,X,Y,Y). \end{aligned}$$

The same rule applies for coends. The usefulness of these constructions comes however from the compactness which we can write the two versions of the Yoneda reduction.

Lemma 2.2 (Yoneda reduction)**.** *For any functor $F\colon \mathcal{C}\to \mathit{Set}$, we have canonical isomorphisms*

$$FA \cong \int_{X\in\mathcal{C}} \mathcal{C}(A,X)\to FX, \qquad FA \cong \int^{X\in\mathcal{C}} FX\times\mathcal{C}(X,A),$$

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

An intuition on this form of Yoneda lemma is that it allows us to integrate interpreting the hom-functor as a Dirac's delta for ends. This is Remark 2.6 in (Lor15).

2.2 In functional programming (assumptions)

The present text works over Set, the category of sets. This means that we are implicitly assuming the existence of every end and coend, as this follows from the completeness and cocompleteness of Set. On the practical side, we are assuming at the same time that ends correspond to parametricity, even if they are less general; and that coends correspond to some form existential types. These simplifying assumptions are expected to hold only in some idealized functional programming language. In particular, we take transparential references for granted and we do not deal with instances of non-termination. This approach will make many of our computations straightforward and would allow easily for an extension to any enriched setting.

2.3 Limitations of this approach (limitations)

As it stands, however, the usage of coend calculus presents some limitations.

- It is not easy to deal with non-reversible arrows, and many instances of parametericity will not be captured by the dinaturality conditions of a coend. In our particular use case, this will not present a problem.
- Coend calculus works over any base of enrichment, but here we limit ourselves to work on sets for simplicity. Changing the base of enrichment would allow us to deal with non termination (in a category of directed continuous partial orders, for instance).
- Coend calculus is not a good tool for analizying non-reversible transformations, and that will require some parts of our discussion to rely on alternative techniques.

3 Results

3.1 The shape and contents comonad

We will be studying the following higher-order functor $K\colon [Set, Set] \to [Set, Set]$ defined as

$$KT(A) = \sum_{n \in \mathcal{N}} T(n) \times A^n.$$

Here $\mathcal{N}$ is meant to be the free monoidal category on one object, with an implict inclusion into Set determined by the terminal object. The functor K 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 $n \in \mathbb{N}$ being the *length*, $s \in T(n)$ being the *shape*, and $c \in A^n$ being the *contents*. The inspiration for considering this functor 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.

The reader familiar with traversables can see that, actually, the elements of $T(n)$ are more thanthe 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.

Construction 3.1. *A functor $K\colon [Set, Set] \to [Set, Set]$, defined on objects by $KT(A) = \sum_{n \in \mathbb{N}} T(n) \times A^n$, and a comonad structure for it.*

Proof

First, we see that this defines a functor. We start by checking that the assignment $A \mapsto \sum_{n \in \mathbb{N}} T(n) \times A^n$ extends to a functor for any $T \in [Set, Set]$. 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. We have shown that K actually outputs functors in $[Set, Set]$, let us describe its action on natural transformations. Given any natural transformation $h\colon T \Rightarrow R$, we can define $Kh\colon KT \Rightarrow KR$ as $Kh(n; s, c) = (n; h(s), c)$, which can be checked to also be functorial and natural.

Secondly, we proceed to describe the comonad structure. In order to define the *counit* $\varepsilon \colon KT \Rightarrow T$, we consider, for every $n \in \mathcal{N}$, the morphism

$$T(n) \times A^n \xrightarrow{\text{map}} T(n) \times T(A)^{T(n)} \xrightarrow{\text{ev}} T(A).$$

In other words, $\varepsilon(n; s, c) = Tc(s)$. In order to define the *comultiplication* $\delta \colon KT \Rightarrow K^2T$, we consider, for every $n \in \mathcal{N}$, the morphism

$$T(n) \times A^n \xrightarrow{\text{id}} T(n) \times n^n \times A^n.$$

In other words, $\delta(n; s, c) = (n; (n; s, \text{id}), c)$.

We will check now the comonad axioms. *Counitality* is the fact that the following two diagrams commute.

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

We provide an equational proof.

$K\varepsilon_T(\delta(n; s, c))$	$\varepsilon_{KT}(\delta(n; s, c))$
$=$ (Definition of δ)	$=$ (Definition of δ)
$K\varepsilon_T(n; (n, s, \text{id}), c)$	$\varepsilon_{KT}(n; (n; s, \text{id}), c)$
$=$ (Definition of K)	$=$ (Definition of ε)
$(n; \varepsilon_T(n, s, \text{id}), c)$	$KTc(n; s, \text{id})$
$=$ (Definition of ε)	$=$ (Definition of KT)
$(n; T(\text{id})(s), c)$	$(n; s, c \circ \text{id})$
$=$ (Identity)	$=$ (Identity)
$(n; s, c)$	$(n; s, c)$.

Coassociativity is the fact that the following diagram commutes.

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

We provide again an equational proof.

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

This finishes the construction of a comonad over K.

We are interested into the coalgebras of this comonad; even if the reader could deduce the coalgebra axioms from the structure, we will write them explicitly and comment on them. Let

$$\sigma\colon T(A) \to \sum_{n \in \mathcal{N}} T(n) \times A^n,$$

be a K-coalgebra. Its first axiom is *counitality* and, in our case, it says that the following diagram commutes.

$$\begin{tikzcd} T(A) \arrow[r, "\sigma"] \arrow[dr, "\mathrm{id}"'] & \sum_n T(n) \times A^n \arrow[d, "\varepsilon_T"] \\ & T(A) \end{tikzcd}$$

Equationally, if $\sigma(t) = (n; s, c)$, we have that $Tc(s) = t$. Intuitively, 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.

$$\begin{tikzcd} T(A) \arrow[rr, "\sigma"] \arrow[d, "\sigma"'] & & \sum_n T(n) \times A^n \arrow[d, "K\sigma"] \\ \sum_n T(n) \times A^n \arrow[r, "\delta"] & \sum_n \left(\sum_m T(m) \times n^m\right) \times A^n & \end{tikzcd}$$

Equationally, if $\sigma(t) = (n; s, c)$, we have that $\sigma(s) = (n; s, \mathrm{id})$. Intuitively, the shape of a shape s is again s. In this sense, taking the shape is idempotent. The intuitive explanation of the coalgebra axioms should convince the reader of our choice of naming the comonad K.

3.2 Putting an end to parameters

3.2.1 End-preserving functors

Definition 3.2. A functor $F\colon \mathcal{C} \to \mathcal{C}$ *preserves ends* if, for every profunctor $P\colon \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$ there exists an isomorphism $\psi\colon F \int_X P(X,X) \cong \int_X FP(X,X)$ natural in P and such that the following diagram commutes for every $A \in \mathcal{C}$.

$$\begin{tikzcd} F \int_X P(X,X) \arrow[d, "\psi"'] \arrow[dr, "F\pi_A"] & \\ & FP(A,A) \\ \int_X FP(X,X) \arrow[ur, "\pi_A"'] & \end{tikzcd}$$

3.2.2 Parameterised comonads

Definition 3.3. A **parameterised comonad** is a triple (R, ε, δ) given by a functor $R\colon \mathcal{C} \times \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$; a family of morphisms $\varepsilon_{A,X}\colon R_{A,A}X \to X$ dinatural in A and natural in X, called the **counit**; and a family of morphisms $\delta_{A,B,C,X}\colon R_{A,C}X \to R_{A,B}(R_{B,C}X)$ natural in A, C, X and dinatural in B. It must satisfy **counitality** and **coassociativity** as described by the commutativity of the following diagrams.

$$\begin{tikzcd} & R_{A,B}X \arrow[dl, "\delta_{A,B,B,X}"'] \arrow[d, "\mathrm{id}"] \arrow[dr, "\delta_{A,A,B,X}"] & \\ R_{A,B}R_{B,B}X \arrow[r, "R_{A,B}(\varepsilon_{B,X})"'] & R_{A,B}X & R_{A,A}R_{A,B}X \arrow[l, "\varepsilon_{A,R_{A,B}X}"] \end{tikzcd}$$

$$\begin{tikzcd}
R_{A,D}X \arrow[r, "\delta_{A,B,D,X}"] \arrow[d, "\delta_{A,C,D,X}"'] & R_{A,B}R_{B,D}X \arrow[d, "R_{A,B}(\delta_{B,C,D,X})"] \\
R_{A,C}R_{C,D}X \arrow[r, "\delta_{A,B,C,R_{C,D}X}"] & R_{A,B}R_{B,C}R_{C,D}X
\end{tikzcd}$$

Remark 3.4. In general a parameterised comonad will not induce a comonad. But there are cases where it does. We will show that it does if the parameterised comonad preserves ends; that is, there exists some $\psi\colon R_{A,B}\int_X(-) \to \int_X R_{A,B}(-)$ satisfying the conditions of Definition 3.2.

First, let $\Theta\colon [\mathcal{C},\mathcal{C}] \to [\mathcal{C},\mathcal{C}]$ be a functor that takes a natural transformation $\alpha\colon T \Rightarrow H$ into the associated $\Theta T \Rightarrow \Theta H$ defined by the universal property of the end in the following diagram.

$$\begin{tikzcd}
\int_X R_{A,X}TX \arrow[r, "\pi_B"] \arrow[d, "(\Theta\alpha)_A"'] & R_{A,B}TB \arrow[d, "R_{A,B}\alpha_B"] \\
\int_X R_{A,X}HX \arrow[r, "\pi_B"] & R_{A,B}HB
\end{tikzcd}$$

Now we can show that $(\Theta T)A = \int_X R_{A,X}TX$ can be made into a comonad where the **counit** $\varepsilon'_T\colon \Theta T \Rightarrow T$ is given by projecting and then using the parameterised counit; and where the **comultiplication** $\delta'_T\colon \Theta T \Rightarrow \Theta(\Theta T)$ is given by the parameterised comultiplication composed with ψ^{-1}.

$$\begin{tikzcd}
\int_B R_{A,B}TB \arrow[r, "\pi_A"] \arrow[dr, "\varepsilon'_{T,A}"'] & R_{A,A}TA \arrow[d, "\varepsilon_{A,TA}"] \\
 & TA
\end{tikzcd}
\qquad
\begin{tikzcd}
\int_C R_{A,C}TC \arrow[r, "\int_C \delta_{A,B,C,TC}"] \arrow[d, "\delta'_{T,A}"'] \arrow[dr, dashed, "\exists!"] & \int_C R_{A,B}R_{B,C}TC \\
\int_B R_{A,B}\int_C R_{B,C}TC \arrow[r, "\psi_{T,A}"'] & \int_C\int_B R_{A,B}R_{B,C}TC \arrow[u, "\pi_B"']
\end{tikzcd}$$

Before proving this is in fact a comonad, we can prove a technical lemma about how to use the property of ψ without making explicit mention to it.

Lemma 3.5. *In the conditions of the previous definition, for any $A, B, C \in \mathcal{C}$ and any functor $T \in [\mathcal{C},\mathcal{C}]$,*

$$R_{A,B}\pi_D \circ \pi_B \circ \delta'_{T,A} = \delta_{A,B,D} \circ \pi_B.$$

Proof

This amounts to check that the following diagram commutes.

$$\begin{tikzcd}
\int_Y R_{A,Y}TY \arrow[rrrr, "\delta'_T"] \arrow[dddd, "\pi_C"'] \arrow[ddr, "\int_Y \delta_{B,Y}"] \arrow[drr, phantom, "\textcircled{1}"] & & & & \int_X R_{A,X}\int_Y R_{X,Y}TY \arrow[dll, "\psi"'] \arrow[ddl, "\int_X R_{A,X}\pi_C"'] \arrow[ddd, "\pi_B"] \arrow[ddl, phantom, bend left, "\textcircled{2}"] \\
 & & \int_X\int_Y R_{A,X}R_{X,Y}TY \arrow[dl, "\pi_C"'] \arrow[dr, "\pi_B"] & & \\
 & \int_Y R_{A,B}R_{B,Y}TY \arrow[ddrrr, "\pi_C"] & & \int_X R_{A,X}R_{X,C}TC \arrow[dr, "\pi_B"'] & \\
 & & & & R_{A,B}\int_Y R_{B,Y}TY \arrow[d, "R_{A,B}\pi_C"] \\
R_{A,C}TC \arrow[rrrr, "\delta_{A,B,C}"'] & & & & R_{A,B}R_{B,C}TC
\end{tikzcd}$$

Area 1 commutes because of the definition of δ'. Area 2 does because of the property we asked of ψ. The other three areas commute because of naturality.

Proposition 3.6. *The triple $(\Theta, \varepsilon', \delta')$ we constructed in Remark 3.4 determines a comonad.*

Proof

We will prove counitality and coassociativity. For left **counitality** on the component determined by T and A, we will prove that $(\Theta\varepsilon'_T)_A \circ (\delta'_T)_A = \mathrm{id}$. By the universal property of the end, that is the same as to prove it after applying a projection on an arbitrary B. Diagramatically, we are saying that the following diagram commutes. Here, area ① commutes by Lemma 3.5; area ② does by definition of Θ; and area ③ does by definition of ε'.

$$\begin{tikzcd}
\int_Y R_{A,Y}TY \arrow[r, "\delta'_{T,A}"] \arrow[d, "\pi_B"'] \arrow[dr, phantom, "①"] & \int_X R_{A,X}\left(\int_Y R_{A,Y}TY\right) \arrow[r, "(\Theta\varepsilon'_T)_A"] \arrow[d, "\pi_B"] \arrow[dr, phantom, "②"] & \int_X R_{A,X}TX \arrow[d, "\pi_B"] \\
R_{A,B}TB \arrow[dr, "\delta_{A,B,B}"'] & R_{A,B}\left(\int_Y R_{B,Y}TY\right) \arrow[r, "R_{A,B}(\varepsilon'_{T,B})"] \arrow[d, "R_{A,B}\pi_B"] \arrow[r, phantom, shift right=3ex, "③"] & R_{A,B}TB \\
& R_{A,B}R_{B,B}TB \arrow[ur, "R_{A,B}\varepsilon"'] &
\end{tikzcd}$$

Next, we prove right **counitality** on the component determined by T and A; that is, that $\varepsilon'_{A,\Theta TX} \circ (\delta'_T)_A = \mathrm{id}$. Again by the universal property of the end, it suffices to prove it after applying a projection on an arbitrary B. Diagramatically, we are saying that the following diagram commutes. Here area ①, after applying naturality, is the one in Lemma 3.5; area ② does by ε' being a natural transformation; and area ③ does by definition of ε'.

$$\begin{tikzcd}
\int_Y R_{A,Y}TY \arrow[r, "\delta'_{T,A}"] \arrow[d, "\pi_B"] \arrow[dr, phantom, "①"] & \int_X R_{A,X}\int_Y R_{X,Y}TY \arrow[r, "\varepsilon'_{A,\Theta TX}"] \arrow[d, "\int_X R_{A,X}\pi_B"] \arrow[dr, phantom, "②"] & \int_Y R_{A,Y}TY \arrow[d, "\pi_B"] \\
R_{A,B}TB \arrow[dr, "\delta_{A,A,B}"'] & \int_X R_{A,X}R_{X,B}TB \arrow[r, "\varepsilon'_{A,R_{A,B}TB}"] \arrow[d, "\pi_A"] \arrow[r, phantom, shift right=3ex, "③"] & R_{A,B}TB \\
& R_{A,A}R_{A,B}TB \arrow[ur, "\varepsilon_{A,R_{A,B}K}"'] &
\end{tikzcd}$$

Finally, we prove **coassociativity** with the following diagram. Note that this does not show that the upper and lower row are the same morphism, but they are the same after some projections.

$$\begin{tikzcd}
\int_Y R_{A,Y}TY \arrow[r, "\delta'_{T,A}"] \arrow[d, "\pi_B"] & \int_X \int_Y R_{X,Y}TY \arrow[r, "(\Theta\delta'_T)_A"] \arrow[d, "\pi_B"] & \int_X R_{A,X}\int_Y R_{X,Y}\int_Z R_{Y,Z}TZ \arrow[d, "\pi_B"] \\
R_{A,B}TB \arrow[dr, "\delta_{A,B,D}"'] \arrow[dddd, "\mathrm{id}"] & R_{A,B}\int_Y R_{B,Y}TY \arrow[r, "R_{A,B}(\delta'_{T,A})"] \arrow[d, "R_{A,B}\pi_D"] & R_{A,B}\int_Y R_{B,Y}\int_Z R_{Y,Z}TZ \arrow[d, "\pi_D"] \\
& R_{A,B}R_{B,D}TD \arrow[dr, "R_{A,B}\delta_{B,C,D}"'] & R_{A,B}\int_Y R_{B,Y}R_{Y,D}TD \arrow[d, "R_{A,B}\pi_C"] \\
& & R_{A,B}R_{B,C}R_{C,D}TD \\
& R_{A,C}R_{C,D}TD \arrow[ur, "\delta_{A,B,C}"] & R_{A,B}R_{B,C}\int_Z R_{C,Z}TZ \arrow[u, "R_{A,B}R_{B,C}\pi_D"'] \\
R_{A,B}TB \arrow[ur, "\delta_{A,C,D}"] & R_{A,B}\int_Z R_{X,Z}TZ \arrow[u, "R_{A,B}\pi_D"'] \arrow[ur, "\delta_{A,B,C}"] & R_{A,B}\int_Y R_{B,Y}\int_Z R_{Y,Z}TZ \arrow[u, "R_{A,B}\pi_C"'] \\
\int_Y R_{A,Y}TY \arrow[r, "\delta'_{T,A}"] \arrow[u, "\pi_B"] & \int_X R_{A,X}\int_Z R_{X,Z} \arrow[r, "\delta'_{\Theta T,A}"] \arrow[u, "\pi_B"] & \int_X R_{A,X}\int_Y R_{X,Y}\int_Z R_{Y,Z}TZ \arrow[u, "\pi_B"']
\end{tikzcd}$$

The projections on the side can be rewritten using ψ to be the three projections of an end over three variables, thus showing that the two sides must be equivalent.

3.2.3 Parameterised coalgebras

Definition 3.7. A **coalgebra** for a parameterised comonad (R, ε, δ) is a pair (T, k) where $T\colon \mathcal{C} \to \mathcal{C}$ is a functor and $k_{A,B}\colon TA \to R_{A,B}TB$ is a family of morphisms natural in A and dinatural in B. It must be such that the following diagrams commute.

$$\begin{tikzcd}
TA \arrow[r, "k_{A,B}"] \arrow[d, "k_{A,C}"'] & R_{A,B}TB \arrow[d, "R_{A,B}k_{B,C}"] \\
R_{A,C}TC \arrow[r, "\delta_{A,B,C}"'] & R_{A,B}R_{B,C}TC
\end{tikzcd}
\qquad
\begin{tikzcd}
TA \arrow[r, "k_{A,A}"] \arrow[dr, "\mathrm{id}"'] & R_{A,A}TA \arrow[d, "\varepsilon_A"] \\
 & TA
\end{tikzcd}$$

Note that the fact that a family of morphisms $k_{A,B}\colon TA \to R_{A,B}TB$ dinatural in B can be equivalently rewritten as a family of morphisms $k'_A\colon TA \to \Theta TA$. We will show now that coalgebras for a parameterised comonad $k_{A,B}\colon TA \to R_{A,B}TB$ are equivalently coalgebras $k'_A\colon TA \to \Theta TA$ for the comonad.

3.2.4 From parameterised coalgebras to coalgebras

Proposition 3.8. *The family of morphisms $k_{A,B}\colon TA \to R_{A,B}TB$ is a coalgebra for the parameterised comonad (R, ε, δ) if and only if it is a coalgebra for the associated comonad $(\Theta, \varepsilon', \delta')$.*

Proof

We first prove that *counitality* in both cases is the same. Note that in the following diagram the two triangles commute by definition and the two sides of the square equate the two counitality conditions.

$$\begin{tikzcd}
TA \arrow[r, "k"] \arrow[d, "k_{A,A}"'] & \int_X R_{A,X}TX \arrow[dl, "\pi_A"] \arrow[d, "\varepsilon'_{T,A}"] \\
R_{A,A}TA \arrow[r, "\varepsilon_A"'] & TA
\end{tikzcd}$$

We need to prove now that *comultiplicativity* in both cases is the same. Consider first the following diagram, where we can see that comultiplicativity for the parameterized coalgebra implies comultiplicativity for the coalgebra.

$$\begin{tikzcd}
TA \arrow[r, "k"] \arrow[dr, "k_{A,B}"'] \arrow[dddd, "\mathrm{id}"'] & \int_Y R_{A,Y}TY \arrow[r, "\delta'"] \arrow[d, "\pi_B"] & \int_X \int_Y R_{X,Y}TY \arrow[d, "\pi_B"] \\
 & R_{A,B}TB \arrow[dr, "\delta_{A,B,C}"'] & R_{A,B} \int_Y R_{B,Y}TY \arrow[d, "R_{A,B}\pi_C"] \\
 & & R_{A,B}R_{B,C}TC \\
 & R_{A,B}TB \arrow[ur, "R_{A,B}k_{B,C}"] \arrow[r, "R_{A,B}k"'] & R_{A,B} \int_Y R_{B,Y}TY \arrow[u, "R_{A,B}\pi_C"'] \\
TA \arrow[r, "k"'] \arrow[ur, "k_{A,B}"] & \int_X R_{A,X}TX \arrow[r, "\Theta k"'] \arrow[u, "\pi_B"] & \int_X R_{A,X} \int_Y R_{X,Y}TY \arrow[u, "\pi_B"']
\end{tikzcd}$$

On the other hand, we will show that comultiplicativity for the coalgebra implies its parameterised counterpart. We can rewrite $R_{A,B}\pi_C \circ \pi_B$ as $\pi_B \circ \pi_C \circ \psi$ because of the property of ψ. This means that $\psi \circ \delta' \circ k = \psi \circ \Theta k \circ k$ by the universal property of the ends; because ψ is an isomorphism, we finally get $\delta' \circ k = \Theta k \circ k$.

We are in these conditions for the case of traversables. Using the result by Jaskelioff-O'Connor, traversables are coalgebras for that associated (non-parameterised!) comonad.

3.3 Traversables

Definition 3.9. Definition 3.3 A *traversable* structure on a functor $T\colon Set \to Set$ is a family of natural transformations

$$\mathrm{trv}_{F,X}\colon T(F(X)) \to F(T(X))$$

natural in $F \in$ which additionally satisfy two laws called *unitarity* and *linearity*.

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

A characterization of traversables in terms of parameterised coalgebras is due to Jaskelioff and O'Connor.

Proposition 3.10. *§5.3 Traversable structures correspond to parameterized coalgebra structures for the parameterized comonad defined by*

$$\mathcal{A}_{A,B}(X) = \sum_{n\in\mathcal{N}} A^n \times (B^n \to X).$$

We can identify the similarities with the *shape-contents* comonad described before, and in fact we will show that it is precisely the comonad associated to this parameterised comonad.

Proposition 3.11. *The functor $\mathcal{A}_{A,B}$ preserves ends for any $A, B \in Set$.*

Proof

We first construct the isomorphism ψ as in Definition 3.2.

$$\begin{aligned}
& \int_{X\in Set} \sum_{n\in\mathcal{N}} A^n \times (B^n \to P(X,X)) \\
\cong \quad & \text{(Discrete colimits commute with connected limits)} \\
& \sum_{n\in\mathcal{N}} \int_{X\in Set} A^n \times (B^n \to P(X,X)) \\
\cong \quad & \text{(Limits commute)} \\
& \sum_{n\in\mathcal{N}} \left(\int_{X\in Set} A^n\right) \times \left(\int_{X\in Set} B^n \to P(X,X)\right) \\
\cong \quad & \text{(Connected limit over a constant functor)} \\
& \sum_{n\in\mathcal{N}} A^n \times \left(\int_{X\in Set} B^n \to P(X,X)\right) \\
\cong \quad & \text{(Continuity of limits)} \\
& \sum_{n\in\mathcal{N}} A^n \times \left(B^n \to \int_{X\in Set} P(X,X)\right).
\end{aligned}$$

Showing that this preserves projections is direct from the definition.

4 Conclusions

We have proven a characterization of traversables as coalgebras for a fairly simple comonad 3.11. Jaskelioff and O'Connor §5.3 showed that traversable structures over an endofunctor $T\colon Set \to Set$ are precisely parameterised coalgebras $t_{A,B}\colon TA \to \sum_{n\in\mathbb{N}} A^n \times (B^n \to TB)$ for the free applicative parameterised comonad. Atkey §2 shows this determines an parameterised adjunction in suitable cases. Our main lemma can be used to transform this parameterised comonad 3.3 into the comonad that has traversable as coalgebras.

This result can be used to simplify already existing results in the literature. For instance, consider the derivation of traversals by Riley §4.6 who used the parameterized adjunction to determine cofree traversables. After this characterization, the proof there simplifies as follows for any $F \in \mathbf{Trv}$ and $H \in [Set, Set]$, then

$$
\begin{aligned}
& \int_{A \in Set} \mathbf{Set}(FA, HA) \\
\cong \quad & \text{(Parameterised adjunction)} \\
& \int_{A \in Set} \mathbf{Trv}\left(F, \sum\nolimits_{n \in \mathbb{N}} (-)^n \times (A^n \to HA)\right) \\
\cong \quad & \text{(Continuity)} \\
& \mathbf{Trv}\left(F, \int_{A \in Set} \sum\nolimits_{n \in \mathbb{N}} (-)^n \times (A^n \to HA)\right) \\
\cong \quad & \text{(Discrete colimits commute with connected limits)} \\
& \mathbf{Trv}\left(F, \sum\nolimits_{n \in \mathbb{N}} (-)^n \times \int_{A \in Set} (A^n \to HA)\right) \\
\cong \quad & \text{(Yoneda lemma)} \\
& \mathbf{Trv}\left(F, \sum\nolimits_{n \in \mathbb{N}} (-)^n \times H(n)\right).
\end{aligned}
$$

Our results heavily rely in notion of preservation of ends by the relevant functors 3.2. We propose the application of similar techniques, relying in coend calculus, to problems in a similar setting (categories enriched over a cartesian Benabou cosmos), such as the ones arising from the practice of functional programming.

Traversables are precisely those functors with a valid shape and contents split. This statement is closer to our intuition than previous characterizations and can be made formal connecting the descriptions of traversables to coend calculus.

References

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

Erich Gamma, Richard Helm, Ralph E. Johnson, and John M. Vlissides. Design patterns: Abstraction and reuse of object-oriented design. In *ECOOP'93 - Object-Oriented Programming, 7th European Conference, Kaiserslautern, Germany, July 26-30, 1993, Proceedings*, pages 406–431, 1993.

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

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.

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

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