# Realization of Coinductive Types

Dexter Kozen
Department of Computer Science
Cornell University
Ithaca, New York 14853–7501, USA
Email: kozen@cs.cornell.edu

February 28, 2011

**Abstract**

We give an explicit combinatorial construction of final coalgebras for a modest generalization of polynomial functors on Set. Type signatures are modeled as directed multigraphs instead of endofunctors. The final coalgebra for a type signature $F$ involves the notion of Brzozowski derivative on sets of paths in $F$.

## 1 Introduction

Final $F$-coalgebras for endofunctors $F$ on Set are useful in defining semantics of coinductive datatypes. The existence of final coalgebras under very general conditions has been studied in several papers [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. These studies are mostly undertaken from an abstract categorical viewpoint, typically involving inverse limits, Cauchy completions, or bisimulation quotients of large powers. Aside from a few specific examples [6, 7], general *concrete* constructions seem to be lacking. Such constructions would be of great use to anyone interested in formal semantics and logics for reasoning about coinductive datatypes. Of lesser concern, but still an issue, is that the traditional representation of a type signature as a set functor introduces an undesirable asymmetry in the case of mutually recursively defined types.

Ordinary deterministic finite automata over an alphabet $\Sigma$ form a family of coalgebras of a particularly simple form [7, 11]. Final coalgebras of this type can be constructed explicitly in terms of the *Brzozowski derivative*

$$D_a(A) = \{x \mid ax \in A\}$$

for $A \subseteq \Sigma^*$ and $a \in \Sigma$.

In this note we give an explicit Brzozowski-like construction of final coalgebras for a modest generalization of polynomial functors on Set. These functors are built from product, coproduct, total and partial functions from a fixed set, constant functors, and compositions thereof. However, instead of functors, we represent type signatures as certain directed multigraphs. This not only addresses the issue of asymmetry mentioned above, but also provides a platform for a definition of a Brzozowski derivative on sets of paths.

## 2 Directed Multigraphs

A *directed multigraph* is a structure $G = (V, E, \mathsf{src}, \mathsf{tgt})$ with vertices $V$, directed edges $E$, and two maps $\mathsf{src}, \mathsf{tgt} : E \to V$ giving the source and target of each edge, respectively. We write $e : s \to t$ if $s = \mathsf{src}\, e$ and $t = \mathsf{tgt}\, e$. When specifying multigraphs, we will sometimes use the notation $s \xrightarrow{n} t$ for the metastatement, "There are exactly $n$ edges from $s$ to $t$."

A *path* is a finite alternating sequence of nodes and edges

$$s_0\, e_0\, s_1\, e_1\, s_2\, \cdots\, s_{n-1}\, e_{n-1}\, s_n,$$

$n \geq 0$, such that $e_i : s_i \to s_{i+1}$, $0 \leq i \leq n-1$. These are the arrows of the free category generated by $G$. The *length* of a path is the number of edges. A path of length 0 is just a single node. The first and last states of a path $p$ are denoted $\mathsf{src}\, p$ and $\mathsf{tgt}\, p$, respectively. As with edges, we write $p : s \to t$ if $s = \mathsf{src}\, p$ and $t = \mathsf{tgt}\, p$.

A *multigraph homomorphism* $\ell : G_1 \to G_2$ is a map $\ell : V_1 \to V_2$, $\ell : E_1 \to E_2$ such that if $e : s \to t$ then $\ell(e) : \ell(s) \to \ell(t)$. This lifts to a functor on the free categories generated by $G_1$ and $G_2$.

A *type signature* is a directed multigraph $F$ along with a designation of each node of $F$ as either *existential* or *universal*. The existential and universal nodes correspond respectively to coproduct and product constructors. The directed edges of the graph represent the corresponding destructors.

For example, consider an algebraic signature consisting of a binary function symbol $f$, a unary function symbol $g$, and a constant $c$. This would ordinarily be represented by the polynomial endofunctor $F = -^2 + - + \mathbb{1}$, or in OCaml by

```
type t = F of t * t | G of t | C
```

We would represent this signature by a directed multigraph consisting of four nodes $\{t, f, g, c\}$, of which $t$ is existential and $f, g, c$ are universal, along with edges

$$t \xrightarrow{1} f \qquad t \xrightarrow{1} g \qquad t \xrightarrow{1} c \qquad f \xrightarrow{2} t \qquad g \xrightarrow{1} t.$$

Here is a more involved example from [12]. In that paper, the state of a computation of a higher-order language with closures is defined in terms of a recursive type definition

$$\begin{aligned} \mathsf{Val} &= \mathsf{Const} + \mathsf{Cl} && \text{values} \\ \mathsf{Cl} &= \lambda\text{-}\mathsf{Abs} \times \mathsf{Env} && \text{closures} \\ \mathsf{Env} &= \mathsf{Var} \rightharpoonup \mathsf{Val} && \text{closure environments} \end{aligned}$$

where Const is a fixed set of constants, $\lambda$-Abs is a fixed set of $\lambda$-abstractions, and Var is a fixed set of variables. The exact nature of these sets is not important here. The set of values is a solution to the recursive equation

$$\mathsf{Val} = \mathsf{Const} + (\lambda\text{-}\mathsf{Abs} \times (\mathsf{Var} \rightharpoonup \mathsf{Val})),$$

which would ordinarily be modeled by an endofunctor

$$F = \mathsf{Const} + (\lambda\text{-}\mathsf{Abs} \times (\mathsf{Var} \rightharpoonup -))$$

on Set. In OCaml, we might write

```
type value = Const of int | Closure of closure
and closure = labs * env
and env = var -> value
```

We model this type signature by a multigraph with existential nodes Val, Const, $\lambda$-Abs, and Env and universal nodes Cl, $\mathbb{1}$, and a node for each $B \subseteq \mathsf{Var}$. The edges are

$$\begin{aligned} \mathsf{Val} &\xrightarrow{1} \mathsf{Const} & \qquad \mathsf{Val} &\xrightarrow{1} \mathsf{Cl} \\ \mathsf{Cl} &\xrightarrow{1} \lambda\text{-}\mathsf{Abs} & \qquad \mathsf{Cl} &\xrightarrow{1} \mathsf{Env} \\ \mathsf{Const} &\xrightarrow{c} \mathbb{1},\ c = |\mathsf{Const}| & \qquad \lambda\text{-}\mathsf{Abs} &\xrightarrow{d} \mathbb{1},\ d = |\lambda\text{-}\mathsf{Abs}| \\ \mathsf{Env} &\xrightarrow{1} B,\ B \subseteq \mathsf{Var} & \qquad B &\xrightarrow{b} \mathsf{Val},\ b = |B| \end{aligned}$$

Note that we are regarding a partial function $\mathsf{Var} \rightharpoonup \mathsf{Val}$ on the fixed set Var as a dependent coproduct $\sum_{B \subseteq \mathsf{Var}} \mathsf{Val}^B$. This is modeled by an existential node to select the domain $B \subseteq \mathsf{Var}$, followed by a universal node to select the value of the function $\mathsf{Val}^B$ on that domain.

## 2.1 Coalgebras and Realizations

Let $F$ be a type signature. An *$F$-coalgebra* is a mapping that associates a pair $(A_s, \alpha_s)$ with each node $s$ of $F$, where the $A_s$ are pairwise disjoint sets and $\alpha_s$ is a function, such that

- $\alpha_s : A_s \to \sum_{\mathsf{src}\, e = s} A_{\mathsf{tgt}\, e}$, if $s$ is existential,

- $\alpha_s : A_s \to \prod_{\mathsf{src}\, e = s} A_{\mathsf{tgt}\, e}$, if $s$ is universal.

A morphism of $F$-coalgebras is a collection of set maps $h_s$ that commute with the $\alpha_s$ in the usual way.

Coalgebras are equivalent to *realizations*. An *$F$-realization* is a directed multigraph $G$ along with a multigraph homomorphism $\ell : G \to F$, called a *typing*, with the following properties.

- If $\ell(u)$ is an existential node, then $u$ has exactly one successor.
- If $\ell(u)$ is a universal node, then $\ell$ is a bijection between the edges of $G$ with source $u$ and the edges of $F$ with source $\ell(u)$.

A homomorphism of $F$-realizations is a multigraph homomorphism that commutes with typings.

**Theorem 2.1** *The categories of $F$-realizations and $F$-coalgebras are isomorphic.*

*Proof.* We first construct a coalgebra from a given realization $G = (U, D)$ and $\ell : G \to F$. For each node $s$ of $F$, let

$$A_s = \ell^{-1}(s) = \{u \in U \mid \ell(u) = s\}$$

and define $\alpha_s$ as follows:

- If $s$ is existential and $u \in A_s$, let $d : u \to v$ be the unique edge with $\mathsf{src}\, d = u$ in $G$. Define $\alpha_s(u) = \mathsf{in}_{\ell(d)}(v)$, where $\mathsf{in}_{\ell(d)} : A_{\ell(v)} \to \sum_{\mathsf{src}\, e = s} A_{\mathsf{tgt}\, e}$ is the natural injection into the coproduct.
- If $s$ is universal and $u \in A_s$, define $\alpha_s(u)$ to be the unique tuple $t \in \prod_{\mathsf{src}\, e = s} A_{\mathsf{tgt}\, e}$ such that $\pi_{\ell(e)}(t) = \mathsf{tgt}\, e$ for all $e$ such that $\mathsf{src}\, e = u$, where $\pi_{\ell(e)} : \prod_{\mathsf{src}\, e = s} A_{\mathsf{tgt}\, e} \to A_{\ell(v)}$ is the natural projection from the product.

Conversely, given a coalgebra with data $A_s, \alpha_s$ for nodes $s$ of $F$, we can define a realization. Let $U$ be the disjoint union of the $A_s$ and let $\ell(u) = s$ for $u \in A_s$. If $u \in A_s$ and $s$ is existential, then $\alpha_s(u) = \mathsf{in}_e(v) \in \sum_{\mathsf{src}\, e = s} A_{\mathsf{tgt}\, e}$ for some $e : s \to t$ and $v \in A_t$. Add an edge $d : u \to v$ and set $\ell(d) = e$. If $u \in A_s$ and $s$ is universal, then $\alpha_s(u) \in \prod_{\mathsf{src}\, e = s} A_{\mathsf{tgt}\, e}$. For each $e : s \to t$, let $v_e = \pi_e(\alpha_s(u)) \in A_t$. Add an edge $d_e : u \to v_e$ and set $\ell(d_e) = e$. □

## 2.2 Final Coalgebras

Realizations allow us to give a concrete construction of final coalgebras that is reminiscent of the Brzozowski derivative on sets of strings (see [11]). Here, instead of strings, the derivative acts on certain sets of paths of $F$.

Let $F$ be a type signature. Construct a realization $R_F, \ell_F$ as follows. A node of $R_F$ is a set $A$ of paths in $F$ such that

(i) $A$ is nonempty and prefix-closed;

(ii) all paths in $A$ have the same first node, which we define to be $\ell_F(A)$;

(iii) if $p$ is a path in $A$ of length $n$ and $\mathsf{tgt}\, p$ is existential, then there is exactly one path of length $n+1$ in $A$ extending $p$;

(iv) if $p$ is a path in $A$ of length $n$ and $\mathsf{tgt}\, p$ is universal, then all paths of length $n+1$ extending $p$ are in $A$.

The edges of $R_F$ are defined as follows. For every node $A$ and edge $e$ of $F$ such that there exists a path in $A$ whose first edge is $e$, include exactly one edge

$$d_{A,e} : A \to \{p \mid s\, e\, p \in A\}$$

in $R_F$, where $s = \mathsf{src}\, e = \ell(A)$, and $\ell_F(d_{A,e}) = e$. It is readily verified that $\mathsf{tgt}\, d_{A,e} = \{p \mid s\, e\, p \in A\}$ satisfies properties (i)–(iv) and that $\ell(\mathsf{tgt}\, d_{A,e}) = \mathsf{tgt}\, e$, so $\ell$ is a typing.

**Theorem 2.2** *The realization $R_F, \ell_F$ is final in the category of $F$-realizations. The corresponding coalgebra as constructed in Theorem 2.1 is final in the category of $F$-coalgebras.*

*Proof.* Let $G, \ell$ be an arbitrary realization. The unique homomorphism $h : G, \ell \to R_F, \ell_F$ is given by: $h(s)$ is the set of paths in $F$ that are images under $\ell$ of paths in $G$ starting with node $s$. The second statement of the theorem is immediate from Theorem 2.1. □

## Acknowledgments

Thanks to Jean-Baptiste Jeannin, Bobby Kleinberg, Alexandra Silva, and Navin Sivakumar for insightful comments.

## References

[1] P. Aczel and P. Mendler, "A final coalgebra theorem," in *Category Theory and Computer Science*, ser. Lecture Notes in Computer Science. Springer, 1989, vol. 389, pp. 357–365.

[2] J. Adámek, "On final coalgebras of continuous functors," *Theor. Comput. Sci.*, vol. 294, pp. 3–29, 2003.

[3] J. Adámek and V. Koubek, "On the greatest fixed point of a set functor," *Theor. Comput. Sci.*, vol. 150, no. 1, pp. 57–75, 1995.

[4] P. America and J. Rutten, "Solving reflexive domain equations in a category of complete metric spaces," *J. Comput. Syst. Sci.*, vol. 39, no. 3, pp. 343–375, 1989.

[5] M. Barr, "Terminal coalgebras in well-founded set theory," *Theor. Comput. Sci.*, vol. 114, no. 2, pp. 299–315, 1993.

[6] D. Hausmann, "Data types and computability via final coalgebras," Ph.D. dissertation, Dresden University, 2004.

[7] J. Rutten, "Universal coalgebra: a theory of systems," *Theor. Comput. Sci.*, vol. 249, pp. 3–80, 2000.

[8] M. B. Smyth and G. D. Plotkin, "The category-theoretic solution of recursive domain equations," *SIAM J. Comput.*, vol. 11, no. 4, pp. 761–783, 1982.

[9] J. Worrell, "Coinduction for recursive data types: partial orders, metric spaces and $\Omega$-categories." *Electr. Notes in Theor. Comput. Sci.*, vol. 33, pp. 337–356, 2000.

[10] ——, "On the final sequence of a finitary set functor," *Theor. Comput. Sci.*, vol. 338, no. 1–3, pp. 184–199, 2005.

[11] A. Silva, "Kleene coalgebra," Ph.D. dissertation, University of Nijmegen, 2010.

[12] J.-B. Jeannin and D. Kozen, "Computing with capsules," Computing and Information Science, Cornell University, Tech. Rep. http://hdl.handle.net/1813/22082, January 2011.