Into C—¿ Onto —-¿¿

Recursion and Corecursion Have the Same Equational Logic

Lawrence S. Moss
Indiana University
lsm@cs.indiana.edu

March 30, 2000*

Abstract

This paper is concerned with the equational logic of corecursion, that is of definitions involving final coalgebra maps. The framework for our study is iteration theories (cf. e.g. Bloom and Ésik [**?**, **?**]), recently re-introduced as models of the FLR_0 fragment of the Formal Language of Recursion [**?**, **?**, **?**]. We present a new class of iteration theories derived from final coalgebras. This allows us to reason with a number of types of fixed-point equations which heretofore seemed to require to metric or order-theoretic ideas. All of the work can be done using finality properties and equational reasoning.

Having a semantics, we obtain the following completeness result: the equations involving fixed-point terms which are valid for final coalgebra interpretations are exactly those valid in a number of contexts pertaining to recursion. For example, they coincide with the equations valid for least-fixed point recursion on dcpo's. We also present a new version of the proof of the well-known completeness result for iteration theories (see Ésik [**?**] and Hurkens et al [**?**]). Our work brings out a connection between coalgebraic reasoning and recursion.

1 Introduction

This paper studies fixed-point definitions pertaining to final coalgebras. To get quickly to the issues, here is an example of the kind of phenomenon which interests us. Consider infinite binary trees labeled with letters of the Roman alphabet. We can define trees by fixed-point terms, e.g.,

$$x \texttt{ where } \{x = \langle \texttt{g}, x, y\rangle, y = \langle \texttt{h}, y, y\rangle\} \tag{1}$$

This term is intended to denote an infinite binary tree x labeled by g whose left subtree is the same tree x, and whose right subtree is labeled by h throughout. We also allow the "where" construct to appear inside terms, as in

$$x \texttt{ where } \{x = \langle \texttt{g}, x, y\rangle, y = \langle \texttt{h}, y,\ z \texttt{ where } \{z = \langle \texttt{i}, z, z\rangle\}\ \rangle\} \tag{2}$$

And once we have a collection of terms like this, it becomes interesting to ask for methods to tell when terms denote the same tree. Intuitively, (**??**) and (**??**) do not denote the same tree,

*This paper was originally presented to the Mathematical Foundations of Programming Semantics meeting in 1999, and an earlier version was published in the proceeding, ENTCS vol. 20. This version contains a number of corrections and additions.

while (**??**) and (**??**) below do:

$$x\ \texttt{where}\ \{x = \langle \mathtt{g}, x, w\rangle, w = \langle \mathtt{h}, w,\ z\ \texttt{where}\ \{z = \langle \mathtt{h}, z, z\rangle\}\ \rangle\} \tag{3}$$

Going further, it seems natural to consider *parametric trees*, defined by terms containing free variables, as in

$$x\ \texttt{where}\ \{x = \langle \mathtt{g}, x, y\rangle\} \tag{4}$$

$$x\ \texttt{where}\ \{x = \langle \mathtt{g}, \langle \mathtt{g}, x, y\rangle, y\rangle\} \tag{5}$$

Such terms are "parametric" in the sense that given a concrete tree t, one should be able to substitute t for y in (**??**) and (**??**). Substitution of the same t into both (**??**) and (**??**) gives terms with the same denotation. So it seems to make sense that (**??**) and (**??**) should themselves have the same denotation.

Of course, at this point we really need a semantics of terms involving the "where" construction. (We also need a more exact syntax, of course.) The usual semantics for fixed-point terms of this sort uses the set of partial trees, and then the semantics itself amounts to solving a recursion equation on this larger set. In this paper, we depart from such approaches and give a semantics of parametric objects using only the notion of a final coalgebra. In this case, the infinite trees over any set S are a final coalgebra for the functor $F : Set \rightarrow Set$ given by $F(a) = S \times a \times a$. ($S$ is $\{\mathsf{g}, \mathsf{h}, \mathsf{i}\}$, say.) F acts on functions in the usual way. Actually our work generalizes to many other functors.

Contributions of this paper This paper presents a semantics for fixed-point terms using final coalgebras. This semantics is applicable in any setting where final coalgebras arise. (There are many such settings; cf. e.g. Rutten [**?**] for a survey.) Some of these settings can also be treated by the more usual method of passing to a larger set which has more limits of some kind. However, there is no known reduction of our work to the usual setting. Technically, our work involves the construction of an iteration theories for a functor F with a final coalgebra. (Actually we need F to satisfy some other properties; these are spelled out in the definition of a parametric corecursion system in Section **??**.) Our construction thus gives a new class of iteration theories.

Once we see how to obtain iteration theories from suitable functors, the next step is to examine the resulting equational logic. This connects our work to the literature on iteration theories; cf. e.g. Bloom and Ésik[**?**, **?**]. Let $A = B$ be a single equation between fixed-point terms. We show that $A = B$ is valid for coalgebra interpretations iff it is valid for all interpretations. It is known that this notion of validity coincides with validity on many other classes; for example the class of all dcpo interpretations.

Contents Section **??** presents the syntax and semantics of FLR_0. The final coalgebra semantics that we work with is deferred, however, until we present some basic results on parametric corecursion systems (Section **??**). We get an easy soundness result for FLR_0 interpreted on final coalgebra interpretations in Section **??**, and the Completeness Theorem for such equations is also presented there. We conclude with some remarks on the significance of these results, the relation to other work, and with open problems.

Background on iteration theories can be found in Bloom and Ésik [**?**] or [**?**]; background on the Formal Language of Recursion can be found in Hurkens et al [**?**]. Sections **??** and **??** quote several results from [**?**]. We do not make use of anything further from that paper.

2 Syntax and Semantics of FLR_0

This section recalls definitions from [?]. A *N-ranked set* is a set S togther with $r : S \to N$, where N is the set of natural numbers. We write S_n for $r^{-1}(n) = \{s \in S : r(s) = n\}$. Another name for an N-ranked set is a *signature*, and here we use the notation of Σ for the set and *arity* for the map. We think of Σ as a set of *function symbols* in this case. Consider such a signature Σ, and also consider and a countably infinite set of *variables* $\{\mathsf{v}_1, \mathsf{v}_2, \ldots\}$. We use letters like x, y and z to denote these variables.

The following inductive definition specifies the *terms* E of the language $FLR_0 = FLR_0(\Sigma)$.

$$E := \mathsf{v}_i \mid f(E_1, \ldots, E_n) \mid E_0 \ \texttt{where} \ \{x_1 = E_1, \ldots, x_n = E_n\}\,.$$

Here $f \in \Sigma_n$, and $E_1, \ldots, E_n$ are again terms. Intuitively, the second clause corresponds to function application, and the third clause gives syntax for the solution of systems of recursive equations. In the third clause, the variables x_1, ..., x_n must be distinct.

Syntactic notions concerning FLR_0 are defined as usual, including *free* and *bound* variable occurrences (`where` binds $x_1, \ldots, x_n$ in the third clause), *closed* and *open* terms, and *fresh* variables. We'll denote the set of free variables occurring in a term E by $\mathsf{fv}(E)$. If X is a set of variables and $\sigma\colon X \to FLR_0$, and A is a term such that $\mathsf{fv}(A) \subseteq X$, then we can perform the usual syntactic substitution of $\sigma(x)$ for each free occurrence of $x \in X$ in A. We write the result as $[\sigma]A$. If a term has been written $A(\vec{x})$ displaying (some of) its free variables, we will generally use the common notation $A(\vec{E})$ for $[\sigma]A$ where $\sigma(x_i) = E_i$ for each x in the sequence $\vec{x}$.

2.1 Semantics

Definition An $FLR_0(\Sigma)$ *structure* is a pair $R = (\Phi, \Lambda)$ where Φ is an N-ranked set called the *universe* of the structure, and Λ is a *denotation map* on $FLR_0(\Sigma \cup \Phi)$; i.e., for any term $E \in FLR_0(\Sigma \cup \Phi)$ and any non-repeating sequence $\vec{x} = x_1, \ldots, x_n$ of variables containing all of the free variables of E, $\Lambda(\vec{x})E \in \Phi_n$.

The denotation map Λ must be *compostional* in the following sense: for any term E and free substitutions σ and τ defined on $\vec{x} = x_1, \ldots, x_n$:

$$\Lambda(x_1, \ldots, x_n) f(x_1, \ldots, x_n) = f \tag{6}$$

$$\Lambda(\vec{y})\Big(\Lambda(\vec{x})E\Big)(\sigma(x_1), \ldots, \sigma(x_n)) = \Lambda(\vec{y})[\sigma]E \tag{7}$$

$$\text{If } \Lambda(\vec{y})\sigma(x_i) = \Lambda(\vec{z})\tau(x_i) \text{ for all } i, \text{ then } \Lambda(\vec{y})[\sigma]E = \Lambda(\vec{z})[\tau]E \tag{8}$$

The first of these properties says that every element of the universe acts as a symbol for itself. The idea behind the other two compositionality conditions is that the denotation of a complex term must depend only on the denotations of its subterms.

The conditions in (**??**)–(**??**) imply a number of standard consequences concerning renaming variables; in particular, if $\vec{x}$ includes all of the free variables of $A(\vec{x})$, then $\Lambda(\vec{x})A(\vec{x}) = \Lambda(\vec{y})A(\vec{y})$.

2.2 Normal FLR_0 structures

The canonical examples of FLR_0 structures are directed-complete partial orders (*dcpo's*) in which the function symbols are interpreted by monotone operations and in which the `where` operation is interpreted as least fixed point. In this paper, we are interested in structures which do not interpret `where` in this way but rather use finality. However, since the dcpo interpretations are much better-studied, we shall use some concepts from their study.

Every FLR_0 structure R gives rise to a notion of semantic consequence $\models_R$. A structure $R = (\Phi, \Lambda)$ *satisfies* an equation $A = B$ if for any list of variables $\vec{x}$ including $Fr(A) \cup Fr(B)$, $\Lambda(\vec{x})A = \Lambda(\vec{x})B$. R satisfies a set Γ of equations if it satisfies each $\gamma \in \Gamma$.

More generally, let Γ be a set of formulas and φ be any formula. Write $[\sigma]\Gamma$ for the application of the term substitution σ to every formula in Γ. Then φ is a consequence of Γ over R, written $\Gamma \models_R \varphi$, when the following condition holds: Whenever R satisfies $[\sigma]\Gamma$ for a term substitution σ, then R satisfies $[\sigma]\varphi$ as well. (If $\Gamma = \emptyset$, then the notion $\Gamma \models_R \varphi$ reduced to that of $\models_R \varphi$ from above.)

It can be shown that if R is a dcpo FLR_0 structure, all instances of the following three classes of identities hold, and all instances of the rule below also hold:

The Fixed Point Identity $\models_R A \texttt{ where } \{x = A\} \;=\; x \texttt{ where } \{x = A\}$.
The Head Identity Let A and $B_1, \ldots, B_n$ be terms. Define the term substitution σ by $\sigma(x_i) = x_i \texttt{ where } \{x_1 = B_1, \ldots, x_n = B_n\}$. Then

$$\models_R A \texttt{ where } \{x_1 = B_1, \ldots, x_n = B_n\} \quad = \quad [\sigma]A.$$

The Bekič-Scott Identity

$$\begin{aligned} \models_R \quad & A \texttt{ where } \{\vec{y} = \vec{C}, \vec{x} = \vec{B}\} \\ = \quad & (A \texttt{ where } \{\vec{y} = \vec{C}\}) \texttt{ where } \{x_1 = B_1 \texttt{ where } \{\vec{y} = \vec{C}\}, \\ & \qquad \ldots, x_n = B_n \texttt{ where } \{\vec{y} = \vec{C}\}\ \} \end{aligned}$$

The Recursion Inference Rule Consider two `where`-terms, say

$$A_0 \texttt{ where } \{x_1 = A_1, \ldots, x_n = A_n\} \quad \text{and} \quad B_0 \texttt{ where } \{y_1 = B_1, \ldots, y_m = B_m\},$$

with no bound variables in common. Let Δ be any set of equations of the form $x_i = y_j$. Then

$$\frac{\Gamma, \Delta \models_R A_0 = B_0 \qquad\qquad \Gamma, \Delta \models_R A_i = B_j \text{ for } x_i = y_j \in \Delta}{\Gamma \models_R A_0 \texttt{ where } \{\vec{x} = \vec{A}\} = B_0 \texttt{ where } \{\vec{y} = \vec{B}\}}.$$

That is, if R satisfies all of the assertions above the line, then it also satisfies the bottom assertion.

Definition An FLR_0 structure R is *normal* if R satisfies all instances of the Fixed Point, Head, and Bekič-Scott Identities, and if the Recursion Inference rule is sound for the consequence relation $\Gamma \models_R A = B$.

3 Background on Parametric Corecursion Systems, Substitution, and Corecursion

3.1 Parametric Corecursion Systems

Let C be a category with a fixed coproduct operation $+$. If a and b are objects, then we have injections $\mathsf{inl} : a \to a + b$ and $\mathsf{inr} : b \to a + b$. When we use subscripts on these injections, we have in mind a special meaning that we introduce below. If the context forces a unique reading, then we prefer not to subscript these injections. If $f : a \to c$ and $g : b \to c$, then we have a unique $\langle f, g\rangle : a + b \to c$ such that $f = \langle f, g\rangle \circ \mathsf{inl}$ and $g = \langle f, g\rangle \circ \mathsf{inr}$. If $f : a \to b$ and $g : c \to d$, then we also have $f + g : a + c \to b + d$ given by $\langle \mathsf{inl} \circ f, \mathsf{inr} \circ g\rangle$.

Let $F : C \to C$ be an endofunctor. For all objects c, F_c denotes the functor $d \mapsto F(c + d)$. If $g : d \to d'$, then $F_c g = F(\mathsf{id}_c + g)$.

Definition Let C be a category with a specified coproduct operation $+$. A *parametric F-corecursion system on C* is an assignment φ taking each object c of C to a final F_c-coalgebra $\varphi_c : \overline{a} \to F(c + \overline{c})$.

This definition is intended to be quite weak. The point is that on the basis of it, one can get further structure which, as we shall see, includes that of an iteration theory. See [**?**] for a development of the theory of parametric corecursion systems and an application in set theory.

Our main class of examples is the following: We take C to be the category *Set* of sets and functions[1]. We take $+$ to be the operation disjoint unions of sets (or any other fixed coproduct operation), *except* that to make our life easier at one point, we make the special assumption that if n and m are natural numbers (von Neumann ordinals) then the coproduct of n and m is the natural number $n + m$. (That is, we do not take the disjoint union in this case). This is a technical point that is not really needed, but as we say it simplifies a few details. Further, we take F be be any endofunctor on sets which is *uniform* in an appropriate sense (see [**?**]). It can be shown that each derived functor F_c has a final coalgebra. So specifying one for each c gives a parametric corecursion system.

Proposition 3.1 *Let F be a uniform functor on sets. Then for all c, F_c has a final coalgebra. Thus, there is a parametric F-corecursion system on C.*

In addition, if one assumes the Antifoundation Axiom (AFA), then the *greatest fixed point* F^* of F gives a final coalgebra $\langle F^*, \mathsf{id}\rangle$. Working with the greatest fixed points makes some of the work easier, but it is not strictly necessary and so we shall not do it in this paper. For more on these matters, see Aczel [**?**], Moss [**?**], or Turi and Rutten [**?**].

We shall be interested especially in functors on sets derived from signatures. Fix some signature Σ, and also fix a constant $\perp$ of Σ. Σ determines a functor $F = F_\Sigma$ on C in the usual way, by setting

$$F(a) = \{\langle f, b_1, \ldots, b_n\rangle : f \in \Sigma, \mathit{arity}(f) = n, \text{ and } b_1, \ldots, b_n \in a\},$$

and for $k : a \to b$, $Fk(\langle f, b_1, \ldots, b_n\rangle) = \langle f, kb_1, \ldots, kb_n\rangle$. A function symbol c of arity 0 is a constant symbol. For all a, $\langle c\rangle \in F(a)$.

[1] It is also possible to be slightly more general by taking C to be the category of classes and definable, set-continuous, operations.

This functor is uniform, and so it follows from Proposition **??** that is gives a parametric corecursion on sets. (The existence of final coalgebras in this case can be deduced not only from results in [**?**] but also from the work of Aczel and Mendler [**?**], Barr [**?**], and Turi and Rutten [**?**].)

We should stress that the final coalgebras may be constructed by general set-theoretic arguments that do not amount to constructing explicit representations. In other words, the existence of final coalgebras should be thought of as the fundamental fact, and the particular representation is secondary.

The intuition behind the sets $\overline{n}$ Recall that (the standard model in set theory of) the natural number n is $\{0, 1, \dots, n-1\}$. As a particular consequence of Proposition **??**, for each n, the functor F_n has a final coalgebra. We fix such a final coalgebra $\varphi_n : \overline{n} \to F(n+\overline{n})$. When F is a signature functor, $\overline{n}$ can be taken to be the set of infinite, ordered trees t such that some of the nodes of t are labeled by Σ and the rest of the nodes are labeled with the numbers 0, …, $n-1$. Moreover, the following requirements holds for t:

1. If a node x of t is labeled with one of the numbers 0, …, $n-1$, then x has no children in t; and
2. if x is labeled with f and $arity(f) = k$, then x has k children; and
3. the root of t is labeled with some function symbol f from Σ.

Our choice of the numbers $0, \dots, n-1$ is simply for notational convenience. One could instead take them to be any sets x_0, …, x_{n-1}. Usually one would take them to be the "variables" x_1, …, x_n. In this way, $\overline{n}$ corresponds to the infinite trees from Σ in these variables. We might add that we have not added the variables to the signature, so that the one-point trees labeled by numbers are not in our set. (This is what requirement (3) says.) But when we want the bigger set, it would be $n + \overline{n}$.

Some further remarks: What we have is actually a map $\varphi_n : \overline{n} \to F(n+\overline{n})$. Our description above assumes that φ_n is the identity (i.e., we suppressed mention of it), and this is certainly the best way to read what we do. In addition, the formalism actually says that every element $t \in \overline{n}$ is of the form $\langle f, z_1, \dots, z_k \rangle$, where $k = arity(f)$, and where each z_i is either a natural number below k or else another t' with the same features as in this definition. The use of the coproduct is to make the two forms distinguishable. What we have done above with trees is to just give the more classical rendering of the same structure.

As an example of how our notation works in this case, consider the term $f(x_1, x_2)$ as an element of $\overline{n}$, where $n > 2$ is arbitrary. For $i < n$, x_i corresponds to $\mathsf{inl}_n(i-1)$, and so term $f(x_1, x_2)$ corresponds to

$$\varphi_n^{-1}(\langle f, \mathsf{inl}_n(0), \mathsf{inl}_n(1) \rangle).$$

A complex term like $g(x_3, f(x_1, x_2))$ then corresponds to

$$\varphi_n^{-1}(\langle g, \mathsf{inl}_n(2), \mathsf{inr}_n(\varphi_n^{-1}(\langle f, \mathsf{inl}_n(0), \mathsf{inl}_n(1) \rangle)) \rangle).$$

We know that our formalism will be unfamiliar to nearly all readers, but it has its advantages for the work we do beginning in the next section. We hope that the remarks in this section will allow a smooth translation to more standard notations.

3.2 Substitution and Corecursion

As it usually appears, *substitution* is an easy consequence of initiality or recursion. Here is the kind of formulation we have in mind. The set T_Σ of Σ-terms is an initial algebra of the functor $F = F_\Sigma$. Moreover, for each set X, we can consider the derived functor F_X and its initial algebra $T_\Sigma(X)$. More precisely, let the initial algebra maps for T_Σ and $T_\Sigma(X)$ be ϵ and ϵ_X, respectively. Now the initiality gives us the following principle: for every map $f : X \to T_\Sigma$ there is a unique $[f] : T_\Sigma(X) \to T_\Sigma$ with the property that $\epsilon \circ [f] = F\langle g, [f]\rangle \circ \epsilon_X$.

In contrast to all of this, we need a formulation of substitution based on finality. The basic idea is the same as the one mentioned above, except that in contrast to the situation above, we cannot define the function we need by *recursion*. Instead, we appeal to finality. We say in this case that the substitution operation is be defined by *corecursion*.
Here is our substitution principle:

Lemma 3.2 (Substitution [?]) *Let $f : a \to b + \overline{b}$. Then there is a unique $[f] : \overline{a} \to \overline{b}$ so that*

$$F\langle f, \mathsf{inr}_b \circ [f]\rangle \circ \varphi_a \quad = \quad \varphi_b \circ [f]. \tag{9}$$

[tight,size=3.5em] $\overline{a}^{\varphi_a}$ $F(a + \overline{a})$
$[f]$
$F\langle f, \mathsf{inr}_b \circ [f]\rangle$
$\overline{b}_{\varphi_b} F(b + \overline{b})$

We also need the notion of a *solution to a system of parametric equations.* This is supplied by the following result.

Theorem 3.3 (Parametric Corecursion [?]) *Let $f : a \to \overline{a + b}$. Then there is a unique $f^\dagger : a \to \overline{b}$ so that $f^\dagger = [\langle \mathsf{inr}_b \circ f^\dagger, \mathsf{inl}_b\rangle] \circ f$: [tight,size=3em]* a ${}^f\overline{a+b}$
$f^\dagger$ $[\langle \mathsf{inr}_b \circ f^\dagger, \mathsf{inl}_b\rangle]$
$\overline{b}$

3.3 Additional Structure in Parametric Corecursion Systems

Theorem **??** plays the role of fixed-point principles in other approaches. That is, our semantics of fixed-point terms is based heavily on this result. We also need some properties of substitution which we mention at this point. As with Theorem **??** and Lemma **??**, the results just below hold in any parametric corecursion system.

Lemma 3.4 *Consider $\mathsf{inl}_a : a \to a + \overline{a}$. Then $[\mathsf{inl}_a] = \mathsf{id}_{\overline{a}}$.*

Lemma 3.5 *Let $f : a \to b + \overline{b}$ and $g : b \to c + \overline{c}$. Then $[g] \circ [f] = [\langle g, \mathsf{inr}_c \circ [g]\rangle \circ f]$.*

To prove Lemma **??**, we only need to check that $\mathsf{id}_{\overline{a}}$ works for $[\mathsf{inl}_a]$ in (**??**). This is an easy consequence of functoriality. For Lemma **??**, we check that $[g] \circ [f]$ has the defining property of $[\langle g, \mathsf{inr}_c \circ [g]\rangle \circ f]$.

We also have additional structure $\langle M, \mathsf{unit}, -^\star\rangle$ given as follows: M is the operation taking the object a to described as a Kleisli triple as follows: $Ma = a + \overline{a}$, unit takes a to the morphism $\mathsf{inl}_a : a \to Ma$, and for each $f : a \to Mb$, $f^\star : Ma \to Mb$ is $\langle f, \mathsf{inr}_b \circ [f]\rangle$.

Lemma 3.6 *$\langle M, \mathsf{unit}, -^\star\rangle$ is a Kleisli triple. That is, $\mathsf{unit}_a^\star = \mathsf{id}_{Ma}$, $f^\star \circ \mathsf{unit}_a = f$, and $f^\star \circ g^\star = (f^\star \circ g)^\star$.*

Proof The first point is by Lemma **??**, the second is by the definition of $f^\star$, and the last is a routine calculation using Lemma **??**. $\dashv$

4 Iteration Theories and FLR_0-structures Derived From Parametric Corecursion Systems

We begin with four data:

1. A category $(C, +)$, a category with a specified coproduct $+$ and containing $(\mathbf{Tot}, +)$ as a substructure (i.e., a subcategory, and the coproduct on **Tot** is the restriction of the one on C).
2. A parametric F-corecursion system φ on C.
3. A signature Σ containing a constant symbol $\perp$.
4. For each $f \in \Sigma_n$, some C-morphism $f_\Lambda : 1 \to n + \overline{n}$. We call the map $f \mapsto f_\Lambda$ a *pre-denotation*, since at some point later we use it to define a denotation map.

Recall that **Tot** is the algebraic theory with morphisms i_n for $1 \le i \le n$. These make each n the n-fold coproduct of 1 in **Tot**. We assume that the coproduct in C works as in **Tot** for natural numbers. (For sets, this is related to the technical assumption mentioned above.)

From the data we will construct an iterative theory $T = T(\varphi)$. By (3) and (4) we have a map $\perp_\Lambda : 1 \to 0 + \overline{0}$. We use $\perp_\Lambda$ to extend the dagger operation of T, and obtain an iteration theory $S = S(\varphi, \perp_\Lambda)$. We use the rest of the maps f_Λ to define an $FLR_0(\Sigma)$-structure $R = R(\varphi, \perp)$.

For the definitions and basic properties of algebraic theories, iterative theories, and iteration theories, see Bloom and Ésik [**?**]. We follow this book in writing $f \cdot g$ for $g \circ f$.

4.1 An Iterative Theory $T(\varphi)$

Lemma **??** shows that the parametric corecursion system φ gives rise to a Kleisli triple which we denoted by $\langle M, \mathsf{unit}, -^\star\rangle$. Let K be the associated Kleisli category for this triple. So K has the same objects as C, and $f : a \to b$ in K iff $f : a \to Mb$ in C. Moreover, $g \cdot f = f \circ g$ in K is $g \cdot f^\star = f^\star \circ g$ in C, and also id_a in K is unit_a. (The import of Lemma **??** is that it implies the category properties for K.)

We next define a theory $T = T(\varphi)$. As a category, T will be a subcategory of K. We put a morphism $f : n \to m$ into T if f is a morphism of K, and if each $i_n \cdot f$ is of exactly one of following two forms in C:

1. $i_n \cdot \mathsf{inl}_n$ for some i_n of **Tot**
2. $g \cdot \mathsf{inr}_m$, for some $g : 1 \to \overline{m}$ of C

(In sets, since the coproduct + gives a *disjoint* union, the "exactly" is of course redundant.)

This specifies T as a category. Let i_n in T be $i_n \cdot \mathsf{inl}_m$ of C. It is easy to check that T is closed under the source tupling operations of C, and we use these operations to define the source tupling of T.

We next check that in T, if $g = \langle f_1, \ldots, f_n \rangle$ and $i \leq n$, then $i_n \cdot g = f_i$. For this, we translate $i_n \cdot g$ to C and calculate there:

$$i_n \cdot \mathsf{inl}_n \cdot g^\star \quad = \quad i_n \cdot \mathsf{unit}_n \cdot g^\star \quad = \quad i_n \cdot g \quad = \quad f_i.$$

Therefore, we have an algebraic theory T at this point. (We might note as well that $1_1 = \mathsf{id}_1$: both are given explicitly as $\mathsf{inl}_1 : 1 \to 1 + \overline{1}$.)

It remains to define the dagger operation. Before we do this, we check that the collection I of nondistinguished scalar morphisms of T is a scalar ideal of T. Let $f \in I$. Then there is some $g : 1 \to \overline{n}$ in C such that $f = g \cdot \mathsf{inr}_n$. Let $h : n \to m$ in T; so $h : n \to m + \overline{m}$ in C. Then we calculate $f \cdot h$ in C: $f \cdot h^\star = g \cdot \mathsf{inr}_n \cdot h^\star = g \cdot [h] \cdot \mathsf{inr}_m$. Since $g \cdot [h] : 1 \to \overline{m}$ in C, we see that $f \cdot h$ belongs to I. This verifies that I is a scalar ideal.

Now we define the dagger operation. Suppose that $f : m \to m + p$ is an ideal morphism of T. Let $f_1 : m \to \overline{m + p}$ in C be such that $f = f_1 \cdot \mathsf{inr}_{m+p}$. By Theorem **??**, we have a unique morphism $f_2 : m \to \overline{p}$ so that $f_2 = f_1 \cdot [\langle f_2 \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle]$. We set $f^\dagger = f_2 \cdot \mathsf{inr}_p$.

Lemma 4.1 *$T(\varphi)$ is an ideal iterative theory.*

Proof Suppose that $f : m \to m + p$ is an ideal morphism of T. We showed how to define $f^\dagger$ in terms of other morphisms, and we'll use the notation from above. We want to show that in T, $f^\dagger = f \cdot \langle f^\dagger, \mathsf{id}_p \rangle$. Now the coproduct $\langle f^\dagger, \mathsf{id}_p \rangle$ in T is actually $\langle f_2 \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle$ in C. We calculate in C:

$$\begin{array}{rcl} f \cdot \langle f_2 \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle^\star & = & f_1 \cdot \mathsf{inr}_{m+p} \cdot \langle \langle f_2 \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle, [\langle f_2 \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle] \cdot \mathsf{inr}_p \rangle \\ & = & f_1 \cdot [\langle f_2 \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle] \cdot \mathsf{inr}_p \\ & = & f_2 \cdot \mathsf{inr}_p \\ & = & f^\dagger. \end{array}$$

This shows that $f^\dagger$ satisfies the appropriate fixed-point equation. For the uniqueness of $f^\dagger$, suppose that $f^* : m \to p$ is such that in T, $f^* = f \cdot \langle f^*, \mathsf{id}_p \rangle$. Let $f_2' : m \to \overline{p}$ be such that $f^* = f_2' \cdot \mathsf{inr}_p$. Again, $\langle f^*, \mathsf{id}_p \rangle$ in T is $\langle f_2' \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle$ in C. And we have $f_2' \cdot \mathsf{inr}_p = f^* = f \cdot \langle f_2' \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle = f_1 \cdot [\langle f_2' \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle] \cdot \mathsf{inr}_p$. So $f_2' = f_1 \cdot [\langle f_2' \cdot \mathsf{inr}_p, \mathsf{inl}_p \rangle]$. Now our uniqueness assertion on f_2 implies that $f_2' = f_2$. Therefore $f^* = f^\dagger$, as required. This concludes the proof. $\dashv$

4.2 An Iteration Theory $S(\varphi, \perp_\Lambda)$

An *iteration theory* is an algebraic theory which is equipped with an operation $f \mapsto f^\dagger$ defined whenever $f : n \to n + p$. This dagger operation must be total, and a number of equational laws must also hold. Looking at FLR_0, the totality requirement corresponds to a choice of a canonical solution to equations like $x = x$. Up until now, we have not used any of the maps f_Λ from datum (4) at the beginning of this section. We use $\perp_\Lambda : 1 \to 00 + \overline{0}$ now. Notice that $\perp_\Lambda$ is also a morphism of T. As such, $\perp_\Lambda : 1 \to 0$.

Lemma 4.2 *Consider $T(\varphi)$ and $\perp_\Lambda : 1 \to 0$.*

1. *There is an extension of the dagger operation of $T(\varphi)$ to an operation defined for all morphisms $f : m \to m+n$, and such that $(\mathsf{id}_1)^\dagger = \perp_\Lambda$.*

2. *Let $S = S(\varphi, \perp)$ be the theory $T(\varphi)$, together with the extended dagger. Then S is an iteration theory, and it also satisfies the functorial dagger implication for all base morphisms (among other things).*

Proof This is just a special case of a result of Bloom, Elgot, and Wright [**?**]. This result also appears as Chapter 6, Theorem 4.5 of Bloom and Ésik [**?**]. $\dashv$

4.3 An $FLR_0(\Sigma)$ Structure $R(\varphi, \Lambda)$

In [**?**] we find a general way to take an iteration theory into and a pre-denotation and to obtain an $FLR_0(\Sigma)$-structure. We apply this to the iteration theory $S(\varphi, \perp_\Lambda)$ from Section **??**.

The universe Φ is given by $\Phi_n = S(1, n)$. Then the denotation map Λ of our FLR_0-structure is given as follows:

- $\Lambda(x_1, \ldots, x_n)x_i = i_n$.
- $\Lambda(x_1, \ldots, x_m)f(E_1, \ldots, E_n) = f_\Lambda \cdot \langle \Lambda(\vec{x})E_1, \ldots, \Lambda(\vec{x})E_n \rangle$.
- $\Lambda(x_1, \ldots, x_m)\, E_0$ `where` $\{y_1 = E_1, \ldots y_n = E_n\}$ is

$$\Lambda(\vec{y}, \vec{x})E_0 \cdot \langle \langle \Lambda(\vec{y}, \vec{x})E_1, \ldots, \Lambda(\vec{y}, \vec{x})E_n \rangle^\dagger, \mathsf{id}_m \rangle.$$

The second clause deals with f from the signature Σ, and again we use the pre-interpretation. In an FLR_0-structure, the elements of each Φ_n are also taken function symbols of arity n. We therefore extend the pre-interpretation by taking $f_\Lambda = f$ for $f \in \Phi_n$. In the last clause, we must make a provision here to cover the case when the sequence $\vec{y}, \vec{x}$ has repeated elements. We deal with this case in the following way: Let $w_1, \ldots, w_k$ be the subsequence of $\vec{x}$ containing the variables which occur in $\vec{y}$. Let $\vec{z}$ be the sequence $\vec{x}$ with w_i replaced by the ith variable (in the natural order) which is not among the x's or y's. Then $\vec{y}$ and $\vec{z}$ have no overlaps, and we set the value to be

$$\Lambda(\vec{y}, \vec{z})E_0 \cdot \langle \langle \Lambda(\vec{y}, \vec{z})E_1, \ldots, \Lambda(\vec{y}, \vec{z})E_n \rangle^\dagger, \mathsf{id}_m \rangle.$$

Theorem 4.3 *$R(\varphi, \Lambda)$ is a normal FLR_0 structure.*

Proof See Proposition **??** and the Appendix of [**?**]. The sketch of the proof in [**?**] unfortunately omits the soundness of the Recursion Inference Rule, and this takes an argument. One can be found in Moss and Whitney [**?**], based on the connection of the functorial dagger implication for all base morphisms and the Recursion Inference Rule. $\dashv$

Theorem **??** is the main result of this paper for FLR_0-structures. (For iteration theories, the parallel results of Lemmas **??** and **??** would be the main ones.) The point is that functors on sets which satisfy the mild condition of having final coalgebras give rise to interpretations of fixed-point terms in a way that satisfies all of the standard identities. That is, the logic of recursion is sound for all of these interpretations.

Incidentally, it is of course possible to define $R(\varphi, \Lambda)$ in a completely elementary way, without going via iteration theories and iterative theories. However, if did things this way, the verifications of all of the needed properties would be quite long. In effect, one would be reproving the results which we quoted, either in their original form or in special cases. This fact is what decided the presentation strategy for this paper.

4.4 Examples

We present several examples of final coalgebra interpretations.

Example 4.4 Perhaps the most natural example is that of a signature functor. This example also comes up in In Section **??**. Take C to be sets, $+$ to be disjoint union modified on the natural numbers, F to be the signature functor corresponding to a signature Σ containing a fixed constant $\perp$. For the pre-denotation, we set f_Λ to be the S-morphism $f_\Lambda : 1 \to n$ whose value is

$$\mathsf{inr}_n(\varphi_n^{-1}(\langle f, \mathsf{inl}_n(0), \ldots, \mathsf{inl}_n(n-1)\rangle)).$$

We shall call this structure $R(\Sigma)$.

Example 4.5 Another example might be the terms in the introduction to this paper. Here again we would take C and $+$ the same way. We would take F to be the functor defined on objects by $F(a) = S \times a \times a$, where S is a fixed set. On morphisms, F is as expected. This functor is uniform (as are practically all functors of interest except for the identity), and so we get a parametric corecursion system. We fix a signature Σ containing $\perp$. In the rest of this example, we will suppress all of the injections and isomorphisms for readability. We chose our pre-interpretation to be defined by $f_\Lambda = \langle \mathsf{g}, 0, 1\rangle$, $g_\Lambda = \langle \mathsf{h}, 1, 1\rangle$, etc. In this way, the trees which interpret terms as in (**??**)–(**??**) are naturally the denotations of terms from $FLR_0(\Sigma)$. Moreover, the assertions we made concerning equalities or inequalities of denotations can be checked formally. (That is, on the basis of our semantics one can check the assertions in the Introduction, or one can check our definitions against the intutions that we presented earlier.)

To summarize the results of this section, we have a general way of taking any parametric corecursion system and obtaining an iteration theory which satisfies some addtional properties. This iteration theory then translates to a normal FLR_0 structure.

5 Coalgebraic Proof of the Completeness of the Logic of Recursion

We recall the equational proof system $\vdash A = B$ for $FLR_0(\Sigma)$ from [**?**]. Its axioms are the identities $x = x$ together with the Fixed Point, Head, and Bekič-Scott Identities. Its rules of inference correspond to the symmetric and transitive properties of equality, to substitution using the function symbols of the underlying signature Σ, and finally to the Recursion Inference Rule.

Let FC be the class of $FLR_0(\Sigma)$-structures which are final coalgebra interpretations. That is, all of the structures defined from parametric corecursion systems and pre-denotations by the definitions of Section **??**. We write $\models_{\mathsf{FC}} A = B$ to mean that $A = B$ holds in all R from FC. Recall also that we have a the functor $F = F_\Sigma$ and also the final coalgebra interpretation $R(\Sigma)$ given in Example **??**. These are used throughout the rest of this proof.

Theorem 5.1 (Completeness) *Suppose that Σ contains a symbol other than $\perp$. $\models_{R(\Sigma)} A = B$ iff $\models_{\mathsf{FC}} A = B$ iff $\vdash A = B$. That is, every equation which holds in $R(\Sigma)$ is provable.*

Proof One can check that for the particular signature functor F_Σ on C, the final coalgebra interpretation $R(\Sigma)$ of Example **??** is isomorphic to the iteration theory $\Sigma\mathbf{tr}$. It is known that an equation holds in $\Sigma\mathbf{tr}$ iff it holds in all iteration theories. This result for iteration theories may be found in Ésik [**?**]. For the class of all FLR_0-structures, $\Sigma\mathbf{tr}$ is called $\mathrm{FLR}_0(\Sigma)$. Theorem 2 of Hurkens et al [**?**] sketches the proof of the completeness theorem in this setting. $\dashv$

Despite the fact that known results easily imply Theorem **??**, we give a different proof. We do this for three reasons: First, there is an intuition that the Recursion Inference Rule has *something* to do with bisimulation. For example, [**?**] contains the following remark on the decidability of the equational theory of iteration theories: "The whole process resembles the minimization of deterministic finite automata." Our proof makes the connection explicit: minimization of automata is also the quotient under the largest bisimulation. A second reason to present our proof is that we use *only* the final coalgebra structures; one never needs to explicitly construct or study Σ-trees. Finally, the fact that the proof is more abstract means that is that might generalize in ways that the older proof does not. (Of course, the abstraction also means that something might be lost. In this case, what is lost is the fact the polynomial-time algorithm for decidability. This extra information does not follow directly from our work below.)

An $FLR_0(\Sigma)$ term A is in *simplified form* if A is of the form

$$x_j \texttt{ where } \{x_1 = A_1, \ldots, x_n = A_n\}, \tag{10}$$

where $1 \leq j \leq n$ and where each A_i, is either x_i, or is of the form $f_i(z_1, \ldots, z_{m_i})$ where $f_i \in \Sigma$ has arity m_i and each z_j is one of the x_k. (When $m_i = 0$, A_i is a constant term.) A is in *tightly simplified form* if A is either a variable; or A is in simplified form, and the only equations in A between variables are of the form $x = x$ (that is, the same variable appears on both sides).

Lemma 5.2 (Simplification Lemma) *Let A be a term of $FLR_0(\Sigma)$.*

1. *There is a term $A' \in FLR_0(\Sigma)$ in simplified form such that $\vdash A = A'$.*

2. *There is a term $A'' \in FLR_0(\Sigma)$ in tightly simplified form such that $\vdash A = A'$.*

Proof The first part is from [**?**]. For the second part, we may assume that A already is a simplified form. If x and y are different and A is $x = x \texttt{ where } \{x = y\}$, then $\vdash A = y$. Also, $x = x \texttt{ where } \{x = x\}$ is tightly simplified. So we assume that A has at least two equations. We show how to take one equation of A between different variables, say $y = z$, and eliminate this equation. For example, suppose A is

$$x \texttt{ where } \{x = f(x,y), y = z, z = g(x,y,z)\}.$$

By the Bekič-Scott Identity,

$$\vdash A = (\, x \texttt{ where } \{x = f(x,y), z = g(x,y,z)\}\,) \texttt{ where } \{y = z\}.$$

By the Head Identity,

$$\vdash A = x \texttt{ where } \{x = f(x, \, y \texttt{ where } \{y = z\}\,), z = g(x, \, y \texttt{ where } \{y = z\}\,, z)\}.$$

Now as we know, $\vdash (\, y \texttt{ where } \{y = z\}\,) = z$. So with an an application of the Recursion Inference rule (or a substitution principle derived from it), we see that

$$\vdash A = x \texttt{ where } \{x = f(x,z), z = g(x,z,z)\}.$$

We thus eliminate one equation between non-identical variables. Doing this as many times as necessary, way obtain a tightly simplified form. $\dashv$

Proof The fact that each interpretation R in FC is a normal FLR_0 structure tells us that if $\vdash A = B$, then $\models_R A = B$. The crux of the matter is showing that if $\models_{R(\Sigma)} A = B$, then $\vdash A = B$.

By Lemma **??**, we assume that A and B are tightly simplified forms. Also, we may assume that neither A nor B is a variable, since in such cases the only way to have $\models_{R(\Sigma)} A = B$ would be to have A and B identical to the same variable. Here we use our assumption that Σ is not just $\perp$.

Let the bound variables of A be $X = \{x_1, \ldots, x_n\}$; let the bound variables of B be $Y = \{y_1, \ldots, y_m\}$; and let the variables with free occurrences in either term be $Z = \{z_1, \ldots, z_p\}$. We may assume that X, Y, and Z are pairwise disjoint. We show that $\vdash A = B$ using an application of the Recursion Inference Rule.

Let $e : X \to F_p(X)$ be defined by taking A_i and replacing the z variables by natural numbers and keeping the x's. For example, we might have $A_i = f(x_j, z_r)$. Then we would set

$$e(x_i) \quad = \quad \langle f, \mathsf{inr}(x_j), \mathsf{inl}(r-1)\rangle \in F(p + X).$$

The precise definition of e would involve specifying each A_i as an element of $F(Z + X)$. Let $s : Z \to p$ be a bijection. Then $e(x_i) = F\langle s, \mathsf{id}_X\rangle(A_i)$.

Recall that for each term A, $\Lambda(z_1, \ldots, z_p)A \in p + \overline{p}$. If A is tightly simplified but not a variable, then $\Lambda(\vec{z})A$ factors through inr_p. So we can define $u : X \to \overline{p}$ by

$$\mathsf{inr}_p u(x_i) \quad = \quad \Lambda(z_1, \ldots, z_p)\, x_i \texttt{ where } \{\vec{x} = \vec{A}\}.$$

We need to review how the semantics works in $R(\Sigma)$ (and hence in $T(\varphi)$ and $S(\varphi, \Lambda)$).

For $i = 0, \ldots, m$, let $a_i \in \Phi_{n+p}$ be $\Lambda(\vec{x}, \vec{z})A_i$. For $1 \leq i \leq m$, a_i is of the form $\mathsf{inr}(b_i)$ for some $b_i \in \overline{n+p}$. Let $k : n \to \overline{n+p}$ be given by $k(i) = b_i$. Let $k^{\dagger} : n \to \overline{p}$ be determined from

k by the Parametric Corecursion Theorem **??**. Let $l : n+p \to p+\overline{p}$ be $\langle \mathsf{inr}_p \circ k^\dagger, \mathsf{inl}_p \rangle$. Then $[l] : \overline{n+p} \to \overline{p}$. We take

$$\Lambda(\vec{z})\, x_i \texttt{ where } \{\vec{x} = \vec{A}\} \quad = \quad \langle l, \mathsf{inr}_n \circ [l] \rangle \mathsf{inl}_n(i-1) \quad = \quad \mathsf{inr}_p \circ k^\dagger(i-1).$$

We conclude that $u(x_i) = k^\dagger(i-1)$ for all i.

Claim $u : \langle X, e \rangle \to \langle \overline{p}, \varphi_p \rangle$ is a morphism of F_p-coalgebras.

Proof To keep the notation manageable, we'll again work with $A_i = f(x_j, z_r)$. We show that $(\varphi_p \circ u)x_i = (F_p u)e(x_i)$. We calculate, using the Fixed Point Identity, the semantics of $R(\Sigma)$, the defining equation for $[l]$ (see equation (**??**)), the definition of F_p, and the relations between u, $k^\dagger$, and l noted above:

$$\begin{array}{cl}
 & \Lambda(\vec{z})\, x_i \texttt{ where } \{\vec{x} = \vec{A}\} \\
= & \Lambda(\vec{z})\, f(x_j, z_r) \texttt{ where } \{\vec{x} = \vec{A}\} \\
= & (\langle l, \mathsf{inr}_p \circ [l] \rangle \circ \mathsf{inr}_{n+p} \circ \varphi_{n+p}^{-1}) \langle f, \mathsf{inl}_{n+p}(p+j-1), \mathsf{inl}_{n+p}(r-1) \rangle \\
= & (\mathsf{inr}_p \circ [l] \circ \varphi_{n+p}^{-1}) \langle f, \mathsf{inl}_{n+p}(p+j-1), \mathsf{inl}_{n+p}(r-1) \rangle \\
= & (\mathsf{inr}_p \circ \varphi_p^{-1} \circ F\langle l, \mathsf{inr}_p \circ [l] \rangle) \langle f, \mathsf{inl}_{n+p}(p+j-1), \mathsf{inl}_{n+p}(r-1) \rangle \\
= & (\mathsf{inr}_p \circ \varphi_p^{-1}) \langle f, k^\dagger(j-1), \mathsf{inl}_n(r-1) \rangle \\
= & (\mathsf{inr}_p \circ \varphi_p^{-1}) \langle f, u(x_j), \mathsf{inl}_n(r-1) \rangle
\end{array}$$

So $(\varphi_p \circ u)x_i = \langle f, u(x_j), \mathsf{inl}_n(r-1) \rangle = (F_p u \circ e)x_i$. ⊣

Of course, we can define functions e' and v on Y in a similar way to u, and again we shall have a morphism of F_p-coalgebras $v : \langle Y, e' \rangle \to \langle \overline{p}, \varphi_p \rangle$.

At this point, we need a number of well-known facts of coalgebra (see, e.g. Rutten [**?**]). First, all signature functors preserve weak pullbacks. Second, for all functors which preserve weak pullbacks, the pulback of a pair of final coalgebra morphisms is a bisimulation. Third, a bisimulation for a signature functor behaves exactly as expected: if two terms are related, then their head function symbols must be identical, and corresponding variables must also be related. These three facts hold for the signature functor F; related facts hold for F_p, mutatis mutandis.

Hence the pullback of u and v is an F_p-bisimulation. This is the set

$$\Delta \quad = \quad \{x_i = y_j : x_i \in A, y_j \in B, \text{ and } u(x_i) = v(y_j)\}.$$

We use the Recursion Inference Rule to show that $\vdash A = B$. By our hypothesis that $\models_{R(\Sigma)} A = B$, Δ contains the equation between the two head variables of A and B.

We only need to show that $\Delta \vdash A_i = B_j$ whenever $x_i = y_j \in \Delta$. Assume that A_i is of the form $f(w_1, \ldots, w_n)$. Then again by the bisimulation property, B_j must be a function application with the same function symbol f. Say B_j is $f(w'_1, \ldots, w'_n)$. By bisimulation, w_k is one of the z's iff w'_k is the same variable. And if w_k is one of the x's, then w'_k must be one of the y's and we have $u(w_i) = v(w'_i)$. In this latter case, the equation $w_i = w'_i$ belongs to Δ. And then by equational logic $\Delta \vdash A_i = B_j$. ⊣

The FLR_0/iteration theory proof system is complete in the same sense for numerous classes of interpretations, and we conclude that *recursion and corecursion have the same equational logic.*

6 Concluding Remarks

But is the semantics and the Completeness Theorem the same one we already know? In view of the running example of trees in this paper, is natural to ask whether we have actually obtained anything new. For that particular example, we get a new semantics for the parametric trees, but of course we do not get a better logic. Moreover, we do not get any new insights into the valid equations by any of the work of this paper.

Nevertheless, we do feel that giving the semantics in terms of parametric corecursion systems is appealing. Many of the fixed points that one finds in theoretical computer science have to do with final coalgebras (see Rutten [**?**], for example). Once one is familiar with the ideas, it seems natural to work as much as possible with notions of finality. The one up-front cost of such an approach is that one would need to have results giving final coalgebras. But for sets at least, there is a body of such results (see [**?**, **?**], for example).

But even more, the methods here would work *wherever* final coalgebras are used. This is the key contribution of this paper.

It is natural to ask whether *all* of the final coalgebra interpretations are in some sense *reducible* to other kinds of interpretations, especially those involving dcpo's. This question was asked in a more precise way in our paper [**?**]. We showed that for essentially all functors on sets which arise "in practice," the final coalgebra interpretations could indeed be obtained as the maximal elements of some dcpo. However, that work involved assuming hypotheses on the functors that go beyond what we needed here (but which nevertheless hold "in practice"). The work also used specific features of sets (the Replacement Axiom, for example). So at the present time, we have a reduction most of the time for functors on sets. For functors on other categories, I know of no general results which reduce corecursion to recursion.

What about the duals of these results? It is natural to ask about dual results to the ones of this paper. This is something that can be asked about many of the results of coalgebra. As it happens, the basic results of the subject are not duals of results concerning algebras: the point is that the category of coalgebras for a functor is not the dual of the category of algebras for it. Turning to matters closer to that of this paper, in [**?**], we noted that one of our results was known in dual form. This was a lemma used in the proof of Lemma **??** on substitution. However, our formulation of both substitution and parametric corecursion do not seem to be the duals of known results.

One paper which presents results which at first glance would seem to be duals of ours is Ésik and Labella [**?**]. The paper shows that "If the fixed point operation over a category is defined by initiality, then the equations satisfied by the fixed point operation are exactly those of iteration theories." Here is what this comes to with comparisons to this paper: For any category T, take $\mathbf{Th}(C)$ to be the 2-theory whose horizontal morphisms $n \to p$ are the functors $C^p \to C^n$. (For $C = Set$, neither the categories C^p nor the functors of this form seem to be related to what we call $\overline{p}$.) An *algebraically complete category* in the sense of [**?**] is category with a collection of $\mathcal{F}$ of functors $C^{n+p} \to C^n$ which is closed in some basic ways and with the property that for each C^p-object y, there is an initial F_y-algebra, where F_y here is $F(-, y) : C^n \to C^n$. (This does look like a dual to the notion of a parametric corecursion system, but again our derived functors are different.) Every algebraically complete category is a sub-2-theory of $\mathbf{Th}(C)$ which is an algebraically complete 2-theory. (As the present time, we do not see any interesting 2-

categorical structure behind our results.) The main result of [**?**] is that "the iteration theory identities hold in all algebraically complete 2-theories satisfying the parameter identity." (The parameter identity corresponds to the Head identity of FLR_0.) This results does not seem to be related to anything here, mostly because the fixed point operation here is derived from our notion of a parametric corecursion system, and this seems quite different from an algebraically complete category. However, we would summarize our results by saying that "If the fixed point operation over a category is defined by finality, then the equations satisfied by the fixed point operation are exactly those of iteration theories." So perhaps there is a connection somewhere.

Future work in this direction This paper suggests a number of questions. One would be to axiomatize the full consequence relation for fixed-point equations on final coalgebras. This is often difficult or impossible (see [**?**]). But it may well be that final coalgebra interpretations are easier to handle than dcpo's (for example), since the $\perp$ here "more disconnected" from the rest of the structure. A completeness result for final coalgebra interpretations would probably be the first result of that type for any proper class of iteration theories. So it certainly would be important for studies of recursion equations.

Also, it should not be hard to add the conditional to the equational logic of recursion and get the corresponding completeness result (see [**?**, **?**]).

In terms of trees, our work here deals with what is usually called *first-order substitution*. This is the substitution of trees for variables. There is also a notion of second-order substitution, where one substitutes trees for *function symbols*. Second-order substitution is more challenging to formulate in terms of final coalgebras. It is also more useful, especially if one has results guaranteeing solutions of appropriate systems of equations, such as

$$\begin{array}{lcl} \mathsf{f}(x,y) & = & \mathsf{F}(\mathsf{g}(\mathsf{g}(x)), \mathsf{f}(x,y)) \\ \mathsf{g}(x) & = & \mathsf{G}(\mathsf{f}(x,x)) \end{array} \tag{11}$$

Here F and G are "given" function symbols, either in the sense that one has a concrete domain with interpretations for these symbols, or else that one wants a solution to (**??**) as an infinite tree labeled by F and G. The algebraic semantics of recursive program schemes depends on principles of second-order substitution and the existence of solutions to systems such as (**??**). The paper [**?**] shows how to extend the work here to handle these problems.

References

[1] Peter Aczel, *Non-Well-Founded Sets*. CSLI Lecture Notes Number 14, CSLI Publications, Stanford, 1988.

[2] Peter Aczel and Nax Mendler, A final coalgebra theorem, in D. H. Pitt et al (eds.) **Category Theory and Computer Science**, Springer-Verlag, Heidelberg, 1989, 357–365.

[3] Michael Barr, Terminal coalgebras in well-founded set theory, *Theoretical Computer Science* **114** (1993) no. 2, 299–315.

[4] Stephen L. Bloom and Zoltán Ésik, *Iteration Theories: The equational logic of iterative processes,* EATCS Monographs on Theoretical Computer Science, Berlin: Springer-Verlag (1993).

[5] Stephen L. Bloom and Zoltán Ésik, "The Equational Logic of Fixed Points," *Theoretical Computer Science* **179** (1–2), 1997, 1–60.

[6] S. L. Bloom, C. C. Elgot, and J. B. Wright, "Vector iteration in pointed iterative theories," SIAM Journal of Computing 9:26–45, 1980.

[7] Zoltán Ésik, "Identities in iterative and rational algebraic theories", *Computational Linguistics and Computer Languages* 4 (1982), 95–110.

[8] Z. Ésik and A. Labella, "Equational properties of iteration in algebraically complete categories," *Theoretical Computer Science* 195 (1998), no. 1, 61–89.

[9] A.J.C. Hurkens, Monica McArthur, Yiannis N. Moschovakis, Lawrence S. Moss, and Glen Whitney, "The logic of recursive equations," *The Journal of Symbolic Logic* 63, No. 2 (1998), 451–478.

[10] Y. N. Moschovakis, "The formal language of recursion," *The Journal of Symbolic Logic* 54 (1989) 1216–1252.

[11] Y. N. Moschovakis, "The logic of functional recursion," in M. L. Dalla Chiara et al (eds.), *Logic and Scientific Methods*, Kluwer Academic Publishers, 1997, 179–207.

[12] Lawrence S. Moss, "Coalgebraic logic," in Festschrift on the occasion of Professor Rohit Parikh's 60th birthday. *Annals of Pure Applied Logic* 96 (1999), no. 1-3, 277–317.

[13] Lawrence S. Moss, "Parametric corecursion," to appear in *Theoretical Computer Science*, 2000.

[14] Lawrence S. Moss, "The algebraic semantics of recursive program schemes, revisited," ms., 2000.

[15] Lawrence S. Moss and Glen Whitney, "The AFA universe is a recursion structure," manuscript, 1997.

[16] J.J.M.M. Rutten, "Universal coalgebra: a theory of systems," to appear in *Theoretical Computer Science*, 1999.

[17] Daniele Turi and J.J.M.M. Rutten, "On the foundations of final semantics: non-standard sets, metric spaces, partial orders," *Mathematical Structures in Computer Science* 8 (1998), no. 5, 481–540.