

Algebras, Coalgebras, Monads and Comonads

Neil Ghani ^{a,1} Christoph Lüth ^{b,2} Federico de Marchi ^{a,3,5}
John Power ^{c,4}

^a *Department of Mathematics and Computer Science, University of Leicester*

^b *FB 3 — Mathematik und Informatik, Universität Bremen*

^c *Laboratory for Foundations of Computer Science, University of Edinburgh*

Abstract

Whilst the relationship between initial algebras and monads is well-understood, the relationship between final coalgebras and comonads is less well explored. This paper shows that the problem is more subtle and that final coalgebras can just as easily form monads as comonads and dually, that initial algebras form both monads and comonads.

In developing these theories we strive to provide them with an associated notion of syntax. In the case of initial algebras and monads this corresponds to the standard notion of algebraic theories consisting of signatures and equations: models of such algebraic theories are precisely the algebras of the representing monad. We attempt to emulate this result for the coalgebraic case by defining a notion cosignature and coequation and then proving the models of this syntax are precisely the coalgebras of the representing comonad.

1 Introduction

While the theory of coalgebras for an endofunctor is well-developed, less attention has been given to *comonads*. We feel this is a pity since the corresponding theory of monads on **Set** explains the key concepts of universal algebra such as *signature, variables, terms, substitution, equations* etc. Moreover, applications to base categories other than **Set** has proven fruitful in many situations, e.g. the study of multi-sorted algebraic theories as monads over **Set**^A, order-sorted theories as monads over **Pos**, the study of categories with structure

¹ Email: ng13@mcs.le.ac.uk

² Email: cxl@informatik.uni-bremen.de

³ Email: fdm2@mcs.le.ac.uk

⁴ Email: ajp@dcs.ed.ac.uk

⁵ Research supported by EPSRC grant GRM96230/01: *Categorical Rewriting: Monads and Modularity*

using monads over **Graph** or **Cat** [8], the study of rewriting using monads over **Pre** or **Cat** [16,17] and the study of higher order abstract syntax using monads over **Set** ^{\mathbb{F}} [9] (where \mathbb{F} is the skeleton of the category of finite sets). We aim for a similar framework, based upon the theory of comonads, explaining the relationship between the coalgebraic duals of the above concepts, and laying the ground for applications in categories apart from **Set**. This paper is a first step in this direction.

Traditionally, one begins with a signature Σ and then for each set of variables X , defines the term algebra $T_\Sigma(X)$. Next one defines substitution and shows this associative with the variables acting as left and right units. Finally, given equations E , one also defines the quotient algebra $T_{\Sigma,E}(X)$. Generalising this treatment of universal algebra to cover not just sets with extra structure but also algebraic structure over other mathematical objects can be achieved using a categorical reformulation of these ideas. In this categorical framework [14], one constructs from a signature Σ , an endofunctor F_Σ (which can be thought of as sending an object X of variables to the terms of depth 1 with variables from X). The term algebra T_Σ can then be characterised as the free monad over F_Σ . Being a monad gives T_Σ a well behaved notion of substitution, while being free captures the inductive nature of the term algebra. By interpreting the equations of an algebraic theory as pairs of monad morphisms between two free monads, the quotient algebra induced by the equations is modelled as their coequaliser. Crucially, the models of an algebraic theory are isomorphic to the Eilenberg-Moore category of algebras of the monad representing the algebraic theory.

Our aim is to dualise this categorical treatment of universal algebra and in particular derive coalgebraic duals of the key concepts described above. We therefore need to dualise firstly the construction of an endofunctor over a signature and secondly the free monad over an endofunctor. It turns out there is more than one possible dualisation and this is the subject matter of this paper. For example, given a finitary endofunctor $F : \mathcal{A} \rightarrow \mathcal{A}$, one can consider the mapping sending an object X to the underlying object of the initial $X + F$ -algebra, the mapping sending X to underlying object of the final $X + F$ -coalgebra, the mapping sending X to underlying object of the initial $X \times F$ -algebra and the mapping sending X to underlying object of the final $X \times F$ -coalgebra. Each of these four mappings form either a monad or a comonad as described in Table 1. Thus, there is perhaps not one dualisation at play, but two possible, and orthogonal, dualisations giving four concepts worthy of our attention.

We do not claim that this paper provides complete answers to the questions raised above. However we do feel that the questions we address will be of fundamental importance to the CMCS audience and that our approach is novel in being more abstract than most. In particular, we hope that our notions of cosignature, coequations and the categorical representation of this syntax via comonads will be of general interest and stimulate further research.

	Monads	Comonads
Initial Algebras	$\mu Y. X + FY$	$\mu Y. X \times FY$
Final Coalgebras	$\nu Y. X + FY$	$\nu Y. X \times FY$

Table 1
Algebras and Coalgebras forming Monads and Comonads

The rest of this paper is structured as follows: Sect. 2 contains preliminary definitions. In Sect. 3, we briefly review the presentation of finitary monads which we seek to dualise in the rest of the paper (the upper left entry in Table 1). Sections 4 and 5 explore the different possibilities of dualisation, with Section 4 exploring the lower left of Table 1 and Section 5 containing our proposed coalgebraic syntax with associated representation and correctness results (the right side of Table 1).

2 Preliminary Definitions and Notation

We assume the reader is familiar with standard category theory as can be found in [18]. In order to abstract away from the category **Set** we have had to employ certain constructions which some readers may not be completely familiar with. We give a short definition of these here; more details can be found in [18,3].

Locally Presentable and Accessible Categories: Let κ be a regular cardinal. A diagram \mathcal{D} is κ -filtered iff every subcategory with less than κ objects and morphisms has a compatible cocone in \mathcal{D} . An object X of a category \mathcal{A} is said to be κ -presentable iff the hom-functor $\mathcal{A}(X, _)$ preserves κ -filtered colimits. A category is *locally κ -presentable* (abbreviated as $\text{l}\kappa\text{p}$) if it is cocomplete and has, up to isomorphism, a set N_κ of κ -presentable objects such that every object is a κ -filtered colimit of objects from N_κ . A category is κ -accessible providing it is $\text{l}\kappa\text{p}$ except that it only has κ -filtered colimits rather than all colimits. A functor is κ -accessible iff it preserves κ -filtered colimits; we also say it has *rank κ* . The discrete category on N_κ is denoted \mathcal{N}_κ . The full subcategory of κ -presentable objects is denoted \mathcal{A}_κ . The inclusion functors are denoted $J_\kappa : \mathcal{N}_\kappa \rightarrow \mathcal{A}_\kappa$ and $I_\kappa : \mathcal{A}_\kappa \rightarrow \mathcal{A}$. When $\mathcal{A} = \mathbf{Set}$, the set N_{\aleph_0} can be taken to be the natural numbers which we denote \mathbb{N} .

Kan Extensions: Given a functor $I : \mathcal{A} \rightarrow \mathcal{B}$ and a category \mathcal{C} , precomposition with F defines a functor $_ \circ I : [\mathcal{B}, \mathcal{C}] \rightarrow [\mathcal{A}, \mathcal{C}]$. The problem of left and right Kan extensions is the problem of finding left and right adjoints to $_ \circ I$. More concretely, given a functor $F : \mathcal{A} \rightarrow \mathcal{C}$, the left and right Kan

extensions satisfy the natural isomorphisms

$$[\mathcal{B}, \mathcal{C}](\text{Lan}_I F, H) \cong [\mathcal{A}, \mathcal{C}](F, H \circ I) \quad [\mathcal{B}, \mathcal{C}](H, \text{Ran}_I F) \cong [\mathcal{A}, \mathcal{C}](H \circ I, F).$$

Kan extensions can be given pointwise using colimits and limits, or more elegantly using ends and coends (see [18, Chapter X] for details).

3 Initial Algebras and Monads

We begin by recalling the well known equivalence between (finitary) monads on the category **Set** and universal algebra and the generalisation of this equivalence to locally presentable categories [14,21]. We illustrate these ideas with a specific example using **Set** as the base category — examples over other base categories can be found in [21,8].

A signature declares the operations from which the terms are constructed. Usually, a signature is given as a set of operations and a function assigning each operation an arity, but we can equivalently consider it as a function $\Sigma : \mathbb{N} \rightarrow \mathbf{Set}$, assigning to each arity the set of operations of that arity. In order to abstract away from the category of **Set**, we need a notion of arity appropriate for different categories. The key observation is that the usual arities in **Set**, i.e. the natural numbers \mathbb{N} , represent the finitely presentable objects in **Set**. Hence a natural notion for arities in a kp -category is the set N_κ . Hence, a signature is a function $\Sigma : N_\kappa \rightarrow |\mathcal{A}|$, which is equivalent to a functor $\Sigma : \mathcal{N}_\kappa \rightarrow \mathcal{A}$:

Definition 3.1 (Signature) *Let \mathcal{A} be a kp category. A signature is a functor $\Sigma : \mathcal{N}_\kappa \rightarrow \mathcal{A}$, and the object of e -ary operations of Σ is $\Sigma_e \stackrel{\text{def}}{=} \Sigma(e)$.*

We now present two examples of such signatures, which we will later augment by the appropriate equations.

Example 3.2 (Monoids) *The signature $\Sigma_M : \mathbb{N} \rightarrow \mathbf{Set}$ for the theory of monoids is defined as $\Sigma_M(0) = \{e\}$, $\Sigma_M(2) = \{m\}$, $\Sigma_M(n) = \emptyset$ for all other $n \in \mathbb{N}$. Thus Σ_M declares one operation e of arity 0 (a constant) and one binary operation m .*

Example 3.3 (Categories with \top) *Categories with a terminal object can be seen as algebraic over **Cat**. The signature Σ_\top must declare the terminal object and for each object X , the unique map $!_X : X \rightarrow \top$. Since the terminal object does not depend upon any data, its arity is the empty category **0**. Since the map $!_X$ depends upon an object, its arity is the one object category **1**. Thus $\Sigma_\top(\mathbf{0}) = \mathbf{1}$, $\Sigma_\top(\mathbf{1}) = (\circ \rightarrow \circ)$ (i.e. the category with two objects and one arrow) while $\Sigma_\top(c) = \mathbf{0}$ for all other finitely presentable categories c .*

We now turn to the construction of an endofunctor over a signature Σ .

Working over **Set**, e.g. see [23], one defines a functor

$$F_{\Sigma}(X) = \coprod_{f \in \Sigma(n)} X^n$$

which calculates the terms of depth 1 whose variables come from the set X . For example, given the signature of example 3.2, the associated endofunctor is $F_M(X) = 1 + X^2$. Such *polynomial* endofunctors (i.e. built from products and coproducts) are known to preserve all κ -filtered colimits — this is an important property which later will ensure the free monad can be easily calculated. Unfortunately, it is unclear how to generalise it to categories other than **Set** and hence we take an alternative approach. Recall that by definition, left Kan extension is left adjoint to restriction $\text{Lan}_{J_{\kappa}} - \dashv - \circ J_{\kappa} : [\mathcal{N}_{\kappa}, \mathcal{A}] \rightarrow [\mathcal{A}_{\kappa}, \mathcal{A}]$ and hence we define

$$F_{\Sigma}(X) = (\text{Lan}_{J_{\kappa}} \Sigma)X \quad (1)$$

The standard formula for left Kan extensions shows that, in the case $\mathcal{A} = \mathbf{Set}$, both definitions of F_{Σ} agree, as in Example 3.2:

$$\begin{aligned} (\text{Lan}_{J_{\aleph_0}} \Sigma_M)X &= \coprod_{n \in \mathbb{N}} \mathbf{Set}_{\aleph_0}(n, X) \times \Sigma_M(n) \\ &= \mathbf{Set}(0, X) \times 1 + \mathbf{Set}(2, X) \times 1 \\ &= 1 + X^2 \end{aligned}$$

Notice that the domain of F_{Σ} is \mathcal{A}_{κ} and not \mathcal{A} , and hence F_{Σ} is not an endofunctor. However, an endofunctor $F : \mathcal{A} \rightarrow \mathcal{A}$ is κ -accessible iff it is (isomorphic to) the left Kan extension of its restriction to \mathcal{A}_{κ} ; hence there is an equivalence (2) between κ -accessible endofunctors and functors $\mathcal{A}_{\kappa} \rightarrow \mathcal{A}$ and we can regard F_{Σ} as a κ -accessible endofunctor.

$$[\mathcal{A}, \mathcal{A}]_{\kappa} \begin{array}{c} \xrightarrow{- \circ I_{\kappa}} \\ \xleftarrow{\top} \\ \xleftarrow{\text{Lan}_{J_{\kappa}} -} \end{array} [\mathcal{A}_{\kappa}, \mathcal{A}] \quad (2)$$

To recap, we have presented an abstract definition of signature Σ over an lkp -category and a construction of the associated endofunctor F_{Σ} over the signature. In the case of **Set**, these definitions and constructions agree with the usual definitions. The associated term algebra $T_{\Sigma} : \mathcal{A} \rightarrow \mathcal{A}$ is then the free monad over the endofunctor F_{Σ} , constructed in a number of ways:

Lemma 3.4 *Let F be a κ -accessible endofunctor over a lkp -category \mathcal{A} . Then the free monad T over F satisfies*

- (i) *For every X in \mathcal{A} , TX is the carrier of the initial $X + F$ -algebra.*
- (ii) *The forgetful functor $U : F\text{-Alg} \rightarrow \mathcal{A}$ from the category of F -algebras to \mathcal{A} has a left adjoint L , and $T \cong UL$*

(iii) T is colimit $\text{colim}_{i < \kappa} S_i$ of the sequence S_i where

$$S_0 = J \quad S_{n+1} = J + F \diamond S_n \quad S_\lambda = \text{colim}_{i < \lambda} S_i \quad (3)$$

and the composition of $F, E : \mathcal{A}_f \rightarrow \mathcal{A}$ is given as $F \diamond E \stackrel{\text{def}}{=} (\text{Lan}_I F \cdot E)$.

Proof. Most of the proofs are standard and can be found in [14,12]. However, note that in the last of the claims, to calculate the free monad we start forming the sequence of endofunctors S_i and we do not need to go further than the κ -filtered colimit $\text{colim}_{i < \kappa} S_i$ because F is κ -filtered. Hence T is κ -accessible and thus shares the same rank as F . \square

When we come to the dual of this construction in Sect. 5, a construction like the above will not be possible since one would be interested in the limit of co-chain, and there is no reason for an accessible endofunctor to preserve such limits. The effect is that the rank of the cofree comonad over an accessible endofunctor may increase. This change of rank underlies the technical difficulties which will arise in Sect. 5.

We obtain the two adjunctions in (4) which compose to give an adjunction $F \dashv U$ between signatures and monads.

$$[\mathcal{N}, \mathcal{A}] \begin{array}{c} \xrightarrow{\text{Lan}_{J_\kappa} -} \\ \xleftarrow[- \circ J_\kappa]{\perp} \end{array} [\mathcal{A}_\kappa, \mathcal{A}] \begin{array}{c} \xrightarrow{H} \\ \xleftarrow[V]{\perp} \end{array} \mathbf{Mon}_\kappa(\mathcal{A}) \quad (4)$$

The sequence S_i in (3) in Lemma 3.4 is called the *free algebra sequence* [12] and can be seen as a uniform calculation of the initial $X + F$ algebra. As an example of this construction, consider the signature Σ_M . As we have seen $\text{Lan}_{J_\kappa} \Sigma_M(X) = 1 + X^2$. The free algebra sequence then specialises to $S_0 X = X$ and

$$S_{n+1}(X) = X + \coprod_{e \in \mathbb{N}} \mathcal{A}(e, S_n(X)) \times \Sigma_M(e) = X + 1 + S_n(X)^2$$

from which we can see S_n as defining the terms of depth at most n , e.g. $S_0(X)$ contains the variables X , and $S_1(X)$ contains the variables X , the canonical element of 1 representing the unit of the monoid and a pair of elements of $S_0(X)$ which can be thought of as the multiplication of these elements.

Within this framework, to give equations is to give another signature E and two natural transformations $\sigma, \tau : E \rightarrow UFB$. One should regard E as giving the shape of the equations and σ' and τ' as the actual equations.

Example 3.5 *Given the monoids example, there are three equations we wish to assert: left unit, right unit (both unary) and associativity (ternary). Hence we set $E(3) = \{a\}$, $E(1) = \{l, r\}$ and $E(n) = \emptyset$ for all other n .*

σ and τ define the left, and the right hand side of equations, respectively:

$$\begin{array}{lll} \sigma(l) = m(e, x) & \sigma(r) = m(x, e) & \sigma(a) = m(x, m(y, z)) \\ \tau(l) = x & \tau(r) = x & \tau(a) = m(m(x, y), z) \end{array}$$

For example 3.3, one equation is needed to force the uniqueness of $!_X : X \rightarrow \top$, i.e. $!_Y \cdot f = !_X$ for all $f : X \rightarrow Y$ (see [8] for details).

The advantage of this definition of equations is that it allows an elegant derivation of the representing monad for an algebraic theory: given such a theory $\sigma, \tau : E \rightarrow UFB$, under adjunction (4) we have two monad morphisms σ', τ' whose coequaliser is the representing monad T . This construction makes sense, because the category of models of the algebraic theory is isomorphic to the Eilenberg-Moore-category of the representing monad.

In (4), both adjunctions are monadic and their composition is of *descent type*, which means that each component of the counit $\varepsilon_B : FUB \Rightarrow B$ is a coequaliser. This means that *every* κ -accessible monad $T = (T, \eta, \mu)$ on \mathcal{A} is a coequaliser of two free monads over two signatures $B, E : \mathcal{N} \rightarrow \mathcal{A}$ in the category of κ -accessible monads over \mathcal{A} , or every monad is represented by equations in this general sense.

4 Final Coalgebras and Monads

Given a signature Σ we have constructed an associated endofunctor F_Σ , which can be thought of as calculating the closed terms of depth 1. By closing F_Σ under composition, or more formally by considering the free monad over F_Σ , we get the term algebra T_Σ . Concretely $T_\Sigma X$ is the carrier of the initial $X + F_\Sigma$ -algebra. In this section we investigate the map T_Σ^∞ sending an object X to the carrier of the final $X + F_\Sigma$ -coalgebra, and apply our results to simplify some recent developments in the semantics of lazy datatypes in functional programming languages. A more general version of Lemma 4.2 appears in [1] and is implicit in [20].

In comparing $\mu Y.X + FY$ with $\nu Y.X + FY$, the first observation [5] is that the final coalgebra of a **Set**-endofunctor can be regarded as a Cauchy completion of the initial algebra. Thus, if F arises from a signature and hence $\mu Y.X + FY$ is the usual term algebra, then $\nu Y.X + FY$ is the set of terms of finite and infinite depth. This result was extended to lfp categories in [2]. The goal of this section is to ascertain the structure possessed by the map sending X to the carrier of the final $X + F_\Sigma$ -coalgebra. Our answer is that this map also forms a monad. In proving this result we use the following result [2]:

Lemma 4.1 *Assume that \mathcal{A} is lfp, that the unique map $! : 0 \rightarrow 1$ is a strong monomorphism, that F preserves monos and ω^{op} -chains and that there is a map $p : 1 \rightarrow F0$. Then for each object $X \in \mathcal{A}$, $\mathcal{A}(X, \mu F)$ and $\mathcal{A}(X, \nu F)$ are metric spaces with the latter being the Cauchy completion of the former.*

We use this result to prove our main lemma:

Lemma 4.2 *Let F be a polynomial endofunctor and \mathcal{A} satisfy the premises of lemma 4.1. Then the map T^∞ assigning an object X to the carrier of the final $X + F$ -coalgebra carries the structure of a monad.*

Proof. We actually prove the lemma for the case $\mathcal{A} = \mathbf{Set}$ since the general case involves slightly clumsy reasoning with hom-sets but is entirely analogous. We prove that T^∞ is a monad by constructing a Kleisli structure for it, namely by defining the following operations: i) for each object X , an arrow $\eta_X^\infty : X \rightarrow T^\infty X$; ii) for each pair of objects X, Y a function $s_{X,Y} : \mathcal{A}(X, T^\infty Y) \rightarrow \mathcal{A}(T^\infty X, T^\infty Y)$ such that

$$s_{X,X}(\eta_X^\infty) = 1_{T^\infty X} \quad s_{X,Z}(s_{Y,Z}g \circ f) = s_{Y,Z}g \circ s_{X,Y}f \quad s_{X,Y}(f) \circ \eta_X^\infty = f$$

Firstly define $\eta_X^\infty = \epsilon_X \circ \eta_X$ where η is the unit of the free monad on F and ϵ_X is the unique comparison map between the initial $X + F$ -algebra and the final $X + F$ -coalgebra.⁶ Next define $s_{X,Y}(f)$ as follows

$$\begin{array}{ccccc} X & \xrightarrow[\quad f \quad]{\eta_X} & TX & \xrightarrow{\epsilon_X} & T^\infty X \\ & & \downarrow !_{[f,h]} & & \\ & & T^\infty Y & \xleftarrow{s(f) = !_{[f,h]}^*} & \end{array}$$

where $h : FT^\infty Y \rightarrow T^\infty Y$ is the obvious map derived from the structure map of $T^\infty Y$. Then $[f, h] : X + FT^\infty Y \rightarrow T^\infty Y$ endows $T^\infty Y$ with an $X + F$ -algebra structure and hence induces a unique map $!_{[f,h]} : TX \rightarrow T^\infty Y$ from the initial $X + F$ -algebra. Diagram chasing shows that $!_{[f,h]}$ is uniformly continuous with respect to the relevant metrics and hence by lemma 4.1 there is an extension $!_{[f,h]}^*$ which can be taken to be $s_{X,Y}(f)$. The first equation follows since by uniqueness $!_{[\eta_X^\infty, h]} = \epsilon_X$ and the extension of ϵ_X is the identity. The equation $s_{X,Z}(s_{Y,Z}g \circ f) = s_{Y,Z}g \circ s_{X,Y}f$ holds if their restrictions via ϵ_X are the same which in turn holds if they arise from the same algebra structure on $T^\infty Z$. This holds because the two algebra structures are both inherited from the map $s_{Y,Z}(g) \circ f : X \rightarrow T^\infty Z$. The last equation follows since $s_{X,Y}(f) \circ \eta_X^\infty = s_{X,Y}(f) \circ \epsilon_X \circ \eta_X = !_{[f,h]} \circ \eta_X = f$ \square

Unlike the monad T , the monad T^∞ has a larger rank than F :

Lemma 4.3 *Let F be a polynomial endofunctor on \mathbf{Set} . Then the monad T_Σ^∞ has rank \aleph_1 .*

Proof. The lemma can be proven directly using an interchange law between limits and filtered colimits. \square

⁶ Actually applying Lemma 4.1 requires that $X + F0$ be non empty which is the case when $X \neq 0$. Thus the maps η_0^∞ and $s_{0,Y}$ and their properties must be established separately, but this all follows trivially from the initiality of 0

Notice that lemma 4.2 indicates that there is no need to develop a new syntax for the monad T_Σ^∞ since the canonical syntax should be the finite and infinite depth terms of the associated monad T_Σ . We finish this section with an application of this result to reasoning in functional programming.

4.1 The Generic Approximation Lemma

The *generic approximation lemma* [10] is a proof principle for reasoning about functions in a lazy functional programming language (such as Haskell). The approximation lemma itself pertains to lists and states that given a function

```
approx (n+1) []      = []
approx (n+1) (x:xs) = x:(approx n xs)
```

two lists **xs** and **ys** are equal iff $\forall n. \text{approx } n \text{ xs} = \text{approx } n \text{ ys}$. Note the lack of a base case: **approx** 0 **x** is \perp (i.e., the denotation of undefined) in the denotational model, but because of non-strictness, **approx** n **x** (with $n > 0$) is defined. This principle can be applied to other datatypes such as trees:

```
data Tree a = Leaf a | Node Tree a Tree
approx (n+1) (Leaf x) = Leaf x
approx (n+1) (Node l x r) = Node (approx n l) x (approx n r)
```

Two trees **t1** and **t2** are equal iff $\forall n. \text{approx } n \text{ t1} = \text{approx } n \text{ t2}$. [10] proves the generic approximation lemma using the standard denotational semantics of functional programming languages, where types are interpreted as cpo's, programs as continuous functions and recursive datatypes as least fixed points of functors. That is, the correctness of the proof principle depends upon the semantic category chosen; we have already seen the implicit use of \perp in the definition of **approx**.

We propose an alternative and, we believe, more natural derivation of the approximation lemma which is independent of the particular denotational model chosen. The definition of a polymorphic datatype is usually interpreted as the free monad T_Σ over its signature Σ . However, this does not capture laziness, since T_Σ consists of only finite terms. Instead, we model such a datatype by the monad T_Σ^∞ . Since $T_\Sigma^\infty(X)$ is the final $X + \Sigma$ -coalgebra and since F_Σ is a polynomial endofunctor, $T_\Sigma^\infty(X)$ can be calculated as the limit of the following ω^{op} -chain $1 \leftarrow (X + F)1 \leftarrow (X + F)^2 1 \leftarrow \dots$. The universal property of the limit states that two elements x and y of this limit will be equal iff for each n , $\pi_n(x) = \pi_n(y)$ where π_n is the n -th projection. But these projections are precisely the approximation function for the datatype. Notice how the categorical argument replaces the semantic dependency on \perp with use of a co-chain beginning with 1. This establishes the correctness of the generic approximation lemma independently of any specific denotational model. More generally, a lazy functional language like Haskell combines aspects appropriately modelled by a monad (being finitely generated), while others are more appropriately captured by final semantics (laziness and non-termination).

5 Final Coalgebras and Comonads

We now turn to another possible dualisation of Sect. 3 based upon the idea of mapping an object X to the carrier of the final $X \times F$ -coalgebra for some κ -accessible functor F . The elegance of the monadic approach to algebraic theories suggests that we can apply these techniques to the development of coalgebraic syntax by dualising them appropriately.

5.1 Cosignatures and their Coalgebras

Recall that the heart of the categorical approach to universal algebra is adjunction (4). The dualisation outlined in this section can be summed up as replacing the left adjoint to U with a right adjoint and the replacement of monads with comonads. As a result, the definition of a cosignature turns out to be formally the same as that of a signature.

Definition 5.1 (Cosignature) *Let \mathcal{A} be a κp -category with arities \mathcal{N}_κ . A κ -cosignature is a functor $B : \mathcal{N}_\kappa \rightarrow \mathcal{A}$.*

Recall that in Section 3, we constructed a κ -accessible endofunctor from a signature by first taking a left Kan extension, and then used equivalence (2) to get a κ -accessible endofunctor. Therefore, in this section we take the right Kan extension of a cosignature B to obtain a functor $\text{Ran}_{J_\kappa} B : \mathcal{A}_\kappa \rightarrow \mathcal{A}$, followed by the same equivalence to obtain a κ -accessible endofunctor on \mathcal{A} . The standard formula for the right Kan extension gives us

$$(\text{Ran}_{J_\kappa} B)(X) = \prod_{c \in \mathcal{N}_\kappa} \mathcal{A}(X, c) \rhd Bc$$

where $U \rhd B$ is the U -fold product of B , or more formally the representing object for the functor $[U, \mathcal{A}(-, B)] : \mathcal{A} \rightarrow \mathbf{Set}$. When $\mathcal{A} = \mathbf{Set}$, then this operation is simply exponentiation, i.e. $U \rhd B = [U, B]$.

Thus, although signatures and cosignatures are formally the same, the endofunctors they generate are very different. For example, note that while the default value for signatures is 0, if there is a single arity c such that $B(c) = 0$, then $(\text{Ran}_{J_\kappa} B)(X) = 0$. In fact, the default value for cosignatures is 1 since $U \rhd 1 = 1$ and hence if c is an arity such that $B(c) = 1$, then this arity will contribute nothing to the right Kan extension. Here are two example cosignatures which we will explore further below:

Example 5.2 *Define the cosignature B_p by $B_p(2) = 2$ and $B_p(c) = 1$ for all other arities. Then $\text{Ran}_{J_\kappa} B_p(X) = [\mathbf{Set}(X, 2), 2] = [[X, 2], 2]$.*

Define the cosignature B_{p_ω} by $B_{p_\omega}(2) = \omega$ and $B_{p_\omega}(c) = 1$ for all other arities. Then $(\text{Ran}_{J_\kappa} B_{p_\omega})(X) = [\mathbf{Set}(X, 2), \omega] = [[X, 2], \omega]$.

Recall that in order to turn a functor $\mathcal{A}_\kappa \rightarrow \mathcal{A}$ into a κ -accessible endofunctor, one takes its left Kan extension. Given a cosignature B , we can thus consider coalgebras of the endofunctor $\text{Lan}_{I_\kappa} \text{Ran}_{J_\kappa} B$. The following is

an important result from [11]:

Lemma 5.3 *Let $F : \mathcal{A} \rightarrow \mathcal{A}$ be an accessible endofunctor on a locally presentable category. Then $F\text{-}\mathbf{coalg}$ is locally presentable and the κ -presentable objects of $F\text{-}\mathbf{coalg}$ are simply those coalgebras whose underlying object is κ -presentable in \mathcal{A} .*

Proof. The proof uses the fact that the category of all (small) accessible categories has all weighted limits [19, Theorem 5.1.6]. Then the category $F\text{-}\mathbf{coalg}$ can be constructed as such a limit, namely the *serter* [13] of

$$\mathcal{C} \begin{array}{c} \xrightarrow{1} \\ \xrightarrow{F} \end{array} \mathcal{C}$$

The second part of the lemma follows from the fact that colimits in $F\text{-}\mathbf{coalg}$ are formed pointwise. \square

This result allows us to consider κ -presentable $\text{Lan}_{I_\kappa} \text{Ran}_{J_\kappa} B$ -coalgebras:

Lemma 5.4 *Let B be a κ -cosignature. A κ -presentable $\text{Lan}_{I_\kappa} \text{Ran}_{J_\kappa} B$ -coalgebra consists of a κ presentable object X together with for every arity c , a function $h_c : \mathcal{A}(X, c) \rightarrow \mathcal{A}(X, Bc)$.*

Proof. Given any κ -accessible endofunctor F , and κ -presentable object X , $\text{Lan}_{I_\kappa} F(X) = F(X)$. Hence coalgebras of the form mentioned are

$$\begin{aligned} \mathcal{A}(X, (\text{Ran}_{J_\kappa} B)(X)) &\cong \mathcal{A}(X, \prod_{c \in \mathcal{N}_\kappa} \mathcal{A}(X, c) \multimap Bc) \\ &\cong \prod_{c \in \mathcal{N}_\kappa} \mathcal{A}(X, \mathcal{A}(X, c) \multimap Bc) \cong \prod_{c \in \mathcal{N}_\kappa} [\mathcal{A}(X, c), \mathcal{A}(X, Bc)] \end{aligned}$$

In the theory of monads, a crucial result states that the models of an algebraic presentation are isomorphic to the category of algebras of the theories representing monad. In order to emulate this result we require a notion of model (or co-model if you like) for a κ -cosignature B , and Lemma 5.4 suggests this be $\text{Lan}_{I_\kappa} \text{Ran}_{J_\kappa} B$ -coalgebra.

Example 5.5 *A model of the cosignature B_p is given by a finite set X together with a function $\mathbf{Set}(X, 2) \rightarrow \mathbf{Set}(X, 2)$. If we regard a map $f : X \rightarrow 2$ as a subset of X , or a predicate over X , then such a model can be regarded as a map between predicates.*

Similarly, a model for the cosignature B_{p_ω} consists of a finite set X and a function $\mathbf{Set}(X, 2) \rightarrow \mathbf{Set}(X, \omega)$. This is clearly related to the previous example if we think of it as mapping binary partitions of X to ω -ary partitions.

5.2 The Representing Comonad of a Cosignature

Recall that in the algebraic case one starts with a signature, constructs an endofunctor, and then considers the free monad over the endofunctor. The second part of the dualisation is to consider the cofree comonad over an endofunctor. The following pair of results from [11] provides the framework for

this discussion:

Lemma 5.6 *The following conditions on a functor $F : \mathcal{A} \rightarrow \mathcal{A}$ are equivalent:*

- (i) *There is a cofree comonad on F .*
- (ii) *The forgetful functor $F\text{-}\mathbf{coalg} \rightarrow \mathcal{A}$ is comonadic.*
- (iii) *The forgetful functor $F\text{-}\mathbf{coalg} \rightarrow \mathcal{A}$ has a right adjoint*
- (iv) *(If \mathcal{A} has products) For every object X , the functor $X \times F$ has a final coalgebra.*

Similar to the monadic case, only accessible endofunctors have a cofree comonad. For example, there is no final coalgebra for the powerset functor $P : \mathbf{Set} \rightarrow \mathbf{Set}$, as such a coalgebra would be an isomorphism $X \cong PX$. The important result from [19,11] is the following:

Lemma 5.7 *If \mathcal{A} is locally presentable and F is accessible, then $F\text{-}\mathbf{coalg}$ is locally presentable and there is a cofree comonad over F .*

Proof. Recall from lemma 5.3 that $F\text{-}\mathbf{coalg}$ is accessible. Cocompleteness of $F\text{-}\mathbf{coalg}$ follows since the forgetful functor creates colimits and \mathcal{A} is cocomplete. Since the forgetful functor is a colimit-preserving functor between locally presentable categories, it follows from the Special Adjoint Functor Theorem [18, Theorem V.8.1] that the forgetful functor has a right adjoint. By lemma 5.6, there is a cofree comonad over F . \square

Let $\mathbf{Com}_\kappa(\mathcal{A})$ be the category of κ -accessible comonads on \mathcal{A} , and further $\mathbf{ACom}(\mathcal{A})$ be the category of accessible comonads on \mathcal{A} . For every κ , $\mathbf{Com}_\kappa(\mathcal{A})$ is a full subcategory of $\mathbf{ACom}(\mathcal{A})$. Similarly, let $[\mathcal{A}, \mathcal{A}]_\kappa$ be the category of endofunctors with rank κ , and $\mathbf{AEnd}(\mathcal{A})$ be the category of accessible endofunctors. Piecing together the above results we have the following:

Lemma 5.8 *The forgetful functor $V : \mathbf{ACom}(\mathcal{A}) \rightarrow \mathbf{AEnd}(\mathcal{A})$ has a right adjoint R .*

Proof. Given an accessible endofunctor F , let the right adjoint of the forgetful functor $U_F : F\text{-}\mathbf{coalg} \rightarrow \mathcal{A}$ be K_F . Then choosing $R(F)$ to be $U_F K_F$ we can deduce that $R(F)$ is accessible because U_F is accessible (as a left adjoint, it preserves all colimits) and K_F is accessible by [3, Proposition 1.66 on p. 52].

The action of the right adjoint on accessible endofunctors has already been given. Given a natural transformation $\alpha : F_1 \Rightarrow F_2$, this induces a functor $\alpha^* : F_1\text{-}\mathbf{coalg} \rightarrow F_2\text{-}\mathbf{coalg}$ which commutes with the respective forgetful functors. Using the dual of [6, Theorem 3 on p. 128] and the comonadicity of these categories of coalgebras, this in turn induces a natural transformation $R(\alpha) : R(F_1) \Rightarrow R(F_2)$. This defines the functor $R : \mathbf{AEnd}(\mathcal{A}) \rightarrow \mathbf{ACom}(\mathcal{A})$. The fact that it is right adjoint to V is easily verified. \square

As opposed to the monadic case, the cofree comonad RM on an endofunctor $M : \mathcal{A} \rightarrow \mathcal{A}$ of rank λ need not have rank λ ; as a simple counterexample, consider the endofunctor $M : \mathbf{Set} \rightarrow \mathbf{Set}$ defined as $MX \stackrel{\text{def}}{=} A \times X$ for a

fixed set A , which clearly is finitary (has rank \aleph_0). We know that the value of RM for a set X is given by $RM(X) = \nu Y. X \times MY = \nu Y. X \times A \times Y$ which is the set of all infinite streams of elements of $X \times A$. Now consider a countably infinite set Y , then $RM(Y)$ contains a sequence with infinitely many different elements from Y . This sequence can not be given from any of the finite subsets $Y_0 \subseteq Y$, which shows that RM has a rank larger than \aleph_0 . As we saw in Sect. 4, in general calculating coalgebras of accessible endofunctors invariably seems to increase their rank — see [24] for an investigation of this phenomenon.

Using the equivalence between $[\mathcal{A}_\kappa, \mathcal{A}]$ and $[\mathcal{A}, \mathcal{A}]_\kappa$, we now have the following functors, where $_ \circ J_\kappa$ is the precomposition with J_κ and V is the forgetful U for comonads of rank κ :

$$\begin{array}{ccc}
 [\mathcal{N}_\kappa, \mathcal{A}] & \begin{array}{c} \xrightarrow{\text{Ran}_{J_\kappa}} \\ \top \\ \xleftarrow{_ \circ J_\kappa} \end{array} & [\mathcal{A}_\kappa, \mathcal{A}] & \cong & [\mathcal{A}, \mathcal{A}]_\kappa & \begin{array}{c} \xleftarrow{V} \\ \downarrow \\ \text{AEnd}(\mathcal{A}) \end{array} & \text{Com}_\kappa(\mathcal{A}) & \begin{array}{c} \downarrow \\ \text{ACom}(\mathcal{A}) \end{array} \\
 & & & & & \begin{array}{c} \xrightarrow{R} \\ \top \\ \xleftarrow{V} \end{array} & & (5)
 \end{array}$$

We further define the composite functors

$$\begin{array}{ll}
 V_\kappa : \text{Com}_\kappa(\mathcal{A}) \rightarrow [\mathcal{N}_\kappa, \mathcal{A}] & V_\kappa \stackrel{\text{def}}{=} (_ \circ J_\kappa) \cdot V \\
 R_\kappa : [\mathcal{N}_\kappa, \mathcal{A}] \rightarrow \text{ACom}(\mathcal{A}) & R_\kappa \stackrel{\text{def}}{=} R \cdot \text{Ran}_{J_\kappa}
 \end{array}$$

As we have seen, the rank of $R_\kappa B$ may be greater than the rank of B , and hence the codomain of R_κ is not the domain of V_κ , but rather $\text{ACom}(\mathcal{A})$. So although R_κ and V_κ can not be adjoint, there is the following isomorphism:

Lemma 5.9 *For any κ -accessible comonad G and κ -cosignature $B : \mathcal{N}_\kappa \rightarrow \mathcal{A}$ there is an isomorphism*

$$[\mathcal{N}_\kappa, \mathcal{A}](V_\kappa G, B) \cong \text{ACom}(\mathcal{A})(G, R_\kappa B) \quad (6)$$

which is natural in G and B .

Proof. The isomorphism is shown by the following chain of natural isomorphisms provided by the two adjunctions in (5) and the full and faithful embedding of $[\mathcal{A}, \mathcal{A}]_\kappa$ into $\text{AEnd}(\mathcal{A})$:

$$\begin{aligned}
 [\mathcal{N}_\kappa, \mathcal{A}](V_\kappa G, B) &\cong [\mathcal{N}_\kappa, \mathcal{A}](_ \circ J_\kappa V G, B) \\
 &\cong [\mathcal{A}_\kappa, \mathcal{A}](V G, \text{Ran}_{J_\kappa} B) \\
 &\cong \text{AEnd}(\mathcal{A})(V G, \text{Ran}_{J_\kappa} B) \\
 &\cong \text{ACom}(\mathcal{A})(G, R \text{Ran}_{J_\kappa} B) \\
 &\cong \text{ACom}(\mathcal{A})(G, R_\kappa B)
 \end{aligned}$$

□

So given a κ -cosignature B we have constructed its representing comonad $R_\kappa B$. Note that by lemma 5.6, the category of coalgebras $R_\kappa B\text{-}\mathbf{Coalg}$ is isomorphic to the category $\text{Lan}_{I_\kappa} \text{Ran}_{J_\kappa} B\text{-}\mathbf{coalg}$. Restricting ourselves to κ -presentable coalgebras, we have that the κ -presentable coalgebras of the representing comonad $R_\kappa B$ are isomorphic to the κ -presentable models of the cosignature B as defined earlier. This is our partial dualisation of the result stating that the models of a signature are isomorphic to the Eilenberg-Moore category of the representing monad.

5.3 Coequational Presentations and their Representing Comonad

In this section, we continue the dualisation by defining *coequational presentations*, deriving a representing comonad for such a coequational presentation, and relating the coalgebras of the representing comonad to the models of the presentation. As we have seen above, equations are interpreted as a pair of monad morphisms between free monads; the representing monad for an equational presentation is then defined to be the coequaliser of these monad morphisms. Dualising this requires a coequational presentation to form a pair of comonad morphisms between cofree comonads and taking the representing comonad for the coequational presentation to be the equaliser of these comonad morphisms.

Definition 5.10 (Coequational Presentations) *A coequational presentation is given by two cosignatures $B : \mathcal{N}_\kappa \rightarrow \mathcal{A}$ and $E : \mathcal{N}_\lambda \rightarrow \mathcal{A}$ (where $R_\kappa B$ is λ -accessible), and two comonad morphisms $\sigma, \tau : R_\kappa B \rightarrow R_\lambda E$.*

Under (6) the maps $\sigma, \tau : R_\kappa B \rightarrow R_\lambda E$ are given by maps $\sigma', \tau' : V_\lambda R_\kappa B \rightarrow E$, which in turn are families $\sigma'_c, \tau'_c : R_\kappa Bc \rightarrow Ec$ of maps for $c \in \mathcal{N}_\lambda$. So, as opposed to the monadic case where an equation is given by a pair of terms, a coequation consists of two partitionings of the coterms; for example, if $E(n) = 2$, each of σ and τ partition $R_\kappa B(n)$ into two subsets.

As mentioned above, given a coequational presentation, our intention is to define its representing comonad to be the equaliser of the comonad morphisms:

$$G \longrightarrow R_\kappa B \begin{array}{c} \xrightarrow{\sigma} \\ \xrightarrow{\tau} \end{array} R_\lambda E \quad (7)$$

Proving that these equalisers exist is made easier by our abstract categorical setting which provides us with an alternative definition of the coalgebras of a comonad. Recall that such coalgebras are given by an object $x \in \mathcal{A}$ and a map $\alpha : x \rightarrow Gx$ which commutes with the unit and counit of G . First, observe that an object x of \mathcal{A} is given by a map $X : \mathbf{1} \rightarrow \mathcal{A}$ (where $\mathbf{1}$ is the one-object category). Further, the functor category $[\mathbf{1}, \mathcal{A}]$ is isomorphic to \mathcal{A} , and we have

$$\mathcal{A}(x, Gx) \cong [\mathbf{1}, \mathcal{A}](X, G \circ X) \cong [\mathcal{A}, \mathcal{A}](\text{Lan}_X X, G), \quad (8)$$

so to give a map $x \rightarrow Gx$ is equivalent to giving a natural transformation from $\text{Lan}_X X \Rightarrow G$. In fact, $\text{Lan}_X X$ is a comonad:

Lemma 5.11 *$\text{Lan}_X X$ is a comonad. If X is κ -presentable, then $\text{Lan}_X X$ is κ -accessible.*

Proof. Using the standard formula for left Kan extensions, $\text{Lan}_X X(a) = \mathcal{A}(X, a) \otimes X$ (where \otimes is the *tensor*; e.g. in **Set**, it is given by the usual product). If X is κ -presentable, then $\mathcal{A}(X, -)$ preserves all κ -filtered colimits, as does $- \otimes X$; hence $\text{Lan}_X X$ is κ -accessible.

To have a comonad structure, we need natural transformations $\text{Lan}_X X \Rightarrow 1$ and $\text{Lan}_X X \Rightarrow \text{Lan}_X X \circ \text{Lan}_X X$ which satisfy the comonad laws. The first of these is given by the image of the identity transformation on X under the isomorphism $[1, \mathcal{A}](X, X) \cong [\mathcal{A}, \mathcal{A}](\text{Lan}_X X, 1_A)$. The second is given by the image under the isomorphism $[1, \mathcal{A}](X, \text{Lan}_X X \circ \text{Lan}_X X \circ X) \cong [\mathcal{A}, \mathcal{A}](\text{Lan}_X X, \text{Lan}_X X \circ \text{Lan}_X X)$ of the transformation

$$X \xRightarrow{\epsilon} \text{Lan}_X X \circ X \xRightarrow{\text{Lan}_X X \epsilon} \text{Lan}_X X \circ \text{Lan}_X X \circ X$$

where ϵ is the canonical transformation $X \Rightarrow \text{Lan}_X X \circ X$. That the counit and comultiplication obey the comonad laws is easily verified. \square

The fact that $\text{Lan}_X X$ is a comonad allows us to strengthen equation (8) to obtain the promised characterisation of the coalgebras of a comonad.

Proposition 5.12 *A coalgebra for a comonad G is given by an object X of \mathcal{A} and a map $\text{Lan}_X X \Rightarrow G$ in $\mathbf{ACom}(\mathcal{A})$.*

Proof. We have already seen that the structure map of the coalgebra is precisely a natural transformation between the two functors. It is then routine to verify that the properties of the structure map of a coalgebra correspond to the laws of a comonad morphism. \square

With Proposition 5.12, we can show the existence of equalisers:

Lemma 5.13 *The category of accessible comonads $\mathbf{ACom}(\mathcal{A})$ has all equalisers.*

Proof. Consider two comonad morphisms $M \xrightleftharpoons[\tau]{\sigma} E$. We define the category $(M, E)\text{-}\mathbf{Coalg}$ to be given by coalgebras $\alpha : \text{Lan}_X X \rightarrow M$ of M such that $\sigma \cdot \alpha = \tau \cdot \alpha$. Recall from the proof of lemma 5.4 how we constructed the category of coalgebras as a weighted limit; using a similar diagram called an *equifier*, we can construct $(M, E)\text{-}\mathbf{Coalg}$ as a weighted limit as well, making it accessible. With $(M, E)\text{-}\mathbf{Coalg}$, M and E accessible, the forgetful functor from $(M, E)\text{-}\mathbf{Coalg}$ to \mathcal{A} has a right adjoint; and post-composition of this right adjoint with the forgetful functor gives the equaliser of M and E . \square

We finish by relating the models of a coequational presentation to the coalgebras of its representing comonad. If G is the representing comonad of the coequational presentation $\sigma, \tau : R_\kappa B \rightarrow R_\lambda E$, then a G -coalgebra is a map $\alpha : \text{Lan}_X X \rightarrow G$ which by the universal property of equalisers is a map $\alpha' : \text{Lan}_X X \rightarrow R_\kappa B$ such that $\sigma \cdot \alpha = \tau \cdot \alpha$, or in other words a coalgebra for B which equalises σ and τ . In this case, we say that the coalgebra $\text{Lan}_X X \rightarrow R_\kappa B$ satisfies the coequational presentation $\sigma, \tau : R_\kappa B \rightarrow R_\lambda E$. Following the calculations above, if X is κ -presentable the map $\alpha : \text{Lan}_X X \rightarrow R_\kappa B$, gives us two $R_\lambda E$ -coalgebras, which by Lemma 5.4 are families $\sigma'' : \mathcal{A}(X, c) \rightarrow \mathcal{A}(X, Ec)$ and $\tau'' : \mathcal{A}(X, c) \rightarrow \mathcal{A}(X, Ec)$ of maps for all $c \in \mathcal{N}_\lambda$, which have to be equal if $\alpha : \text{Lan}_X X \rightarrow R_\kappa B$ satisfies the presentation $\sigma, \tau : R_\kappa B \rightarrow R_\lambda E$.

6 Conclusions, Related and Further Work

This paper set out to analyse the essence of the categorical approach to universal algebra based upon the theory of accessible monads and then derive their coalgebraic counterparts. While still only work in progress, we feel we have made a definite contribution which will be of interest to the general coalgebra community. In particular, the derivation of the infinitary monad corresponding to the collection of final $X + F$ -coalgebras seems to have ready applications in lazy functional programming. In addition, the derivation of the cofree comonad representing a cosignature and the interpretation of a coequational presentation as an equaliser of cofree comonads are dualisations of the corresponding results in the standard theory. We have also related the coalgebras of the representing comonad to the models of the coequational presentations.

The major technical challenge was caused by the increase in rank when passing from a cosignature to its representing comonad. For example, this meant that the characterisation of coalgebras only holds for coalgebras up to a certain rank. In addition, this increase in rank makes it unlikely that every accessible comonad can be given as a coequational presentation (i.e. as an equaliser of cofree comonads), as is possible for finitary monads.

Of course there remains much to be done. Our approach has been very abstract and perhaps the most important next step is to make precise the relationship between our approach and others pursued in the community.

For example, Cirstea [7] defines an *abstract cosignature* as a functor $F : \mathcal{C} \rightarrow \mathcal{C}$ where \mathcal{C} has all finite limits and limits of ω^{op} -chains, and F preserves pullbacks and such limits. Then, an *observer* is given by a functor $K : \mathcal{C} \rightarrow \mathcal{C}$ and a natural transformation $c : U \rightarrow K \cdot U$, and a coequation by two observers (K, l) and (K, r) . Thus, an observer is a family $\beta : C \rightarrow KC$ of maps for all F -coalgebras $\alpha : X \rightarrow FX$. This notion corresponds roughly with a comonad morphism $R_\kappa B \Rightarrow R_\lambda E$, since we always have the cofree coalgebra $R_\kappa BX \rightarrow R_\kappa B R_\kappa BX$. However, the difference is the stronger assumption that the abstract cosignature F preserves limits of ω^{op} , which allows the con-

struction of a final coalgebra, and reasoning about it; in our more abstract setting all we have is the existence of a right adjoint and the corresponding weaker properties.

We have not as yet developed the logical calculus underlying the categorical constructions of the comonadic semantics. It seems this underlying logic will have a modal flavour, and hence research in this area, e.g. by Kurz [15], seems relevant. Finally the use of coequations in recent work on co-Birkhoff theorems [4,22] should be compared with our coequations.

References

- [1] P. Aczel, J. Adamek, and J. Velebil. Iteration monads. In U. Montanari, editor, *Proceedings CMCS'01*, volume 44, 2001.
- [2] J. Adamek. On final coalgebras of continuous functors. To appear.
- [3] J. Adamek and J. Rosický. *Locally Presentable and Accessible Categories*. Number 189 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1994.
- [4] S. Awodey and J. Hughes. The coalgebraic dual of birkhoffs variety theorem. <http://www.contrib.andrew.cmu.edu/user/jesse/papers/CoBirkhoff.ps.gz>, 2000.
- [5] M. Barr. Terminal algebras in well-founded set theory. *Theoretical Computer Science*, 114:299–315, 1993.
- [6] M. Barr and C. Wells. *Toposes, Triples and Theories*. Number 278 in Grundlehren der mathematischen Wissenschaften. Springer Verlag, 1985.
- [7] C. Cirstea. An algebra-coalgebra framework for system specification. In *Proceedings CMCS 2000*, volume 33 of *Electronic Notes in Theoretical Computer Science*, 2000.
- [8] E. J. Dubuc and G. M. Kelly. A presentation of topoi as algebraic relative to categories or graphs. *Journal for Algebra*, 81:420–433, 1983.
- [9] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proc. LICS'99*, 1999.
- [10] G. Hutton and J. Gibbons. The generic approximation lemma. *Information Processing Letters*, 2001. To appear.
- [11] P. T. Johnstone, A. J. Power, T. Tsujishita, H. Watanabe, and J. Worrell. The structure of categories of coalgebras, 1998.
- [12] G. M. Kelly. A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves and so on. *Bulletins of the Australian Mathematical Society*, 22:1–83, 1980.

- [13] G. M. Kelly. Elementary observations on 2-categorical limits. *Bulletins of the Australian Mathematical Society*, 39:301–317, 1989.
- [14] G. M. Kelly and A. J. Power. Adjunctions whose counits are coequalizers, and presentations of finitary monads. *Journal for Pure and Applied Algebra*, 89:163–179, 1993.
- [15] A. Kurz. *Logics for Coalgebra and Applications to Computer Science*. Dissertation, Ludwig-Maximilians-Universität München, 2000.
- [16] C. Lüth. Compositional term rewriting: An algebraic proof of Toyama’s theorem. In H. Ganzinger, editor, *Rewriting Techniques and Applications, 7th International Conference*, number 1103 in LNCS, pages 261–275. Springer Verlag, July 1996.
- [17] C. Lüth and N. Ghani. Monads and modular term rewriting. In *Category Theory in Computer Science CTCS’97*, number 1290 in LNCS, pages 69–86. Springer Verlag, September 1997.
- [18] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer Verlag, 1971.
- [19] M. Makkai and R. Paré. *Accessible Categories: The Foundations of Categorical Model Theory*, volume 104 of *Contemporary Mathematics*. American Mathematical Society, 1989.
- [20] L. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.
- [21] E. Robinson. Variations on algebra: monadicity and generalisations of equational theories. Technical Report 6/94, Sussex Computer Science Technical Report, 1994.
- [22] G. Rosu. Equational axiomatisability for coalgebra. *Theoretical Computer Science*, 260(1), 2000.
- [23] D. Turi. *Functorial Operational Semantics and its Denotational Dual*. PhD thesis, Free University, Amsterdam, 1996.
- [24] J. Worrell. Terminal sequences for accessible endofunctors. In *Electronic Notes in Theoretical Computer Science*, volume 19, 1999. Proceedings CMCS ’99.