Theoretical Informatics and Applications Will be set by the publisher
Informatique Théorique et Applications

SOLVING ALGEBRAIC EQUATIONS USING COALGEBRA

Federico De Marchi[1], Neil Ghani and Christoph Lüth[2]

Abstract. Algebraic systems of equations define functions using recursion where parameter passing is permitted. This generalises the notion of a rational system of equations where parameter passing is prohibited. It has been known for some time that algebraic systems in *Greibach Normal Form* have unique solutions.

This paper presents a categorical approach to algebraic systems of equations which generalises the traditional approach in two ways i) we define algebraic equations for locally finitely presentable categories rather than just Set; and ii) we define algebraic equations to allow right-hand sides which need not consist of finite terms. We show these generalised algebraic systems of equations have unique solutions by replacing the traditional metric-theoretic arguments with coalgebraic arguments.

AMS Subject Classification. — Give AMS classification codes —.

1. Introduction

Recursion is a fundamental concept in the theory of computation, eg in functional programming languages like Haskell etc, programs are written as recursive equations. In this paper, we will consider recursion in an algebraic setting. More precisely, given a signature, how can we *define* unknowns by a system of recursive equations, what kind of solution will these recursive equations have, and under what conditions?

[1] Mathematics and Computer Science, University of Leicester; `{fdm2,ng13}@mcs.le.ac.uk`
[2] FB 3 — Mathematics and Computer Science, Universität Bremen; `cxl@tzi.de`

If Σ is a signature, a Σ-algebraic system of equations is of the form

$$\begin{array}{rcl} \phi_1(x_1,\ldots,x_{n_1}) & = & t_1(x_1,\ldots,x_{n_1}) \\ \phi_2(x_1,\ldots,x_{n_2}) & = & t_2(x_1,\ldots,x_{n_2}) \\ & \vdots & \\ \phi_m(x_1,\ldots,x_{n_m}) & = & t_m(x_1,\ldots,x_{n_m}) \end{array} \tag{1}$$

where the unknowns $\phi_1,\ldots,\phi_m$ form a signature Ω and the terms $t_1,\ldots,t_m$ are finite terms built from the signature $\Sigma \cup \Omega$ over a countable set of variables X to which all the x_{i_j} belong. Note that nested recursion is permitted, that is to say, parameters to a recursive call are allowed to contain another recursive call, as in the equation $\psi(x) = \mathsf{A}(x, \psi(\mathsf{B}(\psi(x))))$.

The name *algebraic system* is taken from Courcelle [6]. These equations are also called *recursive program schemes* by Guessarian [10] and are also referred to as second order substitution since they equate term constructors with terms rather than simply variables with terms. In these works, it is proved that algebraic systems in *Greibach Normal Form*, ie where the right-hand sides of the equations all start with a Σ-constructor, have unique solutions in the complete metric space of infinite trees over Σ. The proof uses a least fixed point construction on this space with the distance between two trees being 2^{-n} where n is the lowest depth of a difference between the trees. An algebraic system defines a map, sending an m-tuple of terms to the m-tuple consisting of the right-hand sides of the equations with the recursive calls replaced by the original m-tuple. The Greibach Normal Form condition ensures the contractivity of the map and hence the existence of its fixed point.

As an example, consider a signature Σ consisting of a binary symbol A and a unary symbol B, and let Ω contain only a unary symbol ϕ. Then the Σ-algebraic equation $\phi(x) = \mathsf{A}(x, \phi(\mathsf{B}x))$ has as solution the infinite Σ-term $\phi^\dagger$ given below

$$\begin{tikzcd}[column sep=small] & \mathsf{A} \arrow[dl] \arrow[dr] & & & \\ x & & \mathsf{A} \arrow[dl] \arrow[dr] & & \\ & \mathsf{B}x & & \mathsf{A} \arrow[dl] \arrow[dr] & \\ & & \mathsf{BB}x & & \ldots \end{tikzcd} \tag{2}$$

To see this, start with $\mathsf{A}(x, \phi(\mathsf{B}x))$ and replace $\phi(x)$ with x to obtain $\mathsf{A}(x, \mathsf{B}(x))$. Next take the term $\mathsf{A}(x, \phi(\mathsf{B}x))$ and replace $\phi(x)$ with $\mathsf{A}(x, \mathsf{B}(x))$ thus obtaining the term $\mathsf{A}(x, \mathsf{A}(\mathsf{B}(x), \mathsf{B}(\mathsf{B}x)))$. By iterating the construction, we get a Cauchy sequence, whose limit is $\phi^\dagger$. This process — repeatedly substituting a function symbol $\phi(x)$ by the right-hand side of its defining equation — is precisely the operational semantics of a functional programming language. Viewed denotationally, this is nothing else but approximation to a fixpoint. The solutions of algebraic systems of equations are called *algebraic terms*. *Rational terms* [5, 6] are intuitively those algebraic terms which arise as solutions of systems over a signature Ω consisting of only constant symbols or, equivalently, as solutions to recursive systems where no parameter passing occurs, eg $\psi = \mathsf{A}(x, \psi)$.

Recently Moss [18], Adamek et al. [2, 16] and ourselves [8] have been looking at categorical, specifically coalgebraic, approaches to recursion with the aim of generalising standard concepts such as rational and algebraic terms and to a wider setting. Replacing the concrete representation of the topology on infinite trees given by the tree metric with the more abstract universal property of final coalgebras has considerably simplified reasoning and, unlike metric-based arguments, generalised smoothly to other categories.

For some time is has been known that infinite terms over a specific set of variables form a final coalgebra in much the same way that finite terms form an initial algebra. In order to define a substitution for infinite terms, [2, 9, 18] showed that if F is an endofunctor then the map sending each object X to the (underlying object of the) final $X + F$-coalgebra defines a monad T_F^ν. Adamek et al. [1] further characterised this monad abstractly as the free completely iterative monad over F. Moving on from infinite terms, solution theorems for rational systems of equations have been proven using coalgebra [8, 16]. The natural setting for these results is that of a finitary endofunctor on a locally finitely presentable (lfp) category. The rational monad over an endofunctor is characterised as the free iterative monad in [4] thereby generalising [5].

Given the success of the coalgebraic approach to rational terms, it is natural to ask whether coalgebra can similarly be used to enhance our understanding of algebraic terms. This paper grew out of an extended abstract presented at FICS 2002 [7] which showed that this was indeed possible. Concretely, this paper generalises algebraic systems of equations in two orthogonal ways:

- The definition of signatures, terms and substitution above is highly correlated to working over the category Set. We show how algebraic systems of equations can be defined for lfp categories.
- Σ-algebraic systems of equations can be described as recursion over the initial Σ-algebra since the right hand side of each equation belongs to an initial $\Sigma \cup \Omega$-algebra. We generalise algebraic terms to recursion over other algebras such as infinite terms or rational terms using what we call *coalgebraic monads*.

This paper further proves a unique solution theorem covering both of these extensions and the classic solution theorems of Courcelle and Guessarian arise as a special case by taking the category to be Set and using the free monad over a signature to define the right hand sides of the equations. The generalisation to lfp categories is motivated by our desire to study rational and other forms of infinite rewriting [14] while the generalisation to coalgebraic monads is motivated by lazy functional languages where recursive definitions over infinite data structures are permitted.

Although several groups, including ourselves, have tried to reduce algebraic systems of equations to rational equations in a functor category this has so far proved impossible. Thus our results do not follow from the solution theorems for rational equations cited above. Moss also has an unpublished approach [17] to solving algebraic equations using coalgebra although his approach is more concrete, limited to the category Set and only deals with equations whose right hand sides are finite terms. At a technical level, our use of coproducts of monads gives us an elegant construction of the solution of an algebraic system. Since the first publication of this work, Milius [15] has pointed out to us in an unpublished manuscript that by working in a certain comma category many of our constructions can be explained in more abstract terms. We shall summarise his comments in the running text. We are grateful to both Moss and Milius for sharing their ideas with us.

The rest of this paper will be structured as follows: in Section 2, we will generalise algebraic equations to locally finitely presentable categories. In Section 3, we introduce the notion of a *coalgebraic monad* which allows algebraic equations to have right hand sides which are not necessarily finite terms. The main technical part of the paper is Section 4, in which we show that every such generalised algebraic system has a unique solution. We conclude with a summary and our future research plans in this area.

2. Algebraic Equations in LFP Categories

In this section, we take the traditional definition of algebraic equations and transform it into an equivalent one in terms of monads on Set. We then abstract this monadic version to get a generalised form of algebraic equation in terms of monads over a locally finitely presentable category.

First, in order to treat Courcelle's algebraic equations categorically, we need categorical analogues of signatures, finite terms, infinite terms. A signature can be given as a map $\Sigma\colon \mathbb{N} \longrightarrow \mathsf{Set}$, mapping each number to the set of operations of that arity. The term algebra for the signature Σ may then be constructed as follows. Firstly, from Σ one constructs an associated functor $F_\Sigma\colon \mathsf{Set} \longrightarrow \mathsf{Set}$ which can be thought of as mapping a set of variables X to the set $F_\Sigma X$ of Σ-terms of depth 1.

$$F_\Sigma(X) = \coprod_{f \in \Sigma(n)} X^n, \tag{3}$$

Secondly, note that the functor $X + F_\Sigma$ (where X here refers to the constantly valued X functor) gives us the set of terms of depth 1 or 0. Thus, by iterating this process we obtain the set of all terms built over a set X of variables. Formally, the set of all finite terms built over a set X of variables from a signature Σ is the initial $X + F_\Sigma$-algebra. In addition, the mapping sending X to (the underlying object of) the initial $X + F_\Sigma$-algebra defines the *free monad* T_Σ over F_Σ. A similar argument gives an abstract characterisation of the set of finite and infinite terms: if instead of taking the initial $X + F_\Sigma$-algebra, we take the *final coalgebra* of $X + F_\Sigma$, we get the set $T^\nu_\Sigma(X)$ of all finite and *infinite* terms. The mapping of X to $T^\nu_\Sigma(X)$ again forms a monad meaning we have an abstract way of talking about variables and substitution.

Now recall from the introduction that we can think of an algebraic system as a morphism mapping the unknowns to their matching terms in the equation. Thinking of $\mathbb{N}$ as a discrete category, signatures become functors, and morphisms between them natural transformations. Then, a Σ-algebraic system of equations is a natural transformation from a signature Ω (regarded as a functor) to $T_{\Sigma+\Omega}$. A term being in Greibach Normal Form means that its root is a symbol from Σ, that is, the term is actually an element of $F_\Sigma T_{\Sigma+\Omega} X$. Since our real interest lies in systems in Greibach Normal Form (and the generalisation of this condition to lfp categories), we henceforth only consider such systems and just call them Σ-algebraic systems. Thus an algebraic equation is a natural transformation

$$\Omega \longrightarrow (1 + F_\Sigma T_{\Sigma+\Omega}) \circ J \tag{4}$$

where Ω is some signature. Actually, we have allowed the mild generalisation in that the right-hand side of an equation may be a variable (the 1+ part). This actually makes the mathematics easier and hence the inclusion. The inclusion $J\colon \mathbb{N} \longrightarrow \mathsf{Set}$ simply restricts the domain of the functor $1 + F_\Sigma T_{\Sigma+\Omega}$ to agree with that of Ω.

We have reformulated the definition of a Σ-algebraic equation in this way since it now becomes possible to abstract from the category Set to any *locally finitely presentable* category $\mathbf{C}$ [3] while keeping the traditional algebraic equations as special cases. We follow [12] in making the following definitions:

Definition 2.1. Let $\mathbf{C}$ be an lfp category. A signature for $\mathbf{C}$ is a functor $\Sigma\colon \mathcal{N} \longrightarrow \mathbf{C}$ where $\mathcal{N}$ is the discrete subcategory formed by a set of representatives, up to isomorphism, of the finitely presentable objects of $\mathbf{C}$. The endofunctor F_Σ is the left Kan extension of Σ along the inclusion $J\colon \mathcal{N} \longrightarrow \mathbf{C}$, i.e. $F_\Sigma = \mathsf{Lan}_J \Sigma$

We will be interested in two monads that may be constructed from a functor

Lemma 2.2. *Let* $F: \mathbf{C} \longrightarrow \mathbf{C}$ *be a functor. Then*

- *For each object* X*, assume the initial* $X + F$*-algebra exists (where* X *is regarded as the constantly valued* X*-functor). Then the map sending each object* X *to the underlying object of the initial* $X + F$*-algebra defines a monad* T_F*.*
- *For each object* X*, assume the final* $X + F$*-coalgebra exists. Then the map sending each object* X *to the underlying object of the final* $X + F$*-coalgebra defines a monad* T_F^ν*.*

Proof. The first half of the lemma is folklore [11]. For the second, see [18]. □

As mentioned above T_F is the free monad on F and T_F^ν is the free completely iterative monad on F [1]. If the functor F arises from a signature via Definition (2.1), we denote the associated monads T_Σ and T_Σ^ν. Since F_Σ is always finitary, T_Σ always exists while [?,?] provides mild conditions on an lfp category which guarantee the existence of T_Σ^ν. In the rest of this paper we shall assume T_Σ^ν does indeed exist.

Equation 4 can now be taken as the definition of a Σ-algebraic system in any lfp category and will allow us to develop a theory of algebraic (and rational) terms within such categories. Applications are many-sorted theories where one uses Set^n, term rewriting where one uses Pre, categories with structure where one uses Cat, and others [13,20].

This achieves our first goal of defining algebraic equations over categories other than Set. Now, we would like to allow the right hand sides of equations to come from other terms algebras. The key technical idea is to replace the natural transformation of 4 with an equivalent monad morphism. That T_Σ is free means that if S is a monad on the same category, there is a bijection between monad morphisms $T_\Sigma \longrightarrow S$ and signature morphisms $\Sigma \longrightarrow S \circ J$ [12]. Under this adjunction, a Σ-algebraic system is given by a signature Ω and a monad morphism

$$T_\Omega \longrightarrow 1 + F_\Sigma T_{\Sigma+\Omega} \tag{5}$$

That $1 + F_\Sigma T_{\Sigma+\Omega}$ is a monad follows from Lemma 3.1 below. Since coproducts are preserved by free constructions (ie left adjoints), $T_{\Sigma+\Omega} = T_\Sigma \oplus T_\Omega$ where $\oplus$ is the coproduct in the category of finitary monads. From [12], the coproduct of two finitary monads on an lfp category always exists and we will denote it $\oplus$ to distinguish it from the coproduct in the functor category which we denote by the usual $+$. It turns out that T_Ω can be generalised to any monad E whatsoever, yielding a partial generalisation of a Σ-algebraic system as a monad morphism

$$E \longrightarrow 1 + F_\Sigma(T_\Sigma \oplus E) \tag{6}$$

We cannot, however, generalise T_Σ arbitrarily since the solution of a Σ-algebraic equation must be in the set of finite and infinite Σ terms. The notion of a *coalgebraic monad* we introduced recently [8, 14] provides the right generalisation.

3. Coalgebraic Monads

Given an endofunctor F on a category $\mathbf{C}$, we can think of an F-coalgebraic monad on $\mathbf{C}$ as a functor building, for each object X, some class of F-terms over variables in X. The canonical examples are the set of finite terms and the set of finite and infinite terms. However, in order to formulate a precise definition, we need the following lemma, whose proof can be found in [8, 14].

Lemma 3.1. *Let (T, η, μ) be a monad on $\mathbf{C}$ and F an endofunctor on $\mathbf{C}$. Let $\tau: F \longrightarrow T$ be a natural transformation, and let $\alpha = \mu.\tau_T$. Define $\overline{\eta} = \mathsf{inl}: 1 \longrightarrow 1 + FT$, where 1 is the identity endofunctor, and*

$$\overline{\mu}: 1 + FT + FT(1 + FT) \xrightarrow{1+FT+FT\alpha} 1 + FT + FT^2 \xrightarrow{[1+FT, F\mu]} 1 + FT.$$

Then, $(1+FT, \overline{\eta}, \overline{\mu})$ is a monad, and $\beta = [\eta, \alpha]: 1+FT \to T$ is a monad morphism.

A coalgebraic monad is precisely a monad for which the morphism β described above is an isomorphism.

Definition 3.2 (Coalgebraic Monad)**.** Let F be an endofunctor on a category $\mathbf{C}$. An *F-coalgebraic monad* on $\mathbf{C}$ is a 4-tuple (T, η, μ, τ) such that (T, η, μ) is a monad on $\mathbf{C}$ and τ is a natural transformation between F and T for which the monad morphism $[\eta, \mu.\tau T]: 1 + FT \longrightarrow T$ is an isomorphism.

A *morphism* between two F-coalgebraic monads (T, η, μ, τ) and (T', η', μ', τ') is a monad morphism ϕ between T and T' such that $\phi\tau = \tau'$.

We now give some canonical examples of coalgebraic monads

Example 3.3. If $\Sigma: \mathcal{N} \longrightarrow \mathbf{C}$ is a signature on an lfp category $\mathbf{C}$, then the free monad T_Σ on F_Σ is F_Σ-coalgebraic. Similarly, the monad T_Σ^ν is also F_Σ-coalgebraic. These results hold since T_Σ and T_Σ^ν are respectively the initial and final coalgebras of the functor $1 + F_\Sigma \circ -: [\mathbf{C}, \mathbf{C}] \longrightarrow [\mathbf{C}, \mathbf{C}]$ and such structure maps are always isomorphisms. This is a variable free formulation of Lemma 2.2 — see [1]

Actually, more is true. For any finitary endofunctor F on a locally finitary presentable category, the free monad T_F and the free completely iterative monad T_F^ν are the initial and the final F-coalgebraic monads, respectively [8, 14]. Given any other F-coalgebraic monad H, we will write $!_H\colon H \longrightarrow T_F^\nu$ for the unique mediating morphism. Another example of a coalgebraic monad is the monad of rational terms. Again see [8]. Finally, we conjecture that the collection of algebraic terms forms a coalgebraic monad.

We are now in a position to reach a suitably abstract definition of algebraic systems, by replacing T_Σ with an F-coalgebraic monad H in the map (6).

Definition 3.4 (Algebraic System)**.** Let H be an F-coalgebraic monad over an arbitrary category $\mathbf{C}$. An *algebraic system* over H consists of a monad E and a monad morphism $e\colon E \longrightarrow 1 + F(H \oplus E)$. A *solution* for e is a monad morphism $e^\dagger\colon E \longrightarrow T_F^\nu$ making the following commute:

$$\begin{tikzcd} E \arrow[r, "e^\dagger"] \arrow[d, "e"'] & {T^\nu}_F \arrow[d, "\cong"'] \\ 1 + F(H + E) \arrow[r, "{1+F[!_H, e^\dagger]}"'] & 1 + F{T^\nu}_F. \end{tikzcd} \tag{7}$$

In the more concrete language of sets and signatures, a solution is a mapping of the unknowns to finite or infinite terms, such that for every unknown the value of this map is the same as replacing the unknowns in the right-hand side of the equation — the latter corresponds to going around the left and lower side of diagram (7).

Notice that we achieved a very high level of generality which at the same time is guaranteed to cover many examples since we can instantiate Definition (3.4) with, for example, the free monad over a signature on an lfp category. Thus instead of being able to define recursive functions using algebraic systems, our work can be thought of as allowing the definition of recursive morphisms.

4. A Solution Theorem

We first need to introduce a couple of technical lemmas, whose purpose is to ensure that we can perform the operations we will need in the proof of the solution theorem. We consistently use the notations of Lemma 3.1.

Lemma 4.1. *Let (H, η, μ) and (K, η', μ') be two monads, and let $\tau\colon F \longrightarrow H$ and $\tau'\colon F \longrightarrow K$ be two natural transformations. Let $\psi\colon H \longrightarrow K$ be a monad morphism such that $\psi.\tau = \tau'$. Then ψ induces a monad morphism $1 + F\psi$ between the monads $(1+FH, \overline{\eta}, \overline{\mu})$ and $(1+FK, \overline{\eta'}, \overline{\mu'})$. Moreover, ψ is a $(1+F\circ -)$-algebra morphism.*

Proof. It is trivial to observe that $\psi.\overline{\eta} = \overline{\eta'}$, because $\psi.\eta = \eta'$. The fact that multiplication is preserved is proven by the following diagram

$$\begin{tikzcd}
1+FH+FH(1+FH) \arrow[r, "(1+F\psi)^2"] \arrow[d, "1+FH+FH(1+\mu\tau_H)"] & 1+FK+FK(1+FK) \arrow[d, "1+FK+FK(1+\mu'\tau'_K)"] \\
1+FH+FH(1+H) \arrow[r, "1+F\psi+F\psi_{(1+\psi)}"] \arrow[d, "1+FH+FH[\eta,H]"] & 1+FK+FK(1+K) \arrow[d, "1+FK+FK[\eta',K]"] \\
1+FH+FH^2 \arrow[r, "1+F\psi+F\psi^2"'] \arrow[d, "1+[FH,F\mu]"] & 1+FK+FK^2 \arrow[d, "1+[FK,F\mu']"] \\
1+FH \arrow[r, "1+F\psi"'] & 1+FK
\end{tikzcd}$$

which commutes because, by assumption, ψ is a monad morphism and $\psi.\tau = \tau'$. This shows that $1 + F\psi$ is a monad morphism.

To show that ψ is an algebra morphism amounts to show that

$$[\eta', \mu'.\tau'_K].(1 + F\psi) = \psi.[\eta, \mu.\tau_H],$$

which splits along its two components as $\eta' = \psi.\eta$ and $\mu'.\tau'_K.F\psi = \psi.\mu.\tau_H$. The first holds because ψ is a monad morphism, whereas the second holds again by the properties of ψ:

$$\mu'.\tau'_K.F\psi = \mu'.K\psi.\tau'_H = \mu'.K\psi.\psi_H.\tau_H = \mu'.\psi^2.\tau_H = \psi.\mu.\tau_H.$$

□

Milius has pointed out to us [15] that Lemmas 3.1 and 4.1 can be expressed more abstractly as follows. Given a functor $F\colon \mathbf{C} \longrightarrow \mathbf{C}$, form the comma category $F/\mathrm{Mon}(\mathbf{C})$ whose object consists of monads T equipped with natural transformations $\tau\colon F \longrightarrow T$ and whose maps $\psi\colon(\tau, T) \longrightarrow (\tau', T')$ are monad morphisms $\psi\colon T \longrightarrow T'$ such that $\psi.\tau = \tau'$. Then Lemmas 3.1 and 4.1 amount to defining a copointed endofunctor $\mathcal{F}\colon F/\mathrm{Mon}(\mathbf{C}) \longrightarrow F/\mathrm{Mon}(\mathbf{C})$.

Furthermore, Milius points out that our next lemma states that T_F^ν, which is the final $(1 + F \circ -)$-coalgebra, is also the final $\mathcal{F}$-coalgebra.

Lemma 4.2. *Let $T^\nu X$ be the final $X+F$-coalgebra (therefore T^ν is the final $(1+F\circ -)$-coalgebra, and it is a monad [2]). Let H be a monad and $\tau\colon F \longrightarrow H$ a natural transformation. Suppose also that there is a monad morphism $\gamma\colon H \longrightarrow 1+FH$ such that $\gamma.\tau = \mathsf{inr}.F\eta\colon F \longrightarrow 1+FH$, making H a $(1+F\circ -)$-coalgebra. Then, the unique morphism σ to the final such coalgebra T^ν is also a monad morphism.*

$$
\begin{tikzcd}
H \arrow[r, "\sigma"] \arrow[d, "\gamma"'] & T^\nu \arrow[d, "i"] \\
1+FH \arrow[r, "1+F\sigma"'] & 1+FT^\nu
\end{tikzcd}
\tag{8}
$$

Proof. Let us call i the final coalgebra structure from T^ν to $1+FT^\nu$, which, because of Lambek's Lemma, is an isomorphism, with inverse i^{-1}. Using the same notation as in Lemma 3.1, we can depict the situation in the following diagram

$$
\begin{tikzcd}
 & H \arrow[r, "\sigma"] \arrow[d, shift left, "\gamma"] & T^\nu \arrow[d, shift left, "i"] \\
1 \arrow[ur, "\eta"] \arrow[dr, "\mathsf{inl}"'] & & \\
 & 1+FH \arrow[r, "1+F\sigma"'] \arrow[uu, shift left, "\beta"] & 1+FT^\nu \arrow[uu, shift left, "i^{-1}"]
\end{tikzcd}
\tag{9}
$$

where the two leftmost triangles commute, as well as the square involving γ. Let us write η^ν and μ^ν for the unit and multiplication of the monad T^ν. Then, all we need to prove is that $\sigma\eta = \eta^\nu$ and $\sigma\mu = \sigma^2\mu^\nu$. The first equality follows because

$$\sigma\eta = \sigma\beta\mathsf{inl} = i^{-1}i\sigma\beta\mathsf{inl} = i^{-1}(1+F\sigma)\gamma\eta = i^{-1}(1+F\sigma)\mathsf{inl} = i^{-1}\mathsf{inl} = \eta^\nu.$$

A bit more work is needed in order to prove that σ respects multiplication. In order to achieve this, we will show that both $\sigma\mu$ and $\sigma^2\mu^\nu$ are coalgebra morphism from the same $(1+F\circ -)$-coalgebra into T^ν; finality of T^ν will then prove them equal. Here is the diagram relative to $\sigma\mu$.

$$
\begin{tikzcd}
H^2 \arrow[r, "\mu"] \arrow[d, "\gamma_H"] & H \arrow[r, "\sigma"] \arrow[ddddd, "\gamma"] & T^\nu \arrow[ddddd, "i"] \\
(1+FH)H \arrow[d, "(1+FH)\gamma"] & & \\
1+FH+FH(1+FH) \arrow[d, "1+FH+FH\beta"] & & \\
1+FH+FH^2 \arrow[d, "1+{[FH\eta, FH^2]}"] \arrow[dr, "{[1+FH, F\mu]}"] & & \\
1+FH^2 \arrow[r, "1+F\mu"'] & 1+FH \arrow[r, "1+F\sigma"'] & 1+FT^\nu
\end{tikzcd}
\tag{10}
$$

Here the commutativity of the top-left cell is just the fact that γ is a monad morphism.

The diagram for $\sigma^2\mu^\nu$ is slightly more complicated.

$$
\begin{tikzcd}
H^2 \arrow[r, "H\sigma"] \arrow[d, "\gamma_H"] & HT^\nu \arrow[r, "\sigma_{T^\nu}"] \arrow[d, "\gamma_{T^\nu}"] & T^{\nu 2} \arrow[r, "\mu^\nu"] \arrow[d, "i_{T^\nu}"] & T^\nu \arrow[ddd, "i"] \\
H+FH^2 \arrow[r, "(1+FH)\sigma"] \arrow[d, "\gamma+FH^2"] & T^\nu+FHT^\nu \arrow[r, "(1+F\sigma)_{T^\nu}"] \arrow[d, "i+F\sigma_{T^\nu}"] & T^\nu+FT^{\nu 2} \arrow[d, "i+FT^{\nu 2}"] & \\
1+FH+FH^2 \arrow[r, "1+F\sigma+F\sigma^2"] \arrow[d, "1+{[FH\eta, FH^2]}"] \arrow[dr, "1+{[FH\eta^\nu, FH\sigma]}"] & 1+FT^\nu+FT^{\nu 2} \arrow[dr, "1+{[FT^\nu\eta^\nu, FT^{\nu 2}]}"] & 1+FT^\nu+FT^{\nu 2} \arrow[d, "1+{[F\eta_{T^\nu}, FT^{\nu 2}]}"] & \\
1+FH^2 \arrow[r, "1+FH\sigma"'] & 1+FHT^\nu \arrow[r, "1+F\sigma_{T^\nu}"'] & 1+FT^{\nu 2} \arrow[r, "1+F\mu^\nu"'] & 1+FT^\nu
\end{tikzcd}
\tag{11}
$$

Unfortunately, the two coalgebras are not the same, in that in the second case, after applying γ_H, one applies the identity on FH^2, whereas in the first case one applies $FH(\beta.\gamma)$. This requires one more observation to close the proof. In diagram (11) the identity on FH^2 in the middle arrow of the left side is eventually followed by the map $FH\sigma$. Now, because of the universal property of σ, if we can show that $\beta.\gamma$ is a $(1+F\circ -)$-coalgebra morphism, then we have that necessarily $\sigma = \sigma.\beta.\gamma$, because they are both the (unique) coalgebra homomorphism from H to T^{ν}, therefore we can safely substitute the coalgebra structure in (11) by the one in (10). In other words, we want to show that

$$\gamma.\beta.\gamma = (1+F\beta).(1+F\gamma).\gamma$$

which follows by commutativity of the following diagram, when precomposing the two outermost paths with γ.

$$\begin{tikzcd}
1+FH \arrow[rrr, "1+F\gamma"] \arrow[ddd, "\beta"'] \arrow[dr, "1+\tau_H"] & & & 1+F(1+FH) \arrow[dl, "{[\overline{\eta}\mathsf{inl},\mathsf{inr}F\eta_{(1+FH)}]}"'] \arrow[ddd, "1+F\beta"] \\
& 1+H^2 \arrow[r, "{[\overline{\eta}\mathsf{inl},\gamma^2]}"'] \arrow[ddl, "{[\eta,\mu]}"] & (1+FH)^2 \arrow[ddr, "\overline{\mu}"'] & \\
& & & \\
H \arrow[rrr, "\gamma"'] & & & 1+FH
\end{tikzcd}$$

Here the left triangle is just the definition of β; the bottom square commutes because γ is a monad morphism; the right triangle commutes because of how $\overline{\mu}$ is defined, and, finally, the top square commutes because

$$(\mathsf{inr}.F\eta)_{1+FH}.1+F\gamma = (\gamma.\tau)_{1+FH}.F\gamma = \gamma_{1+FH}.H\gamma.\tau_H = \gamma^2.\tau_H.$$

□

We can now state the solution theorem and give its proof.

Theorem 4.3 (Solution Theorem). *If $e\colon E \longrightarrow 1+F(H+E)$ is an algebraic system over a coalgebraic monad (H,η,μ,τ), then e has a unique solution.*

Proof. By composing τ with the injection of H into $H+E$, one gets a natural transformation $\tau'\colon F \longrightarrow H+E$, which by Lemma 3.1 induces a monad structure on $1+F(H+E)$. Moreover, the equality $\tau' = \mathsf{inl}.\tau$ is satisfied by construction of τ', therefore, because of Lemma 4.1, there is also a monad morphism $1+F\mathsf{inl}\colon 1+FH \longrightarrow 1+F(H+E)$, which, being H coalgebraic, leads to a monad morphism $\delta\colon H \longrightarrow 1+F(H+E)$ (just precompose $1+F\mathsf{inl}$ with the isomorphism between H and $1+FH$). Copairing δ with the equation morphism e gives a monad morphism $[\delta,e]\colon H+E \longrightarrow 1+F(H+E)$, which endows $H+E$ with a $(1+F\circ -)$-coalgebra structure. We therefore have a coalgebra morphism to the final coalgebra

$\beta: H + E \longrightarrow T^\nu$, and we want to use Lemma 4.2 to show that it is a monad morphism. Precomposition with the second injection into the coproduct will then be our candidate solution morphism $e^\dagger: E \longrightarrow T^\nu$. In order to apply the lemma (where the H is now replaced by $H+E$), we have to show that $[\gamma, e].\tau' = \mathsf{inr}.F\eta'$, where η' is the unit of $H+E$. But, since the coproduct is in the category of monads, one has that the left injection is a monad morphism, hence $\eta' = \mathsf{inl}.\eta$. From this and the fact that H is coalgebraic, we get that

$$[\gamma, e].\tau' = [\gamma, e].\mathsf{inl}.\tau = (1 + F\mathsf{inl}).\mathsf{inr}'.F\eta = \mathsf{inr}.F\mathsf{inl}.F\eta = \mathsf{inr}.F\eta',$$

where inr' is the injection of FH in $1 + FH$, inl maps H to $H+E$, and inr is the inclusion of $F(H+E)$ in $1 + F(H+E)$.

All we have to do now is to show that diagram (7) commutes, but this trivially follows by looking at the second component of the commuting diagram (8), where H is now replaced by $H+E$.

Now suppose $d: E \longrightarrow T^\nu$ is another solution of the algebraic system. By copairing it with the mediating morphism $!_H$, one gets a $(1 + F \circ -)$-coalgebra morphism from $H+E$ to T^ν, which is therefore the same morphism as γ. By precomposing with the right injection, now, one gets that $d = e^\dagger$, thus showing the uniqueness of the solution morphism. □

5. Conclusion and Further Work

In this paper, we have introduced a coalgebraic treatment of algebraic equations which generalised the standard treatment of algebraic systems in two directions. Firstly, we defined algebraic systems for arbitrary lfp categories. Secondly, we allowed the right hand sides of equations to be not just be a finite term, but to be given by a coalgebraic monad. This includes having infinite terms or rational terms as the right hand side of an equation. In the main section of the paper, we have shown that every such generalised system of algebraic equations has a unique solution.

An open question at this point is whether algebraic terms form a monad and whether this monad has a universal property, thereby extending the analogous results for rational terms [4]. Since we can encode algebraic terms as rational terms with explicit substitution operators which can be given by first-order operations and equations (which can be given by monad [19]), we conjecture it holds for algebraic terms as well.

Infinite rewriting aims to identify a well behaved class of infinite sequences of rewrites. On the one hand, we want to accept as many infinite rewrite sequences as possible so as to avoid being over-prescriptive. On the other hand, over-generality leads to a bad meta-theory. *Rational rewriting* considers those infinite terms which are rational terms and allows rewriting on them by allowing recursively defined rewrites. For example, let C be a binary symbol, $\psi_1^\dagger$ be the solution of $\psi_1 = C(A, \psi_1)$ and $\psi_2^\dagger$ be the solution of $\psi_2 = C(B, \psi_2)$. Then if $r: A \longrightarrow B$ is a finite rewrite, the equation $r' = C(r, r')$ would have as solution the infinite rewrite $\psi_1^\dagger \longrightarrow \psi_2^\dagger$ which performs an r rewrite on the left hand branch under each node labelled C. Categorically, defining not just infinite terms but also infinite rewrites between them means a shift from the category Set to the category **Pre** in the vein of [13]. However, this rational rewriting is over-prescriptive in that the canonical motivation for infinite rewriting is the implementation of lazy functional languages where recursion involving parameter passing is fundamental. Thus there are strong motivations for generalising *rational rewriting* to *algebraic rewriting* by defining infinite rewrites recursively using parameter passing.

More speculatively, we are interested in a coalgebraic treatment of recursively defined geometric objects such as Sierpinskis triangle, Mandelbrot sets and other fractal like objects. While the recursive nature of these objects is clear, it is currently still unclear to us as to exactly which category one should be working in. Once this is understood, this would be a very exciting application.

References

[1] P. Aczel, J. Admek, S. Milius, and J. Velebil. Infinite trees and completely iterative theories: A coalgebraic view. *Theoretical Computer Science*, 300:1–45, 2003.

[2] Peter Aczel, Jiří Adámek, and Jiří Velebil. A coalgebraic view of infinite trees and iteration. In Andrea Corradini, Marina Lenisa, and Ugo Montanari, editors, *Proceedings 4th Workshop on Coalgebraic Methods in Computer Science, CMCS'01, Genova, Italy, 6–7 Apr. 2001*, volume 44(1) of *Electronic Notes in Theoretical Computer Science*, Amsterdam, 2001. Elsevier.

[3] Jiří Adámek. Final coalgebras are ideal completions of initial algebras. *Journal of Logic and Computation*, 12:217–242, 2002.

[4] Jiří Adámek and Jiří Rosicky. *Locally Presentable and Accessible Categories*, volume 189 of *London Mathematical Society Lecture Notes*. Cambridge University Press, 1994.

[5] J. Admek, S. Milius, and J. Velebil. Free iterative theories: A coalgebraic view. *Mathematical Structures in Computer Science*, 13:259–320, 2003.

[6] Michael Barr. Terminal coalgebras for endofunctors on sets. available from `ftp://www.math.mcgill.ca/pub/barr/trmclgps.zip`, 1999.

[7] C.C.Elgot, S.L.Bloom, and S.Tindell. On the algebraic structure of rooted trees. *J. Comp. Syst. Sciences*, 16:361–399, 1978.

[8] Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25(2):95–169, 1983.

[9] Neil Ghani, Christoph Lüth, and Federico De Marchi. Coalgebraic approaches to algebraic terms. In Zoltán Ésik and Anna Ingólfsdóttir, editors, *Fixed Points in Computer Science*, volume NS-02-2 of *BRICS Notes Series*, pages 6–8, July 20–21 2002.

[10] Neil Ghani, Christoph Lüth, and Federico De Marchi. Coalgebraic monads. In Lawrence M. Moss, editor, *Proceedings 5th Workshop on Coalgebraic Methods in Computer Science, CMCS'02, Grenoble, France, 6–7 Apr. 2002*, 2002.

[11] Neil Ghani, Christoph Lüth, Federico De Marchi, and John Power. Algebras, coalgebras, monads and comonads. In Andrea Corradini, Marina Lenisa, and Ugo Montanari, editors, *Proceedings 4th Workshop on Coalgebraic Methods in Computer Science, CMCS'01, Genova, Italy, 6–7 Apr. 2001*, volume 44(1) of *Electronic Notes in Theoretical Computer Science*. Elsevier, Amsterdam, 2001.

[12] I. Guessarian. *Algebraic Semantics*, volume 99 of *Lecture Notes in Computer Science*. Springer-Verlag, 1979.

[13] G. M. Kelly. A unified treatment of transfinite constructions. *Bull. of Austral. Math. Soc.*, 22:1– 83, 1980.

[14] G. M. Kelly and A. J. Power. Adjunctions whose counits are equalizers, and presentations of finitary monads. *Journal of Pure and Applied Algebra*, 89:163– 179, 1993.

[15] Christoph Lüth and Neil Ghani. Monads and modular term rewriting. In E. Moggi and G. Rosolini, editors, *Proceedings 7th Int. Conf. on Category Theory and Computer Science, CTCS'97, Santa Margherita Ligure, Italy, 4–6 Sept. 1997*, volume 1290 of *Lecture Notes in Computer Science*, pages 69–86. Springer-Verlag, Berlin, 1997.

[16] F. De Marchi. *Monads in Coalgebra*. PhD thesis, Univ. of Leicester, 2003. Submitted.

[17] S. Milius. Final coalgebras in categories of monads. Unpublished.

[18] Stefan Milius. Free iterative theories: a coalgebraic view (extended abstract). presented at FICS 2001 - Fixed Points in Computer Science, 7-8 September 2001, Florence, Italy.

[19] Lawrence Moss. The coalgebraic treatment of second-order substitution and uniinterpreted recursive program schemes. privately circulated manuscript.

[20] Lawrence Moss. Parametric corecursion. preprint, available at `http://math.indiana.edu/home/moss/parametric.ps`.

[21] N.Ghani and T.Uustalu. Explicit substitutions and presheafs. In *Proceedings of MERLIN 2003*. 2003.

[22] Edmund Robinson. Variations on algebra: monadicity and generalisation of equational theories. Technical Report 6/94, Sussex Computer Science, 1994.

Communicated by (The editor will be set by the publisher).
(The dates will be set by the publisher).