Distributive laws for the Coinductive Solution of Recursive Equations

Bart Jacobs

Institute for Computing and Information Sciences,
Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
Email: `B.Jacobs@cs.ru.nl`   *URL:* `http://www.cs.ru.nl/B.Jacobs`

Abstract

This paper illustrates the relevance of distributive laws for the solution of recursive equations, and shows that one approach for obtaining coinductive solutions of equations via infinite terms is in fact a special case of a more general approach using an extended form of coinduction via distributive laws.

1 Introduction

Distribution $x(y + z) = xy + xz$ is common in many equational theories, such as vector spaces. It may also occur in so-called distributive categories, of the form $X \times (Y + Z) \cong (X \times Y) + (X \times Z)$, see *e.g.* [8], where one direction of the isomorphism is canonical and always exists. More generally, one can have distributions $GF \Rightarrow FG$ between two endofunctors F, G on the same category, as first studied in [7]. This phenomenon is especially interesting when the functors F, G form signatures (or interfaces) for certain operations, either in algebraic or in coalgebraic form.

Turi and Plotkin [25] first investigated such a situation where one functor G describes the syntax of a programming language and the other functor F the behaviour of programs (terms) in that language. Having a distributive law $GF \Rightarrow FG$ means that the behaviour on terms is well-defined, and leads to results like: (coalgebraic) bisimilarity is an (algebraic) congruence. Hence distributive laws capture where "algebra meets coalgebra".

The theme of this paper is the same, in a slightly different context, namely recursive equations $x_i = t_i(x_1, \ldots, x_n)$. The t_i are terms from some algebra, and may contain the recursive variables x_j. The solutions of such equations

are typically infinite, and are thus best described via (final) coalgebras. Hence also in this situation algebra and coalgebra meet, and appropriate distributive laws are to be expected.

The finality principle in the theory of coalgebras is usually called coinduction [15]. It involves the existence and uniqueness of suitable coalgebra homomorphisms to final coalgebras. It was realised early on (see [1,6]) that such coinductively obtained homomorphisms can be understood as solutions to recursive (or corecursive, if you like) equations. The equation itself is incorporated in the commuting square expressing that there is a homomorphism from a certain "source" coalgebra to the final coalgebra. Since this diagram arises from the source coalgebra, this source can also be identified with the recursive equation (see Section 3 for examples).

A systematic investigation of the solution of such equations first appeared in [20], followed by [2]. Their coalgebraic approach simplifies results for recursive equations with infinite terms from [10,11]. More recently, a general and abstract approach is proposed in [5], using distributive laws. It builds on earlier work [17] and may also be described dually, for algebras, as developed independently in [26]. One of the main contributions of this paper is that it shows how the approach of [2] for infinite terms fits in the general approach of [5] with distributive laws. This involves the identification of suitable distributive laws of the monads of terms over the underlying interface functor.

This paper is organised as follows. Section 2 briefly reviews the approach of [5] based on distributive laws. It is illustrated in the context of languages and automata in Section 3. Section 4 continues with two distributive laws for canonical monads F^* and F^∞ associated with a functor F. The approach of [2] for solutions of equations with infinite terms is then explained in Section 5. Finally, Section 6 shows that this approach is an instance of the distribution-based approach.

An earlier version of this paper appeared as [12]. The present version extends [12] especially with Section 3 on distributive laws for languages and automata. This topic is further elaborated in [14].

2 Distributive laws and solutions of equations

Distributive laws found their first serious application in the area of coalgebras in the work of Turi and Plotkin [25] (see also [24]), providing a joint treatment of operational and denotational semantics. In that setting a distributive law provides a suitable form of compatibility between syntax and dynamics. The claim of [25] that distributive laws correspond to suitable rule formats for

operators is further substantiated in [5]. The idea of using a distributive law in extended forms of coinduction (and hence equation solving) comes from [17], and is further developed in [5]. In this section we present its essentials.

Distributive laws are natural transformations $GF \Rightarrow FG$ between two endofunctors $F, G \colon \mathbb{C} \to \mathbb{C}$ on a category $\mathbb{C}$. These F and G may have additional structure (of a point or copoint, or a monad or comonad, see [18]), that must then be preserved by the distributive law. We shall concentrate on the case of distribution of a monad over a functor, because it seems to be most common and natural—see the examples in the next section. We shall recall what this means.

Definition 1 *Let (T, η, μ) be a monad on a category $\mathbb{C}$, and $F \colon \mathbb{C} \to \mathbb{C}$ be an arbitrary functor. A **distributive law** of T over F is a natural transformation*

$$TF \xrightarrow{\quad \lambda \quad} FT$$

making for each $X \in \mathbb{C}$ the following two diagrams commute.

$$
\begin{array}{ccc}
FX & & \\
\eta_{FX} \downarrow & \xrightarrow{F(\eta_X)} & \\
TFX & \xrightarrow{\quad \lambda_X \quad} & FTX
\end{array}
\qquad\qquad
\begin{array}{ccccc}
T^2FX & \xrightarrow{T(\lambda_X)} & TFTX & \xrightarrow{\lambda_{TX}} & FT^2X \\
\mu_{FX} \downarrow & & & & \downarrow F(\mu_X) \\
TFX & & \xrightarrow{\quad \lambda_X \quad} & & FTX
\end{array}
$$

Sometimes we shall consider the situation when F is a monad too. When λ then also preserves the unit and multiplication associated with F—in the obvious way, like above—we shall say that λ is a distributive law of monads.

The underlying idea is that the monad T describes the terms in some syntax, and that the functor F is the interface for transitions on a state space. Intuitively, the presence of the distributive law tells us that the terms and behaviours interact appropriately. The associated notion of model is a so-called λ-bialgebra.

Definition 2 *Let $\lambda \colon TF \Rightarrow FT$ be a distributive law, like above. A λ-**bialgebra** consists of an object $X \in \mathbb{C}$ with a pair of maps:*

$$TX \xrightarrow{\ a\ } X \xrightarrow{\ b\ } FX$$

where:

- *a is an Eilenberg-Moore algebra, meaning that it satisfies two standard equations, namely: $a \circ \eta_X = id$ and $a \circ \mu_X = a \circ T(a)$.*
- *a and b are compatible via λ, which means that the following diagram com-*

mutes.

$$TX \xrightarrow{a} X \xrightarrow{b} FX$$

with $T(b)$ downward on the left, $F(a)$ upward on the right, and $TFX \xrightarrow{\lambda_X} FTX$ along the bottom.

A **map of λ-bialgebras**, *from* $(TX \xrightarrow{a} X \xrightarrow{b} FX)$ *to* $(TY \xrightarrow{c} Y \xrightarrow{d} FY)$ *is a map* $f: X \to Y$ *in* $\mathbb{C}$ *that is both a map of algebras and of coalgebras:* $f \circ a = c \circ T(f)$ *and* $d \circ f = F(f) \circ b.$

The following result is standard.

Lemma 3 *Assume a distributive law* $\lambda: TF \Rightarrow FT$, *and let* $\zeta: Z \xrightarrow{\cong} FZ$ *be a final coalgebra. It carries an Eilenberg-Moore algebra obtained by finality in:*

$$
\begin{array}{ccc}
FTZ & \xdashrightarrow{\ \ F(\alpha)\ \ } & FZ \\
{\scriptstyle \lambda_Z}\big\uparrow & & \big\uparrow{\scriptstyle \cong}\,\zeta \\
TFZ & & \\
{\scriptstyle T(\zeta)}\big\uparrow{\scriptstyle \cong} & & \\
TZ & \xdashrightarrow{\ \ \ \alpha\ \ \ } & Z
\end{array}
$$

The resulting pair $(TZ \xrightarrow{\alpha} Z \xrightarrow{\zeta} FZ)$ *is then a final λ-bialgebra.*

Proof. By the uniqueness part of finality one proves that α is an Eilenberg-Moore algebra. By construction, α and ζ are compatible via λ. Assume an arbitrary λ-bialgebra $(TX \xrightarrow{a} X \xrightarrow{b} FX)$. It induces a unique coalgebra map $f: X \to Z$ with $\zeta \circ f = F(f) \circ b$. One then obtains $f \circ a = \alpha \circ T(f)$ by showing that both maps are homomorphisms from the coalgebra $\lambda_X \circ T(b): TX \to FTX$ to the final coalgebra ζ. $\qquad\square$

We shall consider some simple ways to build distributive laws.

Example 4 *Let* $T: \mathbb{C} \to \mathbb{C}$ *be a monad with unit and multiplication* $\eta, \mu.$

(1) *Let* $a: TA \to A$ *be an Eilenberg-Moore algebra. It yields a distributive law* $a: TK_A \Rightarrow K_A T$, *where* $K_A: \mathbb{C} \to \mathbb{C}$ *is the functor which is constantly A.*

(2) *Assume we have an I-indexed collection of functors* $F_i: \mathbb{C} \to \mathbb{C}$ *with distributive laws* $\lambda_i: TF_i \Rightarrow F_i T$. *Then, assuming that the product functor* $F = \prod_{i \in I} F_i$ *exists, there is a distributive law* $\lambda: TF \Rightarrow FT$ *given by*

$$\lambda_X = \left(T(\textstyle\prod_{i \in I} F_i X) \xrightarrow{\langle T(\pi_i)\rangle_{i \in I}} \prod_{i \in I} TF_i X \xrightarrow{\prod_{i \in I} \lambda_i} \prod_{i \in I} F_i TX \right)$$

Special cases worth emphasising are:

- $I = \{1, 2\}$, describing the distributive law $T(F_1 \times F_2) \Rightarrow F_1T \times F_2T$ for a binary product from [5, Lemma 4.4.5];
- each F_i is equal to G, so that F is the exponent functor G^I, with "strength" distributive law $T(G^I) \Rightarrow (GT)^I$.

(3) Dually, if T preserves coproducts, one can construct a distributive law $T(\coprod_{i \in I} F_i) \Rightarrow (\coprod_{i \in I} F_i)T$ from laws $TF_i \Rightarrow F_iT$.

(4) If our category $\mathbb{C}$ is **Sets**, and the functor T preserves weak pullbacks, then there is a distributive law of monads $T\mathcal{P} \Rightarrow \mathcal{P}T$, where $\mathcal{P}$ is the powerset monad. This construction comes from [13], and is called the "power law". Here we sketch the essentials.

We associate the so-called "relation lifting" $\mathrm{Rel}(T)$ with T. It is a functor that maps a relation $\langle r_1, r_2 \rangle : R \rightarrowtail X \times Y$ to a relation $\mathrm{Rel}(T)(R) \rightarrowtail T(X) \times T(Y)$ by taking the image of the map $\langle T(r_1), T(r_2) \rangle : T(R) \to T(X) \times T(Y)$. Applying this relation lifting to the inhabitation relation $\in_X \rightarrowtail X \times \mathcal{P}(X)$ yields $\mathrm{Rel}(T)(\in_X) \rightarrowtail TX \times T\mathcal{P}(X)$. Then we can define $\lambda_X : T\mathcal{P}(X) \to \mathcal{P}(TX)$ as:

$$\lambda_X(u) = \{a \in TX \mid \langle a, u \rangle \in \mathrm{Rel}(T)(\in_X)\}.$$

In [13] it is shown that λ preserves the powerset monad structure. But it also preserves the unit η and multiplication μ of the monad T in case the natural transformations η, μ are Cartesian. This means that their naturality squares are pullbacks.

The following notion of equation and solution comes from [5].

Definition 5 *Assume a distributive law $\lambda : TF \Rightarrow FT$. A **guarded recursive equation** is an FT-coalgebra $e : X \to FTX$. A **solution** to such an equation in a λ-bialgebra $(TY \xrightarrow{a} Y \xrightarrow{b} FY)$ is a map $f : X \to Y$ making the following diagram commute.*

$$
\begin{array}{ccc}
FTX & \xrightarrow{FT(f)} & FTY \\
\uparrow{\scriptstyle e} & & \downarrow{\scriptstyle F(a)} \\
& & FY \\
& & \uparrow{\scriptstyle b} \\
X & \xrightarrow{f} & Y
\end{array}
\qquad (1)
$$

In ordinary coinduction one obtains solutions for equations $X \to FX$. The additional expressive power of the above notion of equation $X \to FTX$ lies in the fact that it allows actions on terms. For convenience we shall often call these equations $X \to FTX$ λ**-equations**—even though their formulation does not involve a distributive law λ. But their intended use is in a context with distributive laws. Similarly, we shall say that the above solution f is defined by λ**-coinduction**.

This notion of solution may seem a bit strange at first, but becomes more natural in light of the following result (see also [5, Lemma 4.3.4]).

Proposition 6 *There exists a bijective correspondence between λ-equations $e\colon X \to FTX$ and λ-bialgebras $(T^2X \xrightarrow{\mu_X} TX \xrightarrow{d} FTX)$ with free algebra μ_X.*

Moreover, let $(TY \xrightarrow{a} Y \xrightarrow{b} FY)$ be a λ-bialgebra. Then there is a bijective correspondence between solutions $f\colon X \to Y$ as in (1) and bialgebra maps $g\colon TX \to Y$—for the associated λ-equations and λ-bialgebras. $\qquad\square$

Now we can formulate the main result of this distribution-based approach to solving equations. It is the dual of [26, Theorem 1].

Theorem 7 *Let $F\colon \mathbb{C} \to \mathbb{C}$ be a functor with a final coalgebra $Z \xrightarrow{\cong} FZ$. For each monad T with distributive law $\lambda\colon TF \Rightarrow FT$ there are unique solutions to λ-equations in the final λ-bialgebra $(TZ \to Z \to FZ)$ from Lemma 3.*

Proof. For a λ-equation $e\colon X \to FTX$, a solution in $(TZ \to Z \to FZ)$ is by the previous proposition the same thing as a map of λ-bialgebras from the associated $(T^2X \to TX \to FTX)$ to $(TZ \to Z \to FZ)$. Since the latter is final, there is precisely one such solution. $\qquad\square$

In the next section, and also in Example 13, we present illustrations.

3 Kleene algebras and differential equations for languages

This section contains two applications of distributive laws in the context of languages: first, in order to obtain a "language" monad whose algebras are Kleene algebras, and second, to describe differential equations for languages with solutions as in the previous section.

3.1 Kleene algebras

A basic observation and starting point in this subsection is that there is a "power" distributive law π in:

$$\mathcal{P}(X)^\star \xrightarrow{\quad\pi_X\quad} \mathcal{P}(X^\star)$$

$$\langle u_1, \ldots, u_n \rangle \longmapsto \{\langle x_1, \ldots, x_n \rangle \mid \forall i \leq n.\, x_i \in u_i\}$$

(2)

It is obtained from the construction in Example 4 (4), using that the list monad $(-)^\star$ is Cartesian. In order to investigate the consequence we use the following general result about distributive laws between monads. It is standard, and may be traced back to [7,16,4] or [25].

Proposition 8 *Let $\pi: ST \Rightarrow TS$ be a distributive law between monads S and T on a category $\mathbb{C}$. Then:*

(1) TS is a monad, with unit and multiplication given as:

$$
\eta = \left(
\begin{array}{c}
\xymatrix{ & S \ar[dr]^{\eta^T S} & \\ Id \ar[ur]^{\eta^S} \ar[dr]_{\eta^T} & & TS \\ & T \ar[ur]_{T\eta^S} & }
\end{array}
\right)
\qquad
\mu = \left(
\begin{array}{c}
\xymatrix{ & & T^2 S \ar[dr]^{\mu^T S} & \\ TSTS \ar[r]^{T\pi S} & T^2 S^2 \ar[ur]^{T^2\mu^S} \ar[dr]_{\mu^T S} & & TS \\ & & TS^2 \ar[ur]_{T\mu^S} & }
\end{array}
\right)
$$

Moreover, there are obvious maps of monads $S \Rightarrow TS$ and $T \Rightarrow TS$ given by units.

(2) There is an induced lifting of T to Eilenberg-Moore algebras of S as in:

$$
\xymatrix{
\mathbf{Alg}(S) \ar[r]^{\overline{T}} \ar[d] & \mathbf{Alg}(S) \ar[d] \\
\mathbb{C} \ar[r]_{T} & \mathbb{C}
}
\qquad \text{given by} \qquad
\left(\begin{array}{c} SX \\ \downarrow \\ X \end{array}\right) \longmapsto
\left(\begin{array}{c} STX \\ \downarrow \pi \\ TSX \\ \downarrow \\ TX \end{array}\right)
$$

This yields a new monad $\overline{T}$. It can be shown that there is a bijective correspondence between such liftings and distributive laws.

(3) There is an isomorphism of categories of algebras:

$$
\xymatrix{
\mathbf{Alg}(TS) \ar[rr]^{\cong} \ar[dr] & & \mathbf{Alg}(\overline{T}) \ar[dl] \\
& \mathbf{Alg}(S) \ar[d] & \\
& \mathbb{C} &
}
$$

$\square$

When we apply this result to our power law $\pi: (-)^\star \mathcal{P} \Rightarrow \mathcal{P}(-)^\star$ from (2) we obtain a new monad $\mathcal{L} = \mathcal{P}(-)^\star$ which we shall call the **language monad**. This name is chosen because the sets $\mathcal{L}(X) = \mathcal{P}(X^\star)$ contain languages $L \subseteq X^*$ with words over the alphabet X.

According to Proposition 8 (1), the unit $\eta_X: X \to \mathcal{L}(X)$ is given by

$$
\eta_X(x) = \{\langle x \rangle\}.
$$

The multiplication $\mu_X \colon \mathcal{L}^2(X) \to \mathcal{L}(X)$ maps a set $V \in \mathcal{L}^2(X) = \mathcal{P}(\mathcal{P}(X^\star)^\star)$ of sequences of languages to the language:

$$
\begin{aligned}
\mu_X(V) &= \{\overline{\langle s_1, \ldots, s_n \rangle} \mid \exists \langle L_1, \ldots, L_n \rangle \in V. \forall i \leq n. s_i \in L_i\} \\
&\quad \text{where } \overline{} \colon X^{\star\star} \to X^\star \text{ is } (-)^\star\text{'s "flattening" multiplication} \\
&= \{s_1 \cdot \ldots \cdot s_n \mid \exists \langle L_1, \ldots, L_n \rangle \in V. \forall i \leq n. s_i \in L_i\} \\
&\quad \text{where } \cdot \text{ is concatenation of sequences} \\
&= \bigcup \{L_1 \cdot \ldots \cdot L_n \mid \langle L_1, \ldots, L_n \rangle \in V\} \\
&\quad \text{where } \cdot \text{ is concatenation for sets of sequences (languages).}
\end{aligned}
$$

The next question is: what are the algebras of the language monad $\mathcal{L}$? Before answering this question we recall the well-known facts that the algebras of the $(-)^*$ monad are monoids, and that the algebras of the powerset monad $\mathcal{P}$ are complete lattices (posets in which each subset has a join). Proposition 8 (3) tells that $\mathcal{L}$-algebras are algebras of the lifted monad $\overline{\mathcal{P}}$ on the category **Mon** of monoids. The functor $\overline{\mathcal{P}}$ maps a monoid $(X, \cdot, 1)$ to the monoid $(\mathcal{P}(X), \bullet, \{1\})$, with composition operation $\bullet$ given on $u, v \in \mathcal{P}(X)$ as:

$$
u \bullet v = \{x \cdot y \mid x \in u \wedge y \in v\}.
$$

An algebra $(\mathcal{P}(X), \bullet, \{e\}) \to (X, \cdot, e)$ is thus a $\mathcal{P}$-algebra $\mathcal{P}(X) \to X$, forming a join-operation $\bigvee$, which is a homomorphism of monoids:

$$
(\bigvee u) \cdot (\bigvee v) = \bigvee u \bullet v = \bigvee \{x \cdot y \mid x \in u \wedge y \in v\}.
$$

This means that the monoid's operation $\cdot$ preserves joins in both variables separately. The next (folklore) result summarises the situation so far.

Theorem 9 *The language monad $\mathcal{L} = \mathcal{P}((-)^\star)$ induced by the "power" distributive law $(-)^\star \mathcal{P} \Rightarrow \mathcal{P}(-)^\star$ from (2) has **Kleene algebras** as Eilenberg-Moore algebras. The latter are complete lattices with a monoid structure in which joins are preserved by the monoid operation, in both variables.* $\qquad \square$

Often one sees the "finite" version of Kleene algebras with only finite joins 0 and $x + y$ satisfying distribution equations like $(x + y) \cdot z = x \cdot z + y \cdot z$ and $z \cdot (x + y) = z \cdot x + z \cdot y$ and $0 \cdot x = 0 = x \cdot 0$. In the theorem we obtain algebras with arbitrary joins, such as used in [9], under the name "standard Kleene algebras". The associated iteration operation is obtained as $x^* = \bigvee_{n \in \mathbb{N}} x^n$. Our $\mathcal{L}$-algebras are also known as unital quantales, see [22].

The set of languages $\mathcal{L}(X)$ carries a free Kleene algebra structure $\mu_X \colon \mathcal{L}^2(X) \to \mathcal{L}(X)$, with the familiar structure induced by the multiplication μ:

$$
\begin{aligned}
0 &= \mu_X(\emptyset) &&= \emptyset \\
1 &= \mu_X(\{\langle\rangle\}) &&= \{\langle\rangle\} \\
L_1 \cdot L_2 &= \mu_X(\{\langle L_1, L_2 \rangle\}) &&= \{s_1 \cdot s_2 \mid s_1 \in L_1 \wedge s_2 \in L_2\} \\
\textstyle\bigvee_{i \in I} L_i &= \mu_X(\{\langle L_i \rangle \mid i \in I\}) &&= \textstyle\bigcup_{i \in I} L_i \\
L^* &= \mu_X(\{\underbrace{\langle L, \ldots, L \rangle}_{n \text{ times}} \mid n \in \mathbb{N}\}) &&= \textstyle\bigvee_{n \in \mathbb{N}} L^n.
\end{aligned}
$$

3.2 Differential equations for languages

In the previous subsection we have seen how sets of languages $\mathcal{L}(A) = \mathcal{P}(A^*)$ form free Kleene algebras. Here we shall investigate them as (carriers of) final coalgebras. We shall do so in three stages, where the first one is well-known (and extensively studied in [23, Section 10]), and the second one comes from [5, Corollary 4.4.6]. The third one builds on the above language monad $\mathcal{L}$.

3.2.1 Languages and deterministic automata

A deterministic automaton, with alphabet A, is a coalgebra $\langle \delta, \varepsilon \rangle \colon X \to X^A \times 2$. The transition function δ maps a state together with an input to a new (next) state, and the output function ε tells of a state $x \in X$ whether x is terminal ($\varepsilon(x) = 1$) or not ($\varepsilon(x) = 0$). We shall write $D = (-)^A \times K_2$ for the functor involved. Typical for these deterministic automata is that for each state x and input letter $a \in A$ there is precisely one successor state x' with $x \xrightarrow{a} x'$, i.e. with $x' = \delta(x)(a)$.

As is well-known, the final D-coalgebra is given by the set of languages $\mathcal{L}(A) = \mathcal{P}(A^*)$ over the alphabet A, with coalgebra structure $\langle \delta, \varepsilon \rangle \colon \mathcal{L}(A) \to \mathcal{L}(A)^A \times 2$ given by the "derivative" function and "is nullable" predicate (see [9,23]): for $L \in \mathcal{L}(A)$ and $a \in A$,

$$
\begin{aligned}
\delta(L)(a) &= L_a \\
&= \{\sigma \in A^\star \mid a \cdot \sigma \in L\} \\
\varepsilon(L) &= (1 \subseteq L) \\
&= (\langle\rangle \in L).
\end{aligned}
$$

For an arbitrary D-coalgebra $X \to X^A \times 2$, the induced homomorphism to

this final coalgebra,

$$
\begin{array}{ccc}
X^A \times 2 & \dashrightarrow & \mathcal{L}(A)^A \times 2 \\
\uparrow & & \cong \uparrow \langle \delta, \varepsilon \rangle \\
X & \dashrightarrow & \mathcal{L}(A)
\end{array}
$$

sends a state $x \in X$ to the language accepted in this state, *i.e.* to the set of those strings $\langle a_1, \ldots, a_n \rangle \in A^\star$ leading from x to a terminal state.

The behaviour—or accepted languages—associated with a deterministic automaton can be described via "differential equations". For instance, the automaton:

with state 1 terminal
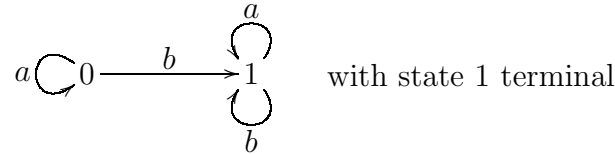
can be described by the equations:

$$
\frac{\partial L_0}{\partial a} = L_0 \quad \frac{\partial L_0}{\partial b} = L_1 \quad \langle \rangle \notin L_0 \qquad \frac{\partial L_1}{\partial a} = L_1 \quad \frac{\partial L_1}{\partial b} = L_1 \quad \langle \rangle \in L_1,
$$

where L_i is the language accepted in state i, and $\frac{\partial L}{\partial x}$ is a fancy notation for the derivative L_x, where $x \in A = \{a, b\}$. The obvious solution of these equations is $L_0 = a^\star b(a^\star b^\star)^\star$ and $L_1 = (a^\star b^\star)^\star$. It is obtained as map $L: 2 \to \mathcal{L}(A)$ by finality, using the above differential equations as description of a coalgebra $2 \to 2^A \times 2$.

By combining several clauses from Example 4 we obtain the following result from [14] describing a sufficient condition for the existence of a distributive law for deterministic automata, together with the associated final bialgebra. For the proof we refer to [14].

Theorem 10 *An Eilenberg-Moore algebra $\beta: T(2) \to 2$ for a monad T induces a distributive law $\lambda: TD \Rightarrow DT$, namely as composite:*

$$
T(X^A \times 2) \xrightarrow{\langle T(\pi_1), T(\pi_2) \rangle} T(X^A) \times T(2) \xrightarrow{\mathsf{st} \times \beta} T(X)^A \times 2
$$

where $\mathsf{st}: T(X^A) \to T(X)^A$ is the so-called strength map $\mathsf{st}(u)(a) = T(\lambda f \in X^A. f(a))(u)$.

The Eilenberg-Moore algebra forming the final λ-bialgebra with the final coalgebra $\mathcal{L}(A) \xrightarrow{\cong} D\mathcal{L}(A)$ like in Lemma 3 is obtained pointwise as:

$$
T(\mathcal{L}(A)) = T(2^{A^\star}) \xrightarrow{\mathsf{st}} T(2)^{A^\star} \xrightarrow{\beta^{A^\star}} 2^{A^\star} = \mathcal{L}(A). \qquad \square
$$

3.2.2 Languages and non-deterministic automata

A non-deterministic automaton, with alphabet A, is a coalgebra of the form $\langle \delta, \varepsilon \rangle \colon X \to \mathcal{P}(X)^A \times 2$. The transition function δ now maps a state x and an input a to a set $\delta(x)(a) \subseteq X$ of successor states.

As observed in [5], there is a distributive law $\mathcal{P}D \Rightarrow D\mathcal{P}$, where $D = (-)^A \times K_2$ as defined in Subsection 3.2.1. It is an instance of Theorem 10, because the set $2 = \{0, 1\} = \mathcal{P}(1)$ carries a (free) $\mathcal{P}$-monad structure, which is of course given by union $\bigvee$ wrt. the standard order $0 \leq 1$. The resulting distributive law, say $\lambda^{\mathcal{P}}$, is given explicitly by:

$$\mathcal{P}(X^A \times 2) \xrightarrow{\quad \lambda_X^{\mathcal{P}} \quad} \mathcal{P}(X)^A \times 2$$

$$U \longmapsto \langle \lambda a \in A. \{f(a) \mid \exists b. (f, b) \in U\},\ \exists f. (f, 1) \in U \rangle$$

It is not hard to see that the (final) $\lambda^{\mathcal{P}}$-bialgebra induced as in Lemma 3 (and given in Theorem 10) involves the union operation $\bigcup \colon \mathcal{P}(\mathcal{L}(A)) \to \mathcal{L}(A)$ in:

$$
\begin{array}{ccc}
D\mathcal{P}\mathcal{L}(A) & \dashrightarrow^{D(\bigcup)} & D\mathcal{L}(A) = \mathcal{L}(A)^A \times 2 \\
\lambda_{\mathcal{L}(A)}^{\mathcal{P}} \uparrow & & \\
\mathcal{P}D\mathcal{L}(A) & & \cong \big\downarrow \langle \delta, \varepsilon \rangle \\
\mathcal{P}(\langle \delta, \varepsilon \rangle) \uparrow \cong & & \\
\mathcal{P}\mathcal{L}(A) & \dashrightarrow_{\bigcup} & \mathcal{L}(A)
\end{array}
$$

with $\langle \delta_\cup, \varepsilon_\cup \rangle$ on the left.

In fact, this says that the union $\bigcup$ of languages can be defined by coinduction via the D-coalgebra $\langle \delta_\cup, \varepsilon_\cup \rangle$ given by:

$$\varepsilon_\cup(U) = (\langle \rangle \in \bigcup U) \quad \text{and} \quad \delta_\cup(U)(a) = \{L_a \mid L \in U\}.$$

One of the nice observations in [5], see its Corollary 4.4.6, is that the languages associated with a non-deterministic automaton can be defined by $\lambda^{\mathcal{P}}$-coinduction, *i.e.* as solution of a $\lambda^{\mathcal{P}}$-equation, namely of the automaton $X \to D\mathcal{P}(X) = \mathcal{P}(X)^A \times 2$ itself, like in:

$$
\begin{array}{ccc}
\mathcal{P}(X)^A \times 2 = D\mathcal{P}(X) & \dashrightarrow & D\mathcal{P}\mathcal{L}(A) \\
\uparrow & & \big\downarrow D(\bigcup) \\
& & D\mathcal{L}(A) = \mathcal{L}(A)^A \times 2 \\
& & \uparrow \cong \\
X & \dashrightarrow & \mathcal{L}(A)
\end{array}
$$

For instance, the non-deterministic automaton

with state 2 terminal
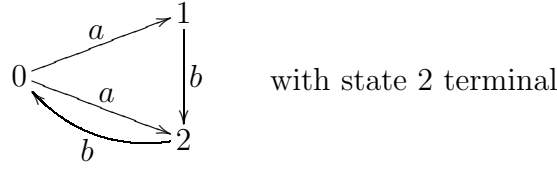
gives rise to the differential equations

$$\frac{\partial L_0}{\partial a} = L_1 + L_2 \quad \frac{\partial L_0}{\partial b} = 0 \quad \frac{\partial L_1}{\partial a} = 0 \quad \frac{\partial L_1}{\partial b} = L_2 \quad \frac{\partial L_2}{\partial a} = 0 \quad \frac{\partial L_2}{\partial b} = L_0$$

$$\langle\rangle \notin L_0 \qquad\qquad\qquad \langle\rangle \notin L_1 \qquad\qquad\qquad \langle\rangle \in L_2$$

What is important is that the expressions on the right-hand-side may now involve a + operation for union. The solution, obtained by $\lambda^{\mathcal{P}}$-coinduction as a function $L\colon 3 \to \mathcal{L}(A)$ can be described explicitly as $L_0 = (a+ab)(b(a+ab))^\star$, $L_1 = b(b(a+ab))^\star$ and $L_2 = (b(a+ab))^\star$.

3.2.3 Languages and language automata

Our next step is to use a new kind of automata, namely of the form $\langle\delta, \varepsilon\rangle\colon X \to \mathcal{L}(X)^A{\times}2$. We call them "language automata" because of the occurrence of the language monad $\mathcal{L}$. Such automata may involve non-deterministic transitions $x \xrightarrow{a} \langle x_1, \ldots, x_n\rangle$ to multiple states, for instance in some decomposed form.

Again by Theorem 10 there is a distributive law $\lambda^{\mathcal{L}}\colon \mathcal{L}D \Rightarrow D\mathcal{L}$. This time we need an algebra $\mathcal{L}(2) \to 2$. It is again obtained by freeness, using that $\mathcal{L}(0) = \mathcal{P}(0^\star) = \mathcal{P}(1) = 2$. The resulting multiplication map $\mu\colon \mathcal{L}(2) \to 2$ is given by $\mu(V) = 1$ iff $\langle 1, \ldots, 1\rangle \in V$ for some sequence $\langle 1, \ldots, 1\rangle$ of 1's only. Concretely, the resulting distributive law $\lambda^{\mathcal{L}}_X\colon \mathcal{P}((X^A \times 2)^\star) \to \mathcal{P}(X^\star)^A \times 2$ is:

$$\lambda^{\mathcal{L}}_X(V)$$
$$= \langle \lambda a \in A. \{\langle f_1(a), \ldots, f_n(a)\rangle \mid \exists b_1, \ldots, b_n \in 2. \langle (f_1, b_1), \ldots, (f_n, b_n)\rangle \in V\},$$
$$\exists f_1, \ldots, f_n \in X^A. (f_1, 1), \ldots, (f_n, 1) \in V \rangle$$

It is not hard to see that the map of monads $\sigma = \mathcal{P}(\eta^\star)\colon \mathcal{P} \Rightarrow \mathcal{L}$—see Proposition 8—commutes with the distributive laws $\lambda^{\mathcal{P}}$ and $\lambda^{\mathcal{L}}$, in the sense that the following diagram commutes.

12

Like before we get a final $\lambda^{\mathcal{L}}$-bialgebra, with algebra structure $\bigsqcup\colon \mathcal{L}^2(A) \to \mathcal{L}(A)$ determined in:

$$
\begin{array}{ccc}
D\mathcal{L}^2(A) & \dashrightarrow\!\!\!\!\!\!\!\overset{D(\bigsqcup)}{}\!\!\!\!\!\!\!\dashrightarrow & D\mathcal{L}(A) = \mathcal{L}(A)^A \times 2 \\
\lambda^{\mathcal{L}}_{\mathcal{L}(A)}\uparrow & & \uparrow \\
\mathcal{L}D\mathcal{L}(A) & \cong \downarrow\langle\delta,\varepsilon\rangle & \\
\mathcal{L}(\langle\delta,\varepsilon\rangle)\uparrow\cong & & \\
\mathcal{L}^2(A) & \dashrightarrow\!\!\!\!\!\!\underset{\bigsqcup}{}\!\!\!\!\!\!\dashrightarrow & \mathcal{L}(A)
\end{array}
$$

$\langle\delta_{\bigsqcup},\varepsilon_{\bigsqcup}\rangle$

This means that $\bigsqcup$ is given on a set $V \in \mathcal{L}^2(A)$ of sequences of languages by:

$$\langle\rangle \in \bigsqcup V \iff \varepsilon_{\bigsqcup}(V) = 1 \iff \exists\langle L_1,\ldots,L_n\rangle \in V.\,\forall i.\,\langle\rangle \in L_i$$

and for $a \in A$,

$$(\bigsqcup V)_a = \bigsqcup\{\langle (L_1)_a,\ldots,(L_n)_a\rangle \mid \langle L_1,\ldots,L_n\rangle \in V\}.$$

This second equation implies that $\bigsqcup$ is not the multiplication $\mu\colon \mathcal{L}^2(A) \to \mathcal{L}(A)$ of the monad $\mathcal{L}$. Following the formula in Theorem 10 we can describe it explicitly as union of intersections:

$$\bigsqcup V = \bigcup\{L_1 \cap \cdots \cap L_n \mid \langle L_1,\ldots,L_n\rangle \in V\}.$$

These language automata $X \to \mathcal{L}(X)^A \times 2$ resemble alternating automata [21]. It is at this stage not clear how useful the additional expressive power is for solving more expressive differential equations (with $\lambda^{\mathcal{L}}$-coinduction).

4 Free monads and their distributive laws

In this section we consider an endofunctor $F\colon \mathbb{C} \to \mathbb{C}$ with two canonical associated monads F^* and F^∞, together with distributive laws λ^* and λ^∞ over F. Propositions 11 and 12 contain standard results about F^* which are not used directly, but provide the setting for similar (new) results about F^∞. The latter form the basis for our main result in Section 6, namely the link between two forms of equation solving.

4.1 The free monad on a functor

Let $F\colon \mathbb{C} \to \mathbb{C}$ be an arbitrary endofunctor on a category $\mathbb{C}$ with (binary) coproducts $+$. The only assumption we make at this stage is that for each

object $X \in \mathbb{C}$ the functor $X + F(-): \mathbb{C} \to \mathbb{C}$ has an initial algebra. We shall use the following notation. The carrier of this initial algebra will be written as $F^*(X)$ with structure map given as:

$$X + F(F^*(X)) \xrightarrow[\cong]{\alpha_X} F^*(X)$$

Further, we shall write

$$\eta_X = \alpha_X \circ \kappa_1 \qquad \tau_X = \alpha_X \circ \kappa_2,$$

so that $\alpha_X = [\eta_X, \tau_X]$.

The mapping $X \mapsto F^*(X)$ is functorial: for $f: X \to Y$ we get:

$$
\begin{array}{ccc}
X + F(F^*(X)) & \xrightarrow{\mathrm{id} + F(F^*(f))} & X + F(F^*(Y)) \\
{\scriptstyle\alpha_X}\downarrow{\scriptstyle\cong} & & \downarrow{\scriptstyle[\eta_Y \circ f, \tau_Y]} \\
F^*(X) & \xdashrightarrow{F^*(f)} & F^*(Y)
\end{array}
$$

This means that

$$F^*(f) \circ \eta_X = \eta_Y \circ f \qquad F^*(f) \circ \tau_X = \tau_Y \circ F(F^*(f)),$$

i.e. that $\eta: \mathrm{id} \Rightarrow F^*$ and $\tau: FF^* \Rightarrow F^*$ are natural transformations.

Next we establish that F^* is a monad. The multiplication μ is obtained in:

$$
\begin{array}{ccc}
F^*(X) + F(F^*(F^*(X))) & \xrightarrow{\mathrm{id} + F(\mu_X)} & F^*(X) + F(F^*(X)) \\
{\scriptstyle\alpha_{F^*(X)}}\downarrow{\scriptstyle\cong} & & \downarrow{\scriptstyle[\mathrm{id}, \tau_X]} \\
F^*(F^*(X)) & \xdashrightarrow{\mu_X} & F^*(X)
\end{array}
$$

This yields one of the monad equations, namely $\mu_X \circ \eta_{F^*(X)} = \mathrm{id}$. The related equation $\mu_X \circ F^*(\eta_X) = \mathrm{id}$ follows from uniqueness of algebra maps $\alpha_X \to \alpha_X$:

$$
\begin{aligned}
\mu_X \circ F^*(\eta_X) \circ \alpha_X &= \mu_X \circ [\eta_{F^*(X)} \circ \eta_X, \tau_{F^*(X)}] \circ (\mathrm{id} + F(F^*(\eta_X))) \\
&= [\eta_X, \tau_X \circ F(\mu_X)] \circ (\mathrm{id} + F(F^*(\eta_X))) \\
&= \alpha_X \circ (\mathrm{id} + F(\mu_X \circ F^*(\eta_X))).
\end{aligned}
$$

Similarly, the other requirements making F^* a monad are obtained.

The following standard result sums up the situation.

Proposition 11 *Let* $F: \mathbb{C} \to \mathbb{C}$ *with induced monad* (F^*, η, μ) *be as described above.*

14

(1) *The mapping $X \mapsto [F(F^*(X)) \xrightarrow{\tau_X} F^*(X)]$ forms a left adjoint to the forgetful functor $U: \mathbf{Alg}(F) \to \mathbb{C}$. The monad induced by this adjunction is (F^*, η, μ).*

(2) *The mapping $\sigma_X = \tau_X \circ F(\eta_X): F(X) \to F^*(X)$ yields a natural transformation $F \Rightarrow F^*$ that makes F^* the free monad on F.* $\square$

The next observation shows that the monad F^* of (finite) F-terms fits with the behaviour of F. It follows from a general observation (made for instance in [5]) that distributive laws $F^*G \Rightarrow GF^*$ correspond to ordinary natural transformations $FG \Rightarrow GF^*$. Hence by taking $G = F$ and unit $FF \Rightarrow FF^*$ one gets $F^*F \Rightarrow FF^*$. But here we shall present the construction explicitly.

Proposition 12 *Let $F: \mathbb{C} \to \mathbb{C}$ have free monad F^*. Then there is a distributive law $\lambda^*: F^*F \Rightarrow FF^*$.*

Proof. We define $\lambda_X^*: F^*(FX) \to F(F^*X)$ as follows.

$$F^*(FX) \xrightarrow[\cong]{\alpha_{FX}^{-1}} FX + F(F^*(FX)) \xrightarrow{[F(\eta_X), F(\mu_X \circ F^*(\sigma_X))]} F(F^*X)$$

where $\sigma_X = \tau_X \circ F(\eta_X): F(X) \to F^*(X)$ as introduced in Proposition 11 (2).

Example 13 *Let $Z = \mathbb{R}^{\mathbb{N}}$ be the set of streams of real numbers. It is of course the final coalgebra of the functor $F = \mathbb{R} \times (-)$, via the head and tail operations $\langle hd, tl \rangle: Z \xrightarrow{\cong} \mathbb{R} \times Z$. It is shown in [23] that on such streams one can coinductively define binary operators $\oplus$ for sum and $\otimes$ for shuffle product satisfying the recursive equations:*

$$x \oplus y = (hd(x) + hd(y)) \cdot (tl(x) \oplus tl(y))$$
$$x \otimes y = (hd(x) \times hd(y)) \cdot ((tl(x) \otimes y) \oplus (x \otimes tl(y))),$$

where $\cdot$ is prefix.

It is easy to see that one defines $\oplus$ by ordinary coinduction, in:

$$
\begin{array}{ccc}
\mathbb{R} \times (Z \times Z) & \xrightarrow{id \times \oplus} & \mathbb{R} \times Z \\
{\scriptstyle c_\oplus} \uparrow & & \cong \downarrow {\scriptstyle \langle hd, tl \rangle} \\
Z \times Z & \xrightarrow{\quad \oplus \quad} & Z
\end{array}
$$

where the coalgebra $c_\oplus$ is defined by:

$$c_\oplus(x, y) = \langle hd(x) + hd(y), \langle tl(x), tl(y) \rangle \rangle.$$

Once we have $\oplus: Z \times Z \to Z$ we show how to obtain $x \otimes y$ as a solution of a λ-equation. We start from the signature functor $\Sigma(X) = X \times X$. There

is an obvious natural transformation $\Sigma F \Rightarrow F\Sigma^*$ given by $(\langle r, x\rangle, \langle s, y\rangle) \longmapsto \langle r+s, (x, y)\rangle$. By [5, Lemma 3.4.24] it lifts to a distributive law $\lambda\colon \Sigma^*F \Rightarrow F\Sigma^*$ involving the associated free monad Σ^*. The algebra $\oplus\colon \Sigma(Z) \to Z$ yields an Eilenberg-Moore algebra $[\![-]\!]\colon \Sigma^*(Z) \to Z$, which is by the same result of [5] a λ-bialgebra. Now we obtain $\otimes$ as solution in:

$$
\begin{array}{ccc}
\mathbb{R} \times \Sigma^*(Z \times Z) & \xdashrightarrow{\ id \times \Sigma^*(\otimes)\ } & \mathbb{R} \times \Sigma^*(Z) \\
{\scriptstyle d_\otimes}\big\uparrow & & \big\downarrow{\scriptstyle id \times [\![-]\!]} \\
& & \mathbb{R} \times Z \\
& & \cong\big\uparrow{\scriptstyle \langle hd, tl\rangle} \\
Z \times Z & \xdashrightarrow{\ \otimes\ } & Z
\end{array}
$$

in which the λ-equation $d_\otimes$ is defined by:

$$d_\otimes(x, y) = \langle hd(x) \times hd(y),\ (tl(x), y) \underline{\oplus} (x, tl(y))\rangle,$$

where $\underline{\oplus}$ is a symbol for sum in the language of terms on pairs from $Z \times Z$. Here we exploit the expressive power of the λ-approach, because we can now write terms as second component.

Clearly, the above diagram says:

$$hd(x \otimes y) = hd(x) \times hd(y).$$

And also, as required:

$$
\begin{aligned}
tl(x \otimes y) &= ([\![-]\!] \circ \Sigma^*(\otimes) \circ \pi_2 \circ d_\otimes)(x, y) \\
&= ([\![-]\!] \circ \Sigma^*(\otimes))(\,(tl(x), y) \underline{\oplus} (x, tl(y))\,) \\
&= [\![\,(tl(x) \otimes y) \underline{\oplus} (x \otimes tl(y))\,]\!] \\
&= (tl(x) \otimes y) \oplus (x \otimes tl(y)).
\end{aligned}
$$

4.2 The free iterative monad on a functor

Let, like in the previous section, $F\colon \mathbb{C} \to \mathbb{C}$ be an arbitrary endofunctor on a category $\mathbb{C}$ with (binary) coproducts $+$. The assumption we now make is that for each object $X \in \mathbb{C}$ the functor $X + F(-)\colon \mathbb{C} \to \mathbb{C}$ has a final coalgebra—instead of an initial algebra. We shall use the following notation. The carrier of this final calgebra will be written as $F^\infty(X)$ with structure map given as:

$$F^\infty(X) \xrightarrow[\cong]{\ \zeta_X\ } X + F(F^\infty(X))$$

The sets $F^*(X)$ in the previous section are understood as the set of finite terms of type F with free variables from X. Here we understand $F^\infty(X)$ as the set of both finite and infinite terms (or trees) with free variables in X.

Like before, we shall write:

$$\eta_X = \zeta_X^{-1} \circ \kappa_1 \qquad \tau_X = \zeta_X^{-1} \circ \kappa_2.$$

Functoriality of F^∞ is obtained as follows. For $f\colon X \to Y$ in $\mathbb{C}$ we get:

$$
\begin{array}{ccc}
Y + F(F^\infty(X)) & \xrightarrow{\mathrm{id} + F(F^\infty(f))} & Y + F(F^\infty(Y)) \\
(f + \mathrm{id}) \circ \zeta_X \uparrow & & \cong \downarrow \zeta_Y \\
F^\infty(X) & \xrightarrow{\quad F^\infty(f) \quad} & F^\infty(Y)
\end{array}
$$

This means that

$$F^\infty(f) \circ \eta_X = \eta_Y \circ f \qquad F^\infty(f) \circ \tau_X = \tau_Y \circ F(F^\infty(f)),$$

i.e. that $\eta\colon \mathrm{id} \Rightarrow F^\infty$ and $\tau\colon FF^\infty \Rightarrow F^\infty$ are natural transformations.

It is shown in [3,19] that F^∞ is a monad[1]. The multiplication operation μ is rather complicated, and can best be introduced via substitution $t[s/x]$. What we mean is replacing all occurrences (if any) of the variable x in the term t by the term s, but now for possibly infinite terms. In most general form, this substitution $t[\vec{s}/\vec{x}]$ replaces all occurrences of all variables $x \in X$ simultaneously. In this way, substitution may be described as an operation which tells how an X-indexed collection $(s_x)_{x \in X}$ of terms $s_x \in F^\infty(Y)$ acts on a term $t \in F^\infty(X)$. More precisely, substitution becomes an operation $\mathsf{subst}(s)\colon F^\infty(X) \to F^\infty(Y)$, for a function $s\colon X \to F^\infty(Y)$. As usual, such a substitution operation should respect the term structure—*i.e.* be a homomorphism—and be trivial on variables. Standardly, substitution is defined by induction on the structure of (finite) terms. But since we are dealing here with possibly infinite terms, we have to use coinduction. This makes the substitution more challenging. In general, it is done as follows.

Lemma 14 *Let X, Y be arbitrary sets. Each function $s\colon X \to F^\infty(Y)$ gives rise to a **coalgebraic substitution** operator $\mathsf{subst}(s)\colon F^\infty(X) \to F^\infty(Y)$,*

---

[1] Similar results appeared earlier in [20], but for the functor $Y \mapsto F(X + Y)$.

namely the unique homomorphism of F-algebras:

$$F(F^\infty(X)) \xrightarrow{F(\mathsf{subst}(s))} F(F^\infty(Y))$$

with the diagrams:

$$
\begin{array}{ccc}
F(F^\infty(X)) & \xrightarrow{F(\mathsf{subst}(s))} & F(F^\infty(Y)) \\
\tau_X \downarrow & & \downarrow \tau_Y \\
F^\infty(X) & \xrightarrow{\mathsf{subst}(s)} & F^\infty(Y)
\end{array}
\qquad \text{with} \qquad
\begin{array}{c}
X \\
\eta_X \downarrow \quad \searrow^{s} \\
F^\infty(X) \xrightarrow{\mathsf{subst}(s)} F^\infty(Y)
\end{array}
$$

Proof. We begin by defining a coalgebra structure on the coproduct $F^\infty(Y) + F^\infty(X)$ of terms, namely as the vertical composite on the left below. This coalgebra on $F^\infty(Y) + F^\infty(X)$ simply unravels on $F^\infty(Y)$ on the left component of $+$, and it applies s to the variables in the right component.

$$
\begin{array}{ccc}
Y + F(F^\infty(Y) + F^\infty(X)) & \xrightarrow{\quad \mathrm{id}_Y + F(f) \quad} & Y + F(F^\infty(Y)) \\
\big\uparrow {\scriptstyle [(\mathrm{id}_Y + F(\kappa_1))\, \circ\, \zeta_Y,\, \kappa_2\, \circ\, F(\kappa_2)]} & & \big\uparrow {\scriptstyle \zeta_Y} \\
F^\infty(Y) + F(F^\infty(X)) & & \\
\big\uparrow {\scriptstyle [\kappa_1,\, s + \mathrm{id}]} & & \cong \\
F^\infty(Y) + (X + F(F^\infty(X))) & & \\
\big\uparrow {\scriptstyle \mathrm{id}_Y + \zeta_X} & & \\
F^\infty(Y) + F^\infty(X) & \xrightarrow{\qquad f \qquad} & F^\infty(Y)
\end{array}
$$

One first proves that $f \circ \kappa_1$ is the identity, using uniqueness of coalgebra maps $\zeta_Y \to \zeta_Y$. Then, $f \circ \kappa_2$ is the required map $\mathsf{subst}(s)$. $\qquad\square$

In the remainder of this paper we shall make frequent use of this substitution operator $\mathsf{subst}(-)$. Computations with substitution are made much easier with the following elementary results. Proofs are obtained via the uniqueness property of substitution.

Lemma 15 *For $s\colon X \to F^\infty(Y)$ we have:*

(1) $\mathsf{subst}(\eta_X) = id_{F(X)}$.
(2) $\mathsf{subst}(s) \circ F^\infty(f) = \mathsf{subst}(s \circ f)$, *for $f\colon Z \to X$.*
(3) $\mathsf{subst}(r) \circ \mathsf{subst}(s) = \mathsf{subst}(\mathsf{subst}(r) \circ s)$, *for $r\colon Y \to F^\infty(Z)$.*
(4) $F^\infty(f) = \mathsf{subst}(\eta_Z \circ f)$, *for $f\colon Y \to Z$, and hence $\mathsf{subst}(F^\infty(f) \circ s) = F^\infty(f) \circ \mathsf{subst}(s)$.*
(5) $\mathsf{subst}(s) = [s, \tau_Y \circ F(\mathsf{subst}(s))] \circ \zeta_X$. $\qquad\square$

Proposition 16 *The map $\mu_X = \mathsf{subst}(id_{F^\infty(X)})\colon F^\infty(F^\infty(X)) \to F^\infty(X)$ makes the triple (F^∞, η, μ) a monad.*

This monad F^∞ is called the **iterative** monad on F, via the natural transformation $\sigma = \tau \circ F\eta\colon F \Rightarrow F^\infty$.

In [2] it is shown that F^∞ is in fact a *free* iterative monad, in a suitable sense. This freeness is not relevant here.

Proof. We check the monad equations, using Lemma 15.

$$
\begin{aligned}
\mu_X \circ \eta_{F^\infty X} &= \mathsf{subst}(\mathrm{id}_{F^\infty(X)}) \circ \eta_{F^\infty X} \\
&= \mathrm{id}_{F^\infty(X)}. \\
\mu_X \circ F^\infty(\eta_X) &= \mathsf{subst}(\mathrm{id}_{F^\infty(X)}) \circ F^\infty(\eta_X) \\
&= \mathsf{subst}(\mathrm{id}_{F^\infty(X)} \circ \eta_X) \\
&= \mathrm{id}_{F^\infty(X)}. \\
\mu_X \circ F^\infty(\mu_X) &= \mathsf{subst}(\mathrm{id}_{F^\infty(X)}) \circ F^\infty(\mu_X) \\
&= \mathsf{subst}(\mu_X) \\
&= \mathsf{subst}(\mathsf{subst}(\mathrm{id}_{F^\infty(X)}) \circ \mathrm{id}_{F^\infty(F^\infty(X))}) \\
&= \mathsf{subst}(\mathrm{id}_{F^\infty(X)}) \circ \mathsf{subst}(\mathrm{id}_{F^\infty(F^\infty(X))}) \\
&= \mu_X \circ \mu_{F^\infty(X)}. \qquad \square
\end{aligned}
$$

The following is less standard.

Proposition 17 *Consider $F\colon \mathbb{C} \to \mathbb{C}$ with its iterative monad F^∞.*

(1) There is a distributive law $\lambda^\infty\colon F^\infty F \Rightarrow F F^\infty$.
(2) The induced mediating map of monads $F^* \Rightarrow F^\infty$ commutes with the distributive laws, in the sense that the following diagram commutes.

$$
\begin{array}{ccc}
F^* F & \longrightarrow & F^\infty F \\
\lambda^* \downarrow & & \downarrow \lambda^\infty \\
F F^* & \longrightarrow & F F^\infty
\end{array}
$$

Proof. Like for λ^* we define $\lambda_X^\infty\colon F^\infty(FX) \to F(F^\infty X)$ as follows:

$$
F^\infty(FX) \xrightarrow[\cong]{\zeta_{FX}} FX + F(F^\infty(FX)) \xrightarrow{[F(\eta_X),\, F(\mu_X \circ F^\infty(\sigma_X))]} F(F^\infty X)
$$

where $\sigma_X = \tau_X \circ F(\eta_X)\colon F(X) \to F^\infty(X)$ as introduced in Proposition 16. It satisfies, like in the proof of Proposition 12,

$$
\begin{aligned}
\mu_X \circ \sigma_{F^\infty X} &= \mathsf{subst}(\mathrm{id}_{F^\infty X}) \circ \tau_{F^\infty X} \circ F(\eta_{F^\infty X}) \\
&= \tau_X \circ F(\mathsf{subst}(\mathrm{id}_{F^\infty X})) \circ F(\eta_{F^\infty X}) \\
&= \tau_X \circ F(\mathrm{id}_{F^\infty X}) \\
&= \tau_X.
\end{aligned}
\tag{3}
$$

Then:

$$\lambda_X^\infty \circ \eta_{FX} = [F(\eta_X), F(\mu_X \circ F^\infty(\sigma_X))] \circ \zeta_{FX} \circ \eta_{FX}$$
$$= [F(\eta_X), F(\mu_X \circ F^\infty(\sigma_X))] \circ \kappa_1$$
$$= F(\eta_X).$$

We shall use the following two auxiliary results:

$$\mu_X \circ \sigma_{F^\infty X} \circ \lambda_X^\infty = \mu_X \circ F^\infty(\sigma_X)$$
$$F(\tau_X) \circ F(\lambda_X^\infty) = \lambda_X^\infty \circ \tau_{FX}. \tag{4}$$

We first prove the first equation, and use it immediately to prove the second one.

$\mu_X \circ \sigma_{F^\infty X} \circ \lambda_X^\infty$

$= [\mu_X \circ \sigma_{F^\infty X} \circ F(\eta_X), \mu_X \circ \sigma_{F^\infty X} \circ F(\mu_X \circ F^\infty(\sigma_X))] \circ \zeta_{FX}$

   by definition of λ

$= [\mu_X \circ F^\infty(\eta_X) \circ \sigma_X, \mu_X \circ F^\infty(\mu_X \circ F^\infty(\sigma_X)) \circ \sigma_{F^\infty FX}] \circ \zeta_{FX}$

   by naturality

$= [\mu_X \circ \eta_{F^\infty X} \circ \sigma_X, \mu_X \circ \mu_{F^\infty X} \circ F^\infty F^\infty(\sigma_X) \circ \sigma_{F^\infty FX}] \circ \zeta_{FX}$

   by the monad laws

$= [\mu_X \circ F^\infty(\sigma_X) \circ \eta_{FX}, \mu_X \circ F^\infty(\sigma_X) \circ \mu_{FX} \circ \sigma_{F^\infty FX}] \circ \zeta_{FX}$

   by naturality

$= \mu_X \circ F^\infty(\sigma_X) \circ [\eta_{FX}, \tau_{FX}] \circ \zeta_{FX}$

   by (3)

$= \mu_X \circ F^\infty(\sigma_X)$

   by definition of η, τ.

$F(\tau_X) \circ F(\lambda_X^\infty)$

$= F(\mu_X \circ \sigma_{F^\infty X} \circ \lambda_X^\infty)$

   by (3)

$= F(\mu_X \circ F^\infty(\sigma_X))$

   as we have just shown

$= [F(\eta_X), F(\mu_X \circ F^\infty(\sigma_X))] \circ \kappa_2$

   obviously

$= \lambda_X^\infty \circ \tau_{FX}$

   by definition of τ.

20

Now we are ready to prove that λ^∞ commutes with multiplications.

$$\lambda_X^\infty \circ \mu_{FX}$$
$$= \lambda_X^\infty \circ [\mathrm{id}, \tau_{FX} \circ F(\mu_{FX})] \circ \zeta_{F\infty FX} \qquad \text{by Lemma 15 (5)}$$
$$= [\lambda_X^\infty, \lambda_X^\infty \circ \tau_{FX} \circ F(\mu_{FX})] \circ \zeta_{F\infty FX}$$
$$\overset{(4)}{=} [\lambda_X^\infty, F(\tau_X \circ \lambda_X^\infty \circ \mu_{FX})] \circ \zeta_{F\infty FX}$$
$$\overset{(3)}{=} [\lambda_X^\infty, F(\mu_X \circ \sigma_{F\infty X} \circ \lambda_X^\infty \circ \mu_{FX})] \circ \zeta_{F\infty FX}$$
$$\overset{(4)}{=} [\lambda_X^\infty, F(\mu_X \circ F^\infty(\sigma_X) \circ \mu_{FX})] \circ \zeta_{F\infty FX}$$
$$= [\lambda_X^\infty, F(\mu_X \circ \mu_{F\infty X} \circ F^\infty F^\infty(\sigma_X))] \circ \zeta_{F\infty FX}$$
$$= [\lambda_X^\infty, F(\mu_X \circ F^\infty(\mu_X \circ F^\infty(\sigma_X)))] \circ \zeta_{F\infty FX}$$
$$\overset{(4)}{=} [\lambda_X^\infty, F(\mu_X \circ F^\infty(\mu_X \circ \sigma_{F\infty X} \circ \lambda_X^\infty))] \circ \zeta_{F\infty FX}$$
$$= [\mathrm{id}, F(\mu_X \circ \mu_{F\infty X} \circ F^\infty(\sigma_{F\infty X}))] \circ (\lambda_X^\infty + F(F^\infty \lambda_X^\infty)) \circ \zeta_{F\infty FX}$$
$$= F(\mu_X) \circ [F(\eta_{F\infty X}), F(\mu_{F\infty X} \circ F^\infty(\sigma_{F\infty X}))] \circ \zeta_{FF\infty X} \circ F^\infty(\lambda_X^\infty)$$
$$\qquad \text{by definition of } F^\infty \text{ on morphisms}$$
$$= F(\mu_X) \circ \lambda_{F\infty X}^\infty \circ F^\infty(\lambda_X^\infty).$$

In order to prove the second point of the proposition we have to disambiguate the notation. Let's write the monad F^* as (F^*, η^*, μ^*) with associated τ^* and σ^*, and F^∞ as $(F^\infty, \eta^\infty, \mu^\infty)$ with τ^∞ and σ^∞. The induced mediating map $\overline{\sigma^\infty} : F^* \Rightarrow F^\infty$ is then given by:

$$
\begin{array}{ccc}
X + F(F^*X) & \xrightarrow{\ \mathrm{id} + F(\overline{\sigma^\infty}_X)\ } & X + F(F^\infty X) \\
\alpha_X \downarrow \cong & & \cong \uparrow \zeta_X \\
F^*X & \xdashrightarrow{\ \overline{\sigma^\infty}_X\ } & F^\infty X
\end{array}
$$

We already know (from Proposition 11) that $\overline{\sigma^\infty}$ is a homomorphism of monads satisfying $\overline{\sigma^\infty} \circ \sigma^* = \sigma^\infty$. Hence $\overline{\sigma^\infty}$ commutes with the distributive laws:

$$\lambda_X^\infty \circ \overline{\sigma^\infty}_{FX} = [F(\eta_X^\infty), F(\mu_X^\infty \circ F^\infty(\sigma_X^\infty))] \circ \zeta_{FX} \circ \overline{\sigma^\infty}_{FX}$$
$$= [F(\eta_X^\infty), F(\mu_X^\infty \circ F^\infty(\sigma_X^\infty))] \circ (\mathrm{id} + F(\overline{\sigma^\infty}_{FX})) \circ \alpha_{FX}^{-1}$$
$$= [F(\eta_X^\infty), F(\mu_X^\infty \circ F^\infty(\sigma_X^\infty) \circ \overline{\sigma^\infty}_{FX})] \circ \alpha_{FX}^{-1}$$
$$= [F(\eta_X^\infty), F(\mu_X^\infty \circ \overline{\sigma^\infty}_{F\infty X} \circ F^*(\sigma_X^\infty))] \circ \alpha_{FX}^{-1}$$
$$= [F(\eta_X^\infty), F(\mu_X^\infty \circ \overline{\sigma^\infty}_{F\infty X} \circ F^*(\overline{\sigma^\infty}_X \circ \sigma_X^*))] \circ \alpha_{FX}^{-1}$$
$$= [F(\overline{\sigma^\infty}_X \circ \eta_X^*), F(\overline{\sigma^\infty}_X \circ \mu_X^* \circ F^*(\sigma_X^*))] \circ \alpha_{FX}^{-1}$$
$$= F(\overline{\sigma^\infty}_X) \circ [F(\eta_X^*), F(\mu_X^* \circ F^*(\sigma_X^*))] \circ \alpha_{FX}^{-1}$$
$$= F(\overline{\sigma^\infty}_X) \circ \lambda_X^*. \qquad \square$$

5 Iteration and solutions of equations

The material in this section comes (again) from [2]. In Definition 5 we have seen an abstract notion of λ-equation and solution. A bit more concretely, for a functor F, a set of recursive equations—often simply called a recursive equation—consists first of all of a set X of recursive variables. For each variable $x \in X$ we have a corresponding term t in an equation $x = t$. We shall allow this term to be infinite. The term t may involve both variables from an already given set Y, and from our new set of recursive variables X. Hence $t \in F^{\infty}(Y + X)$. Summarising, a **recursive equation** is a map $e: X \to F^{\infty}(Y + X)$. We shall often call such an e a ∞-**equation**, in contrast to a λ-equation $X \to FTX$—as in Definition 5.

Definition 18 *Let $F: \mathbb{C} \to \mathbb{C}$ be a functor, with for $X \in \mathbb{C}$ a final coalgebra $F^{\infty}(X) \xrightarrow{\cong} X + F(F^{\infty}(X))$.*

*A **solution** for an ∞-equation $e: X \to F^{\infty}(Y + X)$ is a map $\mathsf{sol}(e): X \to F^{\infty}(Y)$ that produces an appropriate term $\mathsf{sol}(e)(x)$ for each recursive variable $x \in X$. This means that substituting the cotuple $[\eta_Y, \mathsf{sol}(e)]: Y + X \to F^{\infty}(Y)$ in e yields the solution $\mathsf{sol}(e)$, i.e.*

$$
\begin{array}{ccc}
\mathsf{sol}(e) & & X \xrightarrow{\quad e \quad} F^{\infty}(Y + X) \\
& in & \quad\quad \searrow\!\!\scriptstyle{\mathsf{sol}(e)} \quad \Big\downarrow \scriptstyle{\mathsf{subst}([\eta_Y, \mathsf{sol}(e)])} \\
= \mathsf{subst}([\eta_Y, \mathsf{sol}(e)]) \circ e & & F^{\infty}(Y)
\end{array}
$$

This shows that the solution is a fixed point of $\mathsf{subst}([\eta_Y, -]) \circ e$.

Like for λ-equations, we are interested in unique solutions for ∞-equations. Do they always exist? Not in trivial equations, like $x = x$, where any term is a solution. Such equations are standardly excluded by requiring that the terms of the recursive equation are 'guarded', *i.e.* that its terms are not variables from X. This notion can also be formulated in a general categorical setting: an ∞-equation $e: X \to F^{\infty}(Y + X)$ is called **guarded** if it factors (in a necessarily unique way, assuming that coprojections κ_i are monos) as:

$$
\begin{array}{ccc}
 & Y + F(F^{\infty}(Y + X)) & \\
\!\!\!{}^{g}\!\nearrow\!\!\raise2pt{\text{-}} & \Big\downarrow \scriptstyle{\kappa_1 + \mathrm{id}} & \\
\raise2pt{\text{-}}\!\!\nearrow & (Y + X) + F(F^{\infty}(Y + X)) & \quad (5) \\
\raise2pt{\text{-}}\!\!\nearrow & \cong \Big\downarrow \scriptstyle{\zeta_{Y+X}^{-1}} & \\
X \xrightarrow{\quad\quad e \quad\quad} & F^{\infty}(Y + X) &
\end{array}
$$

This says that if we decompose the terms of e using the final coalgebra map, then we do not get variables from X.

Theorem 19 ([2]) *Each guarded ∞-equation has a unique solution.*

Proof. Assume that a guarded ∞-equation $e\colon X \to F^\infty(Y + X)$ factors as $\zeta_{Y+X}^{-1} \circ (\kappa_1+\mathrm{id}) \circ g$, for a map $g\colon X \to Y+F(F^\infty(Y+X))$ like in (5). In order to find a solution one first defines, like in the proof of Lemma 14, an auxiliary map $h\colon F^\infty(Y + X) + F^\infty(Y) \to F^\infty(Y)$ by coinduction, via an appropriate structure map on the left-hand-side below. Like in the proof of Lemma 14, on one of the +-components (the second) this structure map only unravels, while on the other it applies the guard g to the recursive variables from X.

$$
\begin{array}{ccc}
Y + F(F^\infty(Y + X) + F^\infty(Y)) & \xrightarrow{\ \mathrm{id}_Y + F(h)\ } & Y + F(F^\infty(Y)) \\
\big\uparrow{\scriptstyle [\mathrm{id} + F(\kappa_1),\, (\mathrm{id} + F(\kappa_2)) \circ \zeta_Y]} & & \\
(Y + F(F^\infty(Y + X))) + F^\infty(Y) & & \\
\big\uparrow{\scriptstyle [[\kappa_1, g],\, \kappa_2] + \mathrm{id}} & & \cong\big\downarrow{\scriptstyle \zeta_Y} \\
((Y + X) + F(F^\infty(Y + X))) + F^\infty(Y) & & \\
\big\uparrow{\scriptstyle \zeta_{Y+X} + \mathrm{id}} & & \\
F^\infty(Y + X) + F^\infty(Y) & \xrightarrow{\ \ \ h\ \ \ } & F^\infty(Y)
\end{array}
$$

The proof proceeds by showing that $h \circ \kappa_2$ is the identity. The unique solution is then obtained as $\mathsf{sol}(e) = h \circ \kappa_1 \circ \eta \circ \kappa_2\colon X \to Y + X \to F^\infty(Y + X) \to F^\infty(Y + X) + F^\infty(Y) \to F^\infty(Y)$. $\square$

6 ∞-equations and solutions as λ-equations and solutions

In this section we put previous results together. We start by fixing an object $Y \in \mathbb{C}$, and defining the associated functors $G^Y, T^Y\colon \mathbb{C} \to \mathbb{C}$ given by

$$G^Y(X) = Y + F(X) \qquad T^Y(X) = F^\infty(Y + X).$$

Why do we choose these functors? Well, a guard $X \to Y + F(F^\infty(Y + X))$ like in (5) is now simply a $G^Y T^Y$-coalgebra. We like to understand it as a λ-equation, in order to fit the ∞-equations in the framework of λ-equations. The first requirement is thus to establish the appropriate monad and distribution structure for G^Y and T^Y.

It is not hard to see that T^Y is again a monad—formally, via a general dis-

tributive law monads—with unit and multiplication:

$$\eta_X^Y = \eta_{Y+X}^\infty \circ \kappa_2 \qquad\qquad : X \longrightarrow Y + X \longrightarrow F^\infty(Y + X)$$
$$\mu_X^Y = \mathsf{subst}([\eta_{Y+X}^\infty \circ \kappa_1, \mathrm{id}]) : F^\infty(Y + F^\infty(Y + X)) \longrightarrow F^\infty(Y + X).$$

For convenience we shall drop the superscript Y whenever confusion is unlikely.

Next we note that T^Y is isomorphic to $(G^Y)^\infty$, since each $(G^Y)^\infty(X)$ forms by construction the final coalgebra for the mapping:

$$X + G^Y(-) = X + (Y + F(-)) \cong (Y + X) + F(-).$$

Hence $(G^Y)^\infty(X) \cong F^\infty(Y + X) = T^Y(X)$. Proposition 17 then yields the required distributive law. The next lemma describes it concretely.

Lemma 20 *In the above situation Proposition 17 yields a distributive law*

$$T^Y G^Y \xrightarrow{\quad \lambda^Y \quad} G^Y T^Y$$

for each $Y \in \mathbb{C}$. Ommitting the superscript Y, its components are maps of the form:

$$F^\infty(Y + (Y + F(X))) \xrightarrow{\quad \lambda_X \quad} Y + F(F^\infty(Y + X))$$

Via the two obvious natural transformations $\kappa_2 \colon F \Rightarrow G^Y$ and $F^\infty(\kappa_2) \colon F^\infty \Rightarrow T^Y$ we get a commuting diagram of distributive laws:

$$
\begin{array}{ccc}
F^\infty F & \longrightarrow & T^Y G^Y \\
\lambda^\infty \downarrow & & \downarrow \lambda \\
F F^\infty & \longrightarrow & G^Y T^Y
\end{array}
$$

Proof. The distributive law can be described as composite:

$$T^Y G^Y \cong (G^Y)^\infty G^Y \xrightarrow{\text{Proposition 17}} G^Y (G^Y)^\infty \cong G^Y T^Y$$

We shall construct this λ_X explicitly. By first applying the final coalgebra map we get:

$$F^\infty(Y + (Y + FX)) \xrightarrow[\cong]{\quad \zeta \quad} (Y + (Y + FX)) + FF^\infty(Y + (Y + FX))$$

The component on the left of the main $+$ on the right-hand-side readily gives a map to the required target, namely:

$$Y + (Y + FX) \xrightarrow{\quad [\kappa_1, \mathrm{id} + F(\eta_{X+Y}^\infty \circ \kappa_2)] \quad} Y + F(F^\infty(Y + X))$$

24

For the component on the right we have to do more work. We are done if we can find a map $F^\infty(Y + (Y + FX)) \to F^\infty(Y + X)$. Such a map can be obtained via substitution from:

$$Y + (Y + FX) \xrightarrow{[\eta^\infty_{Y+X} \circ \kappa_1, [\eta^\infty_{Y+X} \circ \kappa_1, \sigma^\infty_{Y+X} \circ F(\kappa_2)]]} F^\infty(Y + X)$$

Putting the decomposition via ζ and the two parts of a cotuple together, we obtain the following complicated expression for the resulting distributive law $F^\infty(Y + (Y + F(X))) \to Y + F(F^\infty(Y + X))$.

$$\lambda_X = [\,[\kappa_1, \mathrm{id} + F(\eta^\infty_{X+Y} \circ \kappa_2)],$$
$$\kappa_2 \circ F(\mathsf{subst}([\eta^\infty_{Y+X} \circ \kappa_1, [\eta^\infty_{Y+X} \circ \kappa_1, \sigma^\infty_{Y+X} \circ F(\kappa_2)]]))\,] \circ \zeta_{Y+(Y+FX)}.$$

It is not hard to check that the distributive laws are preserved, as claimed at the end of the lemma. □

Lemma 21 *For each $Y \in \mathbb{C}$, the object $F^\infty(Y)$ carries a final λ^Y-bialgebra structure:*

$$T^Y(F^\infty(Y)) \xrightarrow{\xi_Y} F^\infty(Y) \xrightarrow[\cong]{\zeta_Y} G^Y(F^\infty(Y))$$
$$\|\qquad\qquad\qquad\qquad\qquad\|$$
$$F^\infty(Y + F^\infty(Y)) \qquad\qquad Y + F(F^\infty(Y))$$

where $\xi^Y = \mathsf{subst}([\eta^\infty_Y, id])$.

Proof. By Lemma 3 there is on $F^\infty(Y)$ a unique Eilenberg-Moore algebra structure $T^Y(F^\infty(Y)) \to F^\infty(Y)$ forming a final λ^Y-bialgebra. We establish that it is of the form $\xi_Y = \mathsf{subst}([\eta^\infty_Y, id])$ by checking that this ξ_Y satisfies the

25

defining equation in Lemma 3. We shall drop superscripts as usual.

$$G(\xi_Y) \circ \lambda_{F\infty Y} \circ T(\zeta_Y)$$

$$= G(\xi_Y) \circ [\_,\_] \circ \zeta_{Y+(Y+FF\infty Y)} \circ F^\infty(\mathrm{id} + \zeta_Y)$$

by definition of λ and of T

$$= G(\xi_Y) \circ [\_,\_] \circ ((\mathrm{id} + \zeta_Y) + FF^\infty(\mathrm{id} + \zeta_Y)) \circ \zeta_{Y+F\infty Y}$$

by definition of F^∞ on morphisms

$$= (\mathrm{id} + F(\xi_Y)) \circ [\,[\kappa_1, \mathrm{id} + F(\eta^\infty_{Y+F\infty Y} \circ \kappa_2)] \circ (\mathrm{id} + \zeta_Y),$$
$$\kappa_2 \circ F(\mathsf{subst}(\_)) \circ FF^\infty(\mathrm{id} + \zeta_Y)\,] \circ \zeta_{Y+F\infty Y}$$

by further expansion of the definition of λ

$$= [\,[\kappa_1, (\mathrm{id} + F(\xi_Y \circ \eta^\infty_{Y+F\infty Y} \circ \kappa_2)) \circ \zeta_Y],$$
$$\kappa_2 \circ F(\xi_Y \circ \mathsf{subst}(\_) \circ F^\infty(\mathrm{id} + \zeta_Y))\,] \circ \zeta_{Y+F\infty Y}$$

by a simple calculation with cotuples

$$= [\,[\kappa_1, (\mathrm{id} + F(\mathrm{id})) \circ \zeta_Y],$$
$$\kappa_2 \circ F(\mathsf{subst}(\xi_Y \circ \_ \circ (\mathrm{id} + \zeta_Y)))\,] \circ \zeta_{Y+F\infty Y}$$

by definition of ξ and Lemma 15

$$\overset{(*)}{=} [\,[\kappa_1, \zeta_Y],$$
$$\kappa_2 \circ F(\mathsf{subst}([\eta^\infty_Y, [\eta^\infty_Y, \tau^\infty_Y]] \circ (\mathrm{id} + \zeta_Y)))\,] \circ \zeta_{Y+F\infty Y}$$

see below

$$= [\,[\kappa_1, \zeta_Y],$$
$$\kappa_2 \circ F(\mathsf{subst}([\eta^\infty_Y, \mathrm{id}]))\,] \circ \zeta_{Y+F\infty Y}$$

by definition of η, τ

$$= [\,\zeta_Y \circ [\eta^\infty_Y, \mathrm{id}],$$
$$\zeta_Y \circ \tau^\infty_Y \circ F(\xi_Y)\,] \circ \zeta_{Y+F\infty Y}$$

again by definition of η, τ and also of ξ

$$= \zeta_Y \circ [\,[\eta^\infty_Y, \mathrm{id}], \tau^\infty_Y \circ F(\xi_Y)\,] \circ \zeta_{Y+F\infty Y}$$

$$= \zeta_Y \circ \xi_Y$$

by Lemma 15 (5).

The marked step $(*)$ in this calculation is explained as follows.

$$\xi_Y \circ \sigma^\infty_{Y+F\infty Y} \circ F(\kappa_2)$$
$$= \mathsf{subst}([\eta^\infty_Y, \mathrm{id}]) \circ \tau^\infty_{Y+F\infty Y} \circ F(\eta^\infty_{Y+F\infty Y}) \circ F(\kappa_2) \quad \text{by definition of } \xi, \sigma$$
$$= \tau^\infty_Y \circ F(\mathsf{subst}([\eta^\infty_Y, \mathrm{id}])) \circ F(\eta^\infty_{Y+F\infty Y}) \circ F(\kappa_2) \quad \text{by Lemma 14}$$
$$= \tau^\infty_Y \circ F([\eta^\infty_Y, \mathrm{id}]) \circ F(\kappa_2)$$
$$= \tau^\infty_Y. \qquad\qquad\qquad \square$$

We are finally in a position to see that ∞-equations and solutions are a special case of λ-equations and solutions. This is our main result.

Theorem 22 *Let $F\colon \mathbb{C} \to \mathbb{C}$ be a functor with final coalgebra $F^\infty(X) \xrightarrow{\;\cong\;} X + F(F^\infty(X))$. Then:*

(1) *A guard $g\colon X \to Y + F(F^\infty(Y + X))$ for an ∞-equation $e\colon X \to F^\infty(Y + X)$ is a λ^Y-equation, for the distributive law λ^Y from Lemma 20.*

(2) *A solution $\mathsf{sol}(e)\colon X \to F^\infty(Y)$ of a guarded ∞-equation e is the same thing as a solution of its guard g—as a λ^Y-equation—in the final λ^Y-bialgebra of Lemma 21.*

Proof. The first point is obvious, so we concentrate on the second one. We assume that we can write the guarded ∞-equation $e\colon X \to F^\infty(Y + X)$ as $e = \zeta^{-1}_{Y+X} \circ (\kappa_1 + \mathrm{id}) \circ g$, like in (5), where $g\colon X \to Y + F(F^\infty(Y + X))$ is the guard (or λ-equation) and ζ is as usual the final coalgebra. We observe for a map $f\colon X \to F^\infty(Y)$,

f is a solution of the λ-equation g (see Definition 5)

$\Longleftrightarrow \zeta_Y \circ f = G(\xi_Y) \circ GT(f) \circ g$

$\Longleftrightarrow f = \zeta^{-1}_Y \circ G(\xi_Y) \circ GT(f) \circ g$

$\quad = [\eta^\infty_Y, \tau^\infty_Y] \circ (\mathrm{id} + F(\xi_Y)) \circ (\mathrm{id} + FF^\infty(\mathrm{id} + f)) \circ g$

$\qquad$ by definition of η, τ and of G, T

$\quad = [\eta^\infty_Y, \tau^\infty_Y \circ F(\xi_Y) \circ FF^\infty(\mathrm{id} + f)] \circ g$

$\quad = [\eta^\infty_Y, \tau^\infty_Y \circ F(\mathsf{subst}([\eta^\infty_Y, \mathrm{id}]) \circ F^\infty(\mathrm{id} + f))] \circ g$

$\qquad$ by definition of ξ

$\quad = [\eta^\infty_Y, \tau^\infty_Y \circ F(\mathsf{subst}([\eta^\infty_Y, \mathrm{id}] \circ (\mathrm{id} + f)))] \circ g$

$\qquad$ by Lemma 15 (2)

$\quad = [\eta^\infty_Y, \mathsf{subst}([\eta^\infty_Y, f]) \circ \tau^\infty_{Y+X}] \circ g$

$\qquad$ by Lemma 14

$\quad = \mathsf{subst}([\eta^\infty_Y, f]) \circ [\eta^\infty_{Y+X} \circ \kappa_1, \tau^\infty_{Y+X}] \circ g$

$\quad = \mathsf{subst}([\eta^\infty_Y, f]) \circ \zeta^{-1}_{Y+X} \circ (\kappa_1 + \mathrm{id}) \circ g$

$\qquad$ by definition of η, τ

$\quad = \mathsf{subst}([\eta^\infty_Y, f]) \circ e$

$\Longleftrightarrow f$ is a solution of the ∞-equation e (see Definition 18). $\qquad\square$

7 Conclusion

We have illustrated the use of distributive laws in recursive equations (especially for languages) and have unified the area by showing that one notion developed in [2] (following [20]) is an instance of a more general notion from [5,17,26] based on distributive laws.

Acknowledgments

References

[1] P. Aczel. *Non-well-founded sets*. CSLI Lecture Notes 14, Stanford, 1988.

[2] P. Aczel, J. Adámek, S. Milius, and J. Velebil. Infinite trees and completely iterative theories: a coalgebraic view. *Theor. Comp. Sci.*, 300 (1-3):1–45, 2003.

[3] P. Aczel, J. Adámek, and J. Velebil. A coalgebraic view of infinite trees and iteration. In A. Corradini, M. Lenisa, and U. Montanari, editors, *Coalgebraic Methods in Computer Science*, number 44 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2001.

[4] M. Barr and Ch. Wells. *Toposes, Triples and Theories*. Springer, Berlin, 1985. Revised and corrected version available from URL: `www.cwru.edu/artsci/math/wells/pub/ttt.html`.

[5] F. Bartels. *On generalised coinduction and probabilistic specification formats. Distributive laws in coalgebraic modelling*. PhD thesis, Free Univ. Amsterdam, 2004.

[6] J. Barwise and L.S. Moss. *Vicious Circles: On the Mathematics of Non-wellfounded Phenomena*. CSLI Lecture Notes 60, Stanford, 1996.

[7] J. Beck. Distributive laws. In B. Eckman, editor, *Seminar on Triples and Categorical Homolgy Theory*, number 80 in Lect. Notes Math., pages 119–140. Springer, Berlin, 1969.

[8] J.R.B. Cockett. Introduction to distributive categories. *Math. Struct. in Comp. Sci.*, 3:277–307, 1993.

[9] J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.

[10] C.C. Elgot. Monadic computation and iterative algebraic theories. In H.E. Rose and J.C. Shepherson, editors, *Logic Colloquium '73*, pages 175–230, Amsterdam, 1975. North-Holland.

[11] C.C. Elgot, S.L. Bloom, and R. Tindell. The algebraic structure of rooted trees. *Journ. Comp. Syst. Sci*, 16:361–399, 1978.

[12] B. Jacobs. Relating two approaches to coinductive solution of recursive equations. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, number 106 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2004.

[13] B. Jacobs. Trace semantics for coalgebras. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, number 106 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2004.

[14] B. Jacobs. A bialgebraic review of regular expressions, deterministic automata and languages. Techn. Rep. ICIS-R05003, Inst. for Computing and Information Sciences, Radboud Univ. Nijmegen, 2005.

[15] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:222–259, 1997.

[16] P.T. Johnstone. Adjoint lifting theorems for categories of algebras. *Bull. London Math. Soc.*, 7:294–297, 1975.

[17] M. Lenisa. From set-theoretic coinduction to coalgebraic coinduction: some results, some problems. In B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science*, number 19 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 1999.

[18] M. Lenisa, J. Power, and H. Watanabe. Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads. In H. Reichel, editor, *Coalgebraic Methods in Computer Science*, number 33 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2000.

[19] S. Milius. On iterable endofunctors. In R. Blute and P. Selinger, editors, *Category Theory and Computer Science 2002*, number 69 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2003.

[20] L.S. Moss. Parametric corecursion. *Theor. Comp. Sci.*, 260(1-2):139–163, 2001.

[21] D.E. Muller and P.E. Schupp. Alternating automata on infinite trees. *Theor. Comp. Sci.*, 54(2/3):267–276, 1987.

[22] K.I. Rosenthal. *Quantales and their applications*. Number 234 in Pitman Research Notes in Math. Longman Scientific & Technical, 1990.

[23] J. Rutten. Behavioural differential equations: a coinductive calculus of streams, automata, and power series. *Theor. Comp. Sci.*, 308:1–53, 2003.

[24] D. Turi. *Functorial operational semantics and its denotational dual*. PhD thesis, Free Univ. Amsterdam, 1996.

[25] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Logic in Computer Science*, pages 280–291. IEEE, Computer Science Press, 1997.

[26] T. Uustalu, V. Vene, and A. Pardo. Recursion schemes from comonads. *Nordic Journ. Comput.*, 8(3):366–390, 2001.