Math. Struct. in Comp. Science (2003), *vol.* 13, *pp.* 321–348. © 2003 Cambridge University Press
DOI: 10.1017/S0960129502003900 Printed in the United Kingdom

Generalised coinduction

FALK BARTELS[†]

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
http://www.cwi.nl/~bartels/

Received 31 August 2001; revised 6 February 2002

We introduce the λ-*coiteration* schema for a distributive law λ of a functor T over a functor F. Parameterised by T and λ it generalises the basic coiteration schema uniquely characterising functions into a final F-coalgebra. Furthermore, the same parameters are used to generalise the categorical notion of a bisimulation to that of a λ-*bisimulation*, while still giving rise to a proof technique for bisimilarity. We first present a theorem showing the validity of the resulting definition and proof principles for categories with countable coproducts.

Our approach gives a unifying categorical presentation and justification of several extensions of the basic coinduction schemata that have been treated separately before, and some only for specific types of system. As examples, the duals of primitive recursion and course-of-value iteration, which are known extensions of coiteration, arise as instances of our framework.

Moreover, we derive schemata involving auxiliary operators definable with GSOS-style specifications such as addition of streams, regular operators on languages, or parallel and sequential composition of processes. The argument is based on a variation of the theory in the setting of monads and copointed functors. The schemata justify guarded recursive definitions and an up-to-context proof technique for operators of the type mentioned. The latter can ease bisimilarity proofs considerably.

1. Introduction

Since around the early nineties, the notion of an F-*coalgebra* for a functor F on some category C has been more widely used to formally model dynamical systems in computer science. These include processes or automata, but also datatypes such as infinite streams or trees of possibly unbounded depth (see, for example, the introductions by Jacobs and Rutten (Jacobs and Rutten 1996; Rutten 2000b)). The instance of the functor F determines the type of behaviour under consideration. This uniform description of different kinds of system allowed for an abstract formulation of definition and proof principles that had been studied separately for various applications before. For instance, definitions of *bisimilarity* have been proposed for numerous types of systems to model behavioural equivalence of states. Many of these have been shown to be the corresponding instances of an abstract notion of bisimilarity defined in terms of F-*bisimulations* (Aczel and Mendler 1989).

A *final* F-*coalgebra*, if it exists, provides a canonical domain for behaviours of the type F. The term *coinduction* is used for the definition and proof techniques for such a

[†] The research reported here was part of the NWO project ProMACS.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

system. To be more precise, we will be talking about the *coiteration schema* if the finality of a coalgebra is exploited directly as a definition principle. And, as the *coinduction proof principle*, we refer to a technique that uses F-bisimulations to prove equality among the states of the final F-coalgebra. This is made possible by the fact that bisimilarity and equality coincide on its state space.

Unfortunately, these basic definition and proof principles are often too rigid to cover given examples nicely. Many interesting functions into the carrier of a final coalgebra are not coiterative and many statements about behavioural equivalence require bisimulations that are difficult to exhibit and check. Therefore various extensions have been proposed, again initially for particular kinds of systems. An example are process calculi, where infinite processes are usually specified by systems of recursive equations involving given composition operators, and often bisimilarity proofs are simplified using bisimulations up-to (see, for example, Milner (1989)). On the categorical level, two particular extensions have been formulated that arise as dualisations of schemata from the algebraic world, namely the duals of primitive recursion and course-of-value iteration (see, for example, Uustalu and Vene (1999)). The first is often referred to as *primitive corecursion*.

Our aim is to give a broader categorical account of extended coinduction principles covering in one approach and in greater generality several of the aforementioned aspects. We do so by introducing a parameterised framework that can be instantiated to yield several coinductive definition and proof schemata. It is based on the following idea.

Roughly speaking, the standard way to turn the elements of a set X into states of a dynamical system of the type F (possibly showing an infinite behaviour) is to specify in one go, for each inhabitant, direct observations and successors in X, that is, to declare an F-coalgebra structure on X. Since the successors are taken from the same set, the same specification can be applied to them to reveal the second layer of observations, and so on. Our approach generalises this idea as follows: for another functor T, which enters as a new parameter, the successor states are appointed in $\mathrm{T}X$ instead of X (in other words, we now ask for an FT-coalgebra structure). This is intended to add new possibilities for the following stages by choosing T such that $\mathrm{T}X$ is 'richer' than X alone. For the observations to be continued with these successors, one needs to know how to lift the original specification to this new set of states. This information takes the shape of a *distributive law* λ of T over F (see, for example, Lenisa *et al.* (2000)), in which the framework is also parametric.

In essence, the specification for X is taken as a kernel around which a larger system is constructed in a systematic manner determined by T and λ. The advantage of this approach is that propositions involving the whole system can sometimes be stated in terms of this smaller kernel alone, so that the construction can be left implicit. In particular, certain homomorphisms out of the generated coalgebra can be constructed from functions out of its kernel that we call *homomorphisms up-to*. On the one hand, this lets us introduce the *λ-coiteration schema*, which defines arrows into the final F-coalgebra based on the unique existence of homomorphisms up-to. It allows one also to characterise directly such arrows for sets X that do not carry an appropriate coalgebra structure themselves. On the other hand, we define the notion of a *λ-bisimulation* based on homomorphisms up-to, and prove a *λ-coinduction proof principle* that enables

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

bisimilarity proofs on the basis of simple relations that are the 'kernels' of more complex bisimulations.

The main statements of this paper express that the above principles work under different additional assumptions, which are needed to show that the large system can actually be constructed inside the category. The basic theorem requires the existence of countable coproducts. Later we also present a variant where the functor T comes as a monad, the functor F is taken from a copointed functor, and the distributive law λ is assumed to interact nicely with this additional structure (that is, λ should be a distributive law of the monad over the copointed functor, again see Lenisa *et al.* (2000)).

As a trivial instance of the new framework, one recovers the basic coiteration schema and the standard coinduction proof principle. More interesting settings for T and λ yield the known schemata of primitive corecursion and the dual of course-of-value iteration mentioned above. In some more detail, we explain another instance of the framework, which deals with certain sets of auxiliary operators, like, for example, parallel and sequential composition for labelled transition systems. More precisely, it can handle such operators that are definable by a format introduced by Turi and Plotkin (Turi and Plotkin 1997) as a categorical generalisation of the known GSOS rule format (Bloom *et al.* 1995). On the one hand, one obtains definition principles guaranteeing unique solutions for (guarded) recursive equations involving these operators, and on the other hand, this leads to a proof principle up-to-context for contexts built from them.

1.1. *Related work*

A first proposal for a parameterised description covering several extended coinduction principles on a categorical level was made by Lenisa in the course of her comparison of set-theoretic and coalgebraic (categorical) formulations of coinduction (Lenisa 1999).

Recently, but independently of us, Pardo, Uustalu and Vene introduced a schema for inductive definitions parametric in a comonad over which the algebra functor distributes (Uustalu *et al.* 2001). It turns out to be the dual of our λ-coiteration schema in a version involving monads.

For labelled transition systems, the bisimulation up-to technique has been put into a systematic framework by Sangiorgi (Sangiorgi 1998). In particular, he investigates conditions under which bisimulations up-to-context yield a sound proof principle.

Our presentation uses the notion of a distributive law of a functor T over a functor F, where T may also come with additional structure, namely as a pointed functor or a monad, and the functor F may be taken from a copointed functor. These cases have been presented systematically by Lenisa, Power and Watanabe (Power and Watanabe 1999; Lenisa *et al.* 2000). Bialgebras for a distributive law have been taken from the work of Turi and Plotkin (Turi and Plotkin 1997).

A comparison with some of these papers is given in Section 7.

1.2. *Overview of the paper*

Following this introduction, we recall the basic coinduction principles in Section 2. In Sections 3 and 4, we develop basic versions of the definition schema of λ-coiteration and

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

the λ-bisimulation proof principle. In Section 5 an instance involving auxiliary operators is studied in more detail and requires a reformulation of the principles in a more complex setting. Some further instances of the framework are listed in Section 6 before we conclude by relating our schemata to the work of other authors and mentioning directions for future investigations.

The work reported here was presented at the Coalgebraic Methods in Computer Science 2001 Workshop (Bartels 2001).

1.3. *Notation*

We write C for some category. The category of sets and total functions, to which all our informal explanations refer, is denoted by Set.

For objects X_1, X_2, and X_i $(i \in I)$, we write $X_1 \times X_2$ and $X_1 + X_2$ for the binary categorical product and coproduct and $\coprod_{i \in I} X_i$ for an arbitrary coproduct. The corresponding projections and injections are denoted by $\pi_j : X_1 \times X_2 \to X_j$ and $\mathtt{in}_j : X_j \to X_1 + X_2$ for $j = 1, 2$, and by $\mathtt{in}_j : X_j \to \coprod_{i \in I} X_i$ for $j \in I$. The pairing of two functions $f_j : Y \to X_j$ $(j = 1, 2)$ given by the universal property of the product is denoted by $\langle f_1, f_2 \rangle : Y \to X_1 \times X_2$. Dually, case analysis is written as $[g_1, g_2] : X_1 + X_2 \to Y$ and $[g_j]_{j \in I} : \coprod_{i \in I} X_i \to Y$ for $g_j : X_j \to Y$ $(j = 1, 2$ or $j \in I)$. For $f_i : X_i \to Y_i$ $(i = 1, 2)$ we further abbreviate $\langle f_1 \circ \pi_1, f_2 \circ \pi_2 \rangle$ to $f_1 \times f_2$ and $[\mathtt{in}_1 \circ f_1, \mathtt{in}_2 \circ f_2]$ to $f_1 + f_2$.

We use $\mathrm{T}, \mathrm{F} : \mathsf{C} \to \mathsf{C}$ to denote two functors, write Id for the identity functor, and A again for the constant functor associated to an object A. Furthermore, we use $\mathrm{F} \times \mathrm{T}$ and $\mathrm{F} + \mathrm{T}$ for the product and coproduct of two functors, $\mathscr{P}$ for the power set functor, and $(.)^A$ for exponentiation with an object A. In diagrams we use double arrows for identities as well as morphisms in the functor category (that is, natural transformations).

2. Standard coinduction principles

In this section we will recall the definition of an algebra and a (final) coalgebra of a functor, the coiteration definition schema and the bisimulation proof principle. For detailed expositions we advise the reader again to take a look at the overview papers of Jacobs and Rutten (Jacobs and Rutten 1996; Rutten 2000b).

Definition 2.1 (T-algebra, F-coalgebra). An *algebra for the functor* T, or T-*algebra* for short, is a pair $\langle X, \beta \rangle$ where X is an object of C and $\beta : \mathrm{T}X \to X$ is an arrow. We will sometimes call X and β the *carrier* and *operation* of the algebra. Dually, an F-*coalgebra* is a pair $\langle X, \alpha \rangle$ where the operation is an arrow $\alpha : X \to \mathrm{F}X$.

Generally, algebra operations can be seen as a means for *constructing* elements of their carrier. The operation of a coalgebra – also called *destruction* or *unfolding* elsewhere – gives us information about its states. For this information $\alpha(x) \in \mathrm{F}X$, we will in our explanations sometimes distinguish between the *observation* the state $x \in X$ allows and its *dynamics*. The first is intended to describe the part of $\alpha(x)$ that does not involve elements from X, like attributes, whereas with the second we want to focus on these successor states.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

Definition 2.2 (Homomorphism). An arrow $h : X \to Y$ is a T-*algebra homomorphism* from one T-algebra $\langle X, \beta_X \rangle$ to another T-algebra $\langle Y, \beta_Y \rangle$ if it makes diagram (a) below commute. Similarly, it is an F-*coalgebra homomorphism* from one F-coalgebra $\langle X, \alpha_X \rangle$ to another F-coalgebra $\langle Y, \alpha_Y \rangle$ if it makes diagram (b) commute.

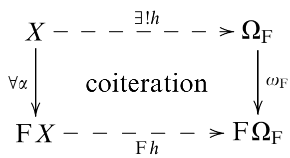
We will often just talk about homomorphisms when their type is clear from the context. T-algebras and F-coalgebras, together with the corresponding homomorphisms form the categories $\mathsf{Alg}^{\mathrm{T}}$ and $\mathsf{Coalg}_{\mathrm{F}}$, respectively.

Definition 2.3 (Final F-coalgebra). A *final* F-*coalgebra* is a final object in $\mathsf{Coalg}_{\mathrm{F}}$, that is, a coalgebra – usually denoted here by $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ – such that there is exactly one homomorphism from every F-coalgebra to it.

Example 2.4 (Stream systems). Consider the Set-functor $\mathrm{F}X := I\!R \times X$. Its coalgebras are of the shape $\langle X, \langle o, s \rangle \rangle$ for a set X and two functions $o : X \to I\!R$ and $s : X \to X$. That is, each state $x \in X$ gives rise to an *observation* $o(x) \in I\!R$ and a *successor state* $s(x) \in X$. We will call such a coalgebra a *stream system*, because, by assuming that we have access to its states only *via* its operation, all we can learn about an element $x \in X$ is the infinite stream $\langle o(x), o(s(x)), o(s^2(x)), \ldots \rangle \in I\!R^\omega$ of observations for all the elements consecutively reachable from it.

In fact, the set of streams of real numbers $I\!R^\omega$ itself forms a stream system when equipped with the F-coalgebra structure $\langle \mathtt{head}, \mathtt{tail} \rangle$, where for a stream $\sigma = \langle s_0, s_1, \ldots \rangle$ the observation is given by its first element $\mathtt{head}(\sigma) := s_0$ and the successor by the stream that remains after removing it, $\mathtt{tail}(\sigma) := \langle s_1, s_2, \ldots \rangle$. Moreover, this system can be shown to be final (see Rutten (2000a) for a proof). We often write the stream σ as $\langle s_0 : \sigma' \rangle$ to mean that $\mathtt{head}(\sigma) = s_0$ and $\mathtt{tail}(\sigma) = \sigma'$.

Every F-coalgebra operation α on an object X determines an arrow from it into the carrier of the final coalgebra, namely the unique homomorphism from $\langle X, \alpha \rangle$ to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$. Such an arrow is then called the *coiterative arrow defined* (*or coinduced*) *by* α:



Example 2.5 (Coiteration for streams). The coiteration schema for stream systems from Example 2.4 states that for every pair of functions $o : X \to I\!R$ and $s : X \to X$ there is a unique function $h : X \to I\!R^\omega$ satisfying

$$\mathtt{head}(h(x)) = o(x) \quad \text{and} \quad \mathtt{tail}(h(x)) = h(s(x)).$$

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

As one example, for $\mathscr{A} := \{f : I\!R \to I\!R \mid f \text{ analytic in } 0\}$ we might want to define the mapping $\mathscr{T} : \mathscr{A} \to I\!R^{\omega}$ sending such a function f to the stream $\langle f(0), f'(0), f''(0), \ldots \rangle$ of its derivatives of all orders at 0 (that is, its Taylor coefficients). It arises as the coiterative arrow from the F-coalgebra $\langle \mathscr{A}, \alpha \rangle$ to $\langle I\!R^{\omega}, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$, where $\alpha(f) := \langle f(0), f' \rangle$. In other words, $\mathscr{T}$ is the unique function satisfying

$$\mathtt{head}(\mathscr{T}(f)) = f(0) \quad \text{and} \quad \mathtt{tail}(\mathscr{T}(f)) = \mathscr{T}(f').$$

Second, we would like to define a binary operation $\oplus$ on $I\!R^{\omega}$ compatible with the addition of analytic functions in the sense that for $f, g \in \mathscr{A}$ we require

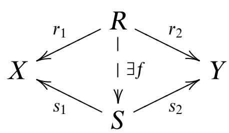
$$\mathscr{T}(f+g) = \mathscr{T}(f) \oplus \mathscr{T}(g). \tag{1}$$

We will show below that by setting $\alpha(\sigma, \tau') := \langle s_0 + t_0, \langle \sigma', \tau' \rangle \rangle$ for $\sigma = \langle s_0 : \sigma' \rangle$ and $\tau = \langle t_0 : \tau' \rangle$ the coiterative morphism $\oplus$ from $\langle I\!R^{\omega} \times I\!R^{\omega}, \alpha \rangle$ to $\langle I\!R^{\omega}, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ has this property. This definition amounts to saying that $\oplus$ is the unique operation satisfying

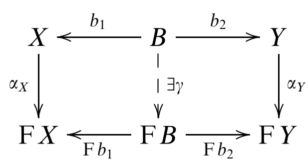
$$\mathtt{head}(\sigma \oplus \tau) = s_0 + t_0 \quad \text{and} \quad \mathtt{tail}(\sigma \oplus \tau) = \sigma' \oplus \tau'.$$

Many types of dynamical systems have been equipped with the notion of a *bisimulation* as a tool to define behavioural equivalence. For coalgebras of a functor, an abstract definition can be given that specialises to many of the concrete proposals. Our version is based on the notion of a span.

Definition 2.6 (Span). A *span* $\mathscr{R} = \langle R, r_1, r_2 \rangle$ between two C objects X and Y consists of an object R and two arrows $r_1 : R \to X$ and $r_2 : R \to Y$. A span between X and itself is called a span *on* X.
There is a preorder $\preceq$ of spans between the objects X and Y defined as $\langle R, r_1, r_2 \rangle \preceq \langle S, s_1, s_2 \rangle$ if and only if there is an arrow $f : R \to S$ such that both triangles in the following diagram commute:



Definition 2.7 (Bisimulation). A *bisimulation* between two F-coalgebras $\langle X, \alpha_X \rangle$ and $\langle Y, \alpha_Y \rangle$ is a span $\mathscr{B} = \langle B, b_1, b_2 \rangle$ between their carriers X and Y such that there is an F-coalgebra operation $\gamma : B \to \mathrm{F}B$ turning b_1 and b_2 into homomorphisms:



A bisimulation between an F-coalgebra $\langle X, \alpha \rangle$ and itself is called a bisimulation *on* $\langle X, \alpha \rangle$.

In Set one often considers only bisimulations that are relations, that is, spans $\langle R, \pi_1, \pi_2 \rangle$ for a relation $R \subseteq X \times Y$ (see, for example, Rutten (2000b)). We use the formulation based on spans because it generalises to other categories and is sometimes easier to work with. We will still often talk about bisimulation *relations*. This is justified by the observation that every span $\langle R, r_1, r_2\rangle$ in Set can be regarded as representing the image $\langle r_1, r_2\rangle[R] \subseteq X \times Y$. The order $\preceq$ of spans corresponds to relational inclusion of images. Furthermore, the image of a (span) bisimulation is a (relational) bisimulation (Rutten 2000b, Lemma 5.3).

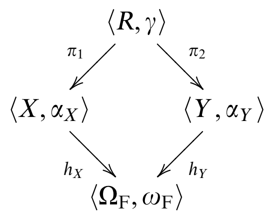
The category Set further allows us to talk about the *states* of a coalgebra, meaning the elements of its carrier (this is also true for many other – say Set-like – categories, but for simplicity we will not elaborate on this point here). Two such states s and t are usually called *bisimilar* (written $s \sim t$) if they are related by some bisimulation. Taking this notion to mean behavioural equivalence is supported by the fact that when a final F-coalgebra exists, bisimilar states are identified by the coiterative morphisms.

Theorem 2.8 (Coinduction proof principle). Let $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$ be two coalgebras for the Set-functor F. Let h_X and h_Y denote the coiterative morphisms from $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$ to a final F-coalgebra $\langle \Omega_F, \omega_F\rangle$. For $x \in X$ and $y \in Y$ we have

$$x \sim y \quad\Rightarrow\quad h_X(x) = h_Y(y).$$

In particular, for $p, q \in \Omega_F$, this means that $p \sim q$ implies $p = q$.

Proof. Let $R \subseteq X \times Y$ be a bisimulation between $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$ containing $\langle x, y\rangle$, and let the bisimulation property be witnessed by $\gamma : R \to \mathrm{F}R$. We get the diagram below in $\mathsf{Coalg}_\mathrm{F}$, which commutes by finality. This yields $h_X(x) = h_X(\pi_1(\langle x, y\rangle)) = h_Y(\pi_2(\langle x, y\rangle)) = h_Y(y)$, as required. The special case follows from the fact that the coiterative morphism from a final coalgebra to itself is the identity.



□

Example 2.9. As an example, we consider again the operator $\oplus$ defined coiteratively in Example 2.5. We can use the coinduction proof principle to prove that it indeed satisfies equation (1), because the relation

$$R := \{\langle \mathcal{T}(f+g), \mathcal{T}(f) \oplus \mathcal{T}(g)\rangle \mid f, g \in \mathcal{A}\} \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$$

can easily be shown to be a bisimulation on the final stream system.

3. Definition by λ-coiteration

The coiteration schema allows us to define a function $f : X \to \Omega_F$ by setting up an F-coalgebra structure α on X and taking f to be the unique homomorphism from the resulting coalgebra to $\langle \Omega_F, \omega_F\rangle$, given by finality. Unfortunately, for many interesting

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

functions $f : X \to \Omega_F$ there is no such α making f the coiterative morphism, or it may not be obvious from the given specification of f.

In this section we are first going to present an example of such a specification. Then we will formulate the pattern encountered more abstractly and – as the main theorems of this paper – state sufficient conditions for it to characterise uniquely arrows into the carrier of a final coalgebra. This yields a definition schema, which we call λ-*coiteration*.

3.1. *Example: multiplication of streams*

In Example 2.5 we designed the operation $\oplus$ such that when applied to the Taylor Series $\mathscr{T}(f)$ and $\mathscr{T}(g)$ of two functions $f, g \in \mathscr{A}$, it produced $\mathscr{T}(f+g)$, the Taylor Series of the sum of the functions. Similarly, we would now like to specify a multiplication $\otimes$ on $\mathbb{R}^\omega$ that agrees with the functional product $f \cdot g$ *via* the translation $\mathscr{T}$, that is, we would like to have

$$\mathscr{T}(f) \otimes \mathscr{T}(g) = \mathscr{T}(f \cdot g). \tag{2}$$

Computing the head and tail of the right-hand expression (using (1)), and generalising $\mathscr{T}(f) = \langle f(0) : \mathscr{T}(f') \rangle$ to $\sigma = \langle s_0 : \sigma' \rangle$ and $\mathscr{T}(g) = \langle g(0) : \mathscr{T}(g') \rangle$ to $\tau = \langle t_0 : \tau' \rangle$, yields

$$\texttt{head}(\sigma \otimes \tau) = s_0 \cdot t_0 \quad \text{and} \quad \texttt{tail}(\sigma \otimes \tau) = (\sigma \otimes \tau') \oplus (\sigma' \otimes \tau).$$

These two equations do not form a coiterative definition as in Example 2.5 because of the use of $\oplus$ in the expression for the tail. To get a better picture of the type of definition we have here, we set $X := \mathbb{R}^\omega \times \mathbb{R}^\omega$ and (for $o : X \to \mathbb{R}$, $s_l, s_r : X \to X$)

$$o(\sigma, \tau) := s_0 \cdot t_0, \quad s_l(\sigma, \tau) := \langle \sigma, \tau' \rangle, \quad \text{and} \quad s_r(\sigma, \tau) := \langle \sigma', \tau \rangle,$$

to get that $\otimes$ satisfies the above equations just when it fits into the following diagram:

$$\begin{array}{ccc}
X & \overset{\otimes}{\dashrightarrow} & \mathbb{R}^\omega \\
{\scriptstyle \phi := \langle o, \langle s_l, s_r \rangle \rangle} \downarrow & & \downarrow {\scriptstyle \langle \texttt{head}, \texttt{tail} \rangle} \\
\mathbb{R} \times (X \times X) & \underset{\mathrm{id}_{\mathbb{R}} \times (\oplus \circ (\otimes \times \otimes))}{\dashrightarrow} & \mathbb{R} \times \mathbb{R}^\omega
\end{array}$$

Furthermore, consider the functor $\mathrm{T} := \mathrm{Id} \times \mathrm{Id}$. It makes $\oplus$ a T-algebra operation on $\mathbb{R}^\omega$ and ϕ an FT-coalgebra operation on X, since we can express the object in the lower left-hand corner as $\mathrm{FT}X$ (remember that we are using $\mathrm{F} = \mathbb{R} \times \mathrm{Id}$). Moreover, the arrow on the bottom can be rewritten as $\mathrm{F}(\oplus \circ \mathrm{T} \otimes)$. Taken together, we can redraw the diagram above as follows (adding an arrow to illustrate the typing of $\oplus$):

$$\begin{array}{ccc}
 & & \mathrm{T}\mathbb{R}^\omega \\
 & & \downarrow {\scriptstyle \oplus} \\
X & \overset{\otimes}{\dashrightarrow} & \mathbb{R}^\omega \\
{\scriptstyle \phi} \downarrow & & \downarrow {\scriptstyle \langle \texttt{head}, \texttt{tail} \rangle} \\
\mathrm{FT}X & \underset{\mathrm{F}(\oplus \circ \mathrm{T} \otimes)}{\dashrightarrow} & \mathrm{F}\mathbb{R}^\omega
\end{array} \tag{3}$$

Does such an arrow $\otimes$ exist, and if so, is it unique? This situation will turn out to be an instance of a more general framework to be developed below.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

3.2. *The λ-coiteration definition schema*

To describe the format above more abstractly, the following notions are helpful.

Definition 3.1 ($\langle \mathrm{T}, \mathrm{F}\rangle$-bialgebra). A $\langle \mathrm{T}, \mathrm{F}\rangle$-*bialgebra* is a triple $\langle X, \beta, \alpha\rangle$ of an object X and two arrows $\beta : \mathrm{T}X \to X$ and $\alpha : X \to \mathrm{F}X$, that is, a T-algebra and an F-coalgebra operation on a common carrier. Given two $\langle \mathrm{T}, \mathrm{F}\rangle$-bialgebras $\langle X, \beta_X, \alpha_X\rangle$ and $\langle Y, \beta_Y, \alpha_Y\rangle$, a $\langle \mathrm{T}, \mathrm{F}\rangle$-*bialgebra homomorphism* from $\langle X, \beta_X, \alpha_X\rangle$ to $\langle Y, \beta_Y, \alpha_Y\rangle$ is an arrow $h : X \to Y$ that is both a T-algebra homomorphism from $\langle X, \beta_X\rangle$ to $\langle Y, \beta_Y\rangle$ and an F-coalgebra homomorphism from $\langle X, \alpha_X\rangle$ to $\langle Y, \alpha_Y\rangle$. As with T-algebras and F-coalgebras, $\langle \mathrm{T}, \mathrm{F}\rangle$-bialgebras and their homomorphisms form a category $\mathsf{Bialg}^{\mathrm{T}}_{\mathrm{F}}$.

Definition 3.2 (Homomorphism up-to). Let $\langle X, \phi\rangle$ be an FT-coalgebra, $\langle Y, \beta, \alpha\rangle$ be a $\langle \mathrm{T}, \mathrm{F}\rangle$-bialgebra, and $f : X \to Y$ be an arrow. Writing

$$f|^{\beta} := \beta \circ \mathrm{T}f : \mathrm{T}X \to Y$$

as depicted in diagram (a), we call f a *homomorphism up-to* from $\langle X, \phi\rangle$ to $\langle Y, \beta, \alpha\rangle$, if it makes diagram (b) commute (again we have added the arrow β to illustrate its typing).

The following straightforward statement about homomorphisms up-to will be useful later.

Lemma 3.3. Let $\langle X, \phi\rangle$ and $\langle X', \phi'\rangle$ be FT-coalgebras and let $\langle Y, \beta, \alpha\rangle$ and $\langle Y', \beta', \alpha'\rangle$ be $\langle \mathrm{T}, \mathrm{F}\rangle$-bialgebras.

(i) If f is a homomorphism up-to from $\langle X, \phi\rangle$ to $\langle Y, \beta, \alpha\rangle$ and h is a bialgebra homomorphism from $\langle Y, \beta, \alpha\rangle$ to $\langle Y', \beta', \alpha'\rangle$, then $h \circ f$ is a homomorphism up-to from $\langle X, \phi\rangle$ to $\langle Y', \beta', \alpha'\rangle$.

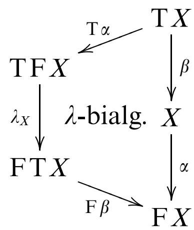
(ii) If g is an FT-coalgebra homomorphism from $\langle X', \phi'\rangle$ to $\langle X, \phi\rangle$ and f is a homomorphism up-to from $\langle X, \phi\rangle$ to $\langle Y, \alpha, \beta\rangle$, then $f \circ g$ is a homomorphism up-to from $\langle X', \phi'\rangle$ to $\langle Y, \alpha, \beta\rangle$.

The specification of the arrow $\otimes$ in (3) now amounts to asking it to be a homomorphism up-to from $\langle X, \phi\rangle$ to the $\langle \mathrm{T}, \mathrm{F}\rangle$-bialgebra $\langle \mathbb{R}^{\omega}, \oplus, \langle \mathtt{head}, \mathtt{tail}\rangle\rangle$ built from the final F-coalgebra (note that $\otimes|^{\oplus} := \oplus \circ \mathrm{T}\otimes$). Thus, we are looking for a setting in which any FT-coalgebra $\langle X, \phi\rangle$ uniquely determines a homomorphism up-to from $\langle X, \phi\rangle$ to a $\langle \mathrm{T}, \mathrm{F}\rangle$-bialgebra obtained by equipping the final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ with a T-algebra operation β. We will have to come up with a characterisation of suitable such β, since it is easy to find candidates for which this does not work. Our approach uses the notion of a bialgebra for a *distributive law*.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

Definition 3.4 ((Bialgebra for a) Distributive law). A natural transformation $\lambda : \mathrm{TF} \Rightarrow \mathrm{FT}$ is called a *distributive law* of the functor T over the functor F.

A *bialgebra for the distributive law* λ, or a λ*-bialgebra* for short, is a $\langle \mathrm{T}, \mathrm{F} \rangle$-bialgebra $\langle X, \beta, \alpha \rangle$ such that the following diagram commutes:



The full subcategory of $\mathsf{Bialg}_{\mathrm{F}}^{\mathrm{T}}$ containing all λ-bialgebras is denoted by λ-Bialg.

A major application of bialgebras for a distributive law λ in computer science has been given by Turi and Plotkin (Turi and Plotkin 1997), who used them as semantic models of programming languages with given operational rules. There the two functors represent program syntax and behaviour, and they come as a monad and a comonad, respectively. Additional coherence axioms, about the interaction of the extra monad and comonad structure with λ on the one hand, and with the algebra and coalgebra operations of the bialgebras on the other hand, are assumed (see Power and Watanabe (1999) for a structured account of this setting). Subsequently, distributive laws were also used in situations where less of the extra structure is given and – correspondingly – fewer coherence axioms are considered (Lenisa *et al.* 2000). For the moment we only treat plain functors and no coherence axioms, but we will also encounter some of them later.

Lemma 3.5 (λ-lifting). Given a distributive law λ of a functor T over a functor F, we can lift $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ to the functor $\mathrm{T}_\lambda : \mathsf{Coalg}_{\mathrm{F}} \to \mathsf{Coalg}_{\mathrm{F}}$ by setting

$$\mathrm{T}_\lambda \langle X, \alpha \rangle := \langle \mathrm{T}X, \lambda_X \circ \mathrm{T}\alpha \rangle \quad \text{and} \quad \mathrm{T}_\lambda h := \mathrm{T}h,$$

for any F-coalgebra $\langle X, \alpha \rangle$ and homomorphism h.

Proof. We need to show that for a homomorphism $h : \langle X, \alpha_X \rangle \to \langle Y, \alpha_Y \rangle$ we get a homomorphism $\mathrm{T}h : \langle \mathrm{T}X, \lambda_X \circ \mathrm{T}\alpha_X \rangle \to \langle \mathrm{T}Y, \lambda_Y \circ \mathrm{T}\alpha_Y \rangle$. This follows easily from the naturality of λ (see also Rutten (2000b, Theorem 15.3)). □

Note that the condition for a $\langle \mathrm{T}, \mathrm{F} \rangle$-bialgebra $\langle X, \beta, \alpha \rangle$ to be a λ-bialgebra is equivalent to saying that β is a homomorphism from $\mathrm{T}_\lambda \langle X, \alpha \rangle$ to $\langle X, \alpha \rangle$. In other words, it is equivalent to $\langle \langle X, \alpha \rangle, \beta \rangle$ being a T_λ-algebra in $\mathsf{Coalg}_{\mathrm{F}}$. With this observation we easily get the following statement.

Lemma 3.6. Let λ be a distributive law of T over F. For a final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ there exists a unique T-algebra operation β_λ on Ω_{F} such that $\langle \Omega_{\mathrm{F}}, \beta_\lambda, \omega_{\mathrm{F}} \rangle$ is a λ-bialgebra. Furthermore, this λ-bialgebra is final.

Proof. Trivially, for any functor $\tilde{\mathrm{T}}$ on any category $\tilde{\mathsf{C}}$ with a final object 1 the final arrow $!_{\tilde{\mathrm{T}}1}$ is a unique $\tilde{\mathrm{T}}$-algebra operation on 1, and $\langle 1, !_{\tilde{\mathrm{T}}1} \rangle$ is final:

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

$$\begin{array}{ccc} \tilde{\mathrm{T}}X & \overset{\tilde{\mathrm{T}}\,!_X}{- - - \to} & \tilde{\mathrm{T}}1 \\ {\scriptstyle \beta}\downarrow & \text{finality} & \downarrow{\scriptstyle !_{\tilde{\mathrm{T}}1}} \\ X & \underset{!_X}{- - - \to} & 1 \end{array}$$

The statement follows by considering $\tilde{\mathsf{C}} = \mathsf{Coalg}_{\mathrm{F}}$, $\tilde{\mathrm{T}} = \mathrm{T}_\lambda$, and $1 = \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$. □

In the case of our example ($\mathrm{F} = I\!R \times \mathrm{Id}$ and $\mathrm{T} = \mathrm{Id} \times \mathrm{Id}$), the distributive law λ should give a global account of the addition of two states. For a set X we thus define λ_X as

$$\lambda_X := \langle \langle o_x, s_x \rangle, \langle o_y, s_y \rangle \rangle \mapsto \langle o_x + o_y, \langle s_x, s_y \rangle \rangle. \tag{4}$$

This definition turns the bialgebra $\langle I\!R^\omega, \oplus, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ under consideration into a λ-bialgebra, which is final according to Lemma 3.6.

Our aim is to show the unique existence of homomorphisms up-to into this final λ-bialgebra to yield a definition principle. Therefore, we take $\langle X, \phi \rangle$ as a basis for the construction of a (larger) F-coalgebra $\langle L_X, \alpha_\phi \rangle$. We first show that for any λ-bialgebra $\langle Y, \beta, \alpha \rangle$, homomorphisms up-to from $\langle X, \phi \rangle$ to $\langle Y, \beta, \alpha \rangle$ factor through an F-coalgebra homomorphism from $\langle L_X, \alpha_\phi \rangle$ to $\langle Y, \alpha \rangle$. Second, we prove that for the final λ-bialgebra the converse is also true: by precomposing the coiterative morphism from $\langle L_X, \alpha_\phi \rangle$ to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ with a suitable arrow we get a homomorphism up-to. Our first statement makes an assumption on the category C, allowing the construction of $\langle L_X, \alpha_\phi \rangle$. Later we will also present a different approach, which exploits extra structure coming with T and F instead.

Lemma 3.7. Assume the category C has countable coproducts and let λ be a distributive law of the functor T over the functor F. For an FT-coalgebra $\langle X, \phi \rangle$, consider the F-coalgebra $\langle L_X, \alpha_\phi \rangle$ with $L_X := \coprod_{i=0}^{\infty} \mathrm{T}^i X$ and $\alpha_\phi := [\mathrm{F}\,\mathtt{in}_{i+1} \circ \phi_i]_{i=0}^{\infty}$ for $\phi_0 := \phi$ and $\phi_{i+1} := \lambda_{\mathrm{T}^{i+1}X} \circ \mathrm{T}\phi_i$. A homomorphism up-to f from $\langle X, \phi \rangle$ to a λ-bialgebra $\langle Y, \beta, \alpha \rangle$ factors as $h \circ \mathtt{in}_0$ for an F-coalgebra homomorphism h from $\langle L_X, \alpha_\phi \rangle$ to $\langle Y, \alpha \rangle$.

Proof. With the universal property of the countable coproduct one easily gets that h is a homomorphism from $\langle L_X, \alpha_\phi \rangle$ to $\langle Y, \alpha \rangle$ if and only if

$$\alpha \circ h_i = \mathrm{F}h_{i+1} \circ \phi_i \quad \text{for all } i \in I\!N \tag{5}$$

where $h_i := h \circ \mathtt{in}_i$. We show that $h := [f_i]_{i=0}^{\infty}$ with $f_0 := f$ and $f_{i+1} := f_i|^\beta$ (for which $f = h \circ \mathtt{in}_0$ holds trivially) satisfies (5) by induction on i. For $i = 0$ we find back the assumption on f being a homomorphism up-to. For the induction step, by exploiting (a) the assumption on $\langle Y, \beta, \alpha \rangle$ being a λ-bialgebra, (b) the induction hypothesis, and (c) the naturality of λ, we get

$$\begin{aligned} \alpha \circ f_{i+1} &= \alpha \circ \beta \circ \mathrm{T}f_i \overset{(a)}{=} \mathrm{F}\beta \circ \lambda_Y \circ \mathrm{T}(\alpha \circ f_i) \overset{(b)}{=} \mathrm{F}\beta \circ \lambda_Y \circ \mathrm{T}(\mathrm{F}f_{i+1} \circ \phi_i) \\ &\overset{(c)}{=} \mathrm{F}(\beta \circ \mathrm{T}f_{i+1}) \circ \lambda_{\mathrm{T}^{i+1}X} \circ \mathrm{T}\phi_i = \mathrm{F}f_{i+2} \circ \phi_{i+1}. \end{aligned}$$

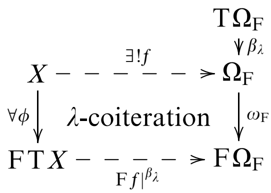
□

In the case of the example, the states of the coalgebra $\langle L_X, \alpha_\phi \rangle$ can be viewed as full binary trees, that is, binary trees where all paths have the same length. Each inner node

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

represents an application of $\oplus$ and each leaf one application of $\otimes$ to two streams. Given such a tree of depth i, the structure α_ϕ first expands each leaf into a first element and another sum for the tail (by applying $\mathrm{T}^{\mathrm{i}}\phi$), and then sums up all the heads level by level (by applying $\mathrm{T}^{\mathrm{i}-1}\lambda_{\mathrm{T}X}$ through $\lambda_{\mathrm{T}^{\mathrm{i}}X}$), resulting in one first element and a tree of depth $i+1$ for the tail. If f is a homomorphism up-to from $\langle X, \phi\rangle$ to the λ-bialgebra $\langle Y, \oplus_Y, \alpha\rangle$, then h in the above lemma evaluates a tree by first mapping f to the leaves and then removing the inner nodes step-by-step by applying $\oplus_Y$.

Theorem 3.8 (λ-coiteration (1)). Assume the category C has countable coproducts. Let λ be a distributive law of the functor T over the functor F and let $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ be a final F-coalgebra. There exists a unique homomorphism up-to f from any FT-coalgebra $\langle X, \phi\rangle$ to $\langle \Omega_{\mathrm{F}}, \beta_\lambda, \omega_{\mathrm{F}}\rangle$ as in Lemma 3.6, which we call the *λ-coiterative arrow coinduced by ϕ.*



Proof. Using Lemma 3.7 and its notation, the only candidate for f is $h_0 = h \circ \mathtt{in}_0$ for the unique coiterative arrow $h : \langle L_X, \alpha_\phi\rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$. We show that it is indeed a homomorphism up-to: from (5) we get $\alpha \circ h_0 = \mathrm{F}h_1 \circ \phi$, from which the statement follows when $h_1 = h_0|^{\beta_\lambda}$. This equation is an immediate consequence of $[h_{i+1}]_{i=0}^{\infty} = [h_i|^{\beta_\lambda}]_{i=0}^{\infty}$ which in turn follows by finality from the fact that both arrows are homomorphisms from $\langle L_{\mathrm{T}X}, \alpha_{\phi_1}\rangle$ (that is, the coalgebra that arises by leaving out the X-component of $\langle L_X, \alpha_\phi\rangle$) to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$. For the left-hand one this is easy to see. The right-hand one can be rewritten as the composition $\beta_\lambda \circ [\mathrm{T}h_i]_{i=0}^{\infty}$ of two homomorphisms $[\mathrm{T}h_i]_{i=0}^{\infty} : \langle L_{\mathrm{T}X}, \alpha_{\phi_1}\rangle \to \mathrm{T}_\lambda\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ and $\beta_\lambda : \mathrm{T}_\lambda\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$. For the first one we show that it satisfies the corresponding instance of condition (5), where we use (a) condition (5) for h and (b) naturality of λ:

$$\lambda_{\Omega_{\mathrm{F}}} \circ \mathrm{T}(\omega_{\mathrm{F}} \circ h_i) \overset{(a)}{=} \lambda_{\Omega_{\mathrm{F}}} \circ \mathrm{T}(\mathrm{F}h_{i+1} \circ \phi_i) \overset{(b)}{=} \mathrm{FT}h_{i+1} \circ \lambda_{\mathrm{T}^{\mathrm{i}+1}X} \circ \mathrm{T}\phi_i = \mathrm{FT}h_{i+1} \circ \phi_{i+1}.$$

And β_λ is a homomorphism by definition. □

Using this theorem, we can conclude that the specification in Section 3.1 did indeed uniquely define the function $\otimes$, because the example was living in Set, which has countable coproducts. In Section 4 we will further show that this $\otimes$ satisfies (2) as intended.

We conclude this section by showing that the proof principle from Theorem 2.8 can easily be adapted to λ-coiterative arrows.

Corollary 3.9. In Set let F, T, λ, and $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ be as in Theorem 3.8. For two FT-coalgebras $\langle X, \phi_X\rangle$ and $\langle Y, \phi_Y\rangle$ with λ-coiterative morphisms f_X and f_Y, respectively, we have

$$x \sim y \quad \Rightarrow \quad f_X(x) = f_Y(y),$$

where $\sim$ now denotes FT-bisimilarity.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

Proof. In a similar way to the proof of Theorem 2.8, let $R \subseteq X \times Y$ be an FT-bisimulation with $\langle x, y\rangle \in R$ and let ϕ_R be an FT-coalgebra operation witnessing the bisimulation property of R. The statement follows from the equation $f_X \circ \pi_1 = f_Y \circ \pi_2$, which holds by the uniqueness part of Theorem 3.8 since both composites are homomorphisms up-to from $\langle R, \phi_R\rangle$ to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ by Lemma 3.3 (ii). □

We should not conceal an important difference in the significance of the two statements: in many important cases the converse direction of Theorem 2.8 is also true (for example, when the functor F weakly preserves pullbacks), so that it yields a *complete* proof principle. Unfortunately, this is not true for Corollary 3.9.

4. Proof by λ-coinduction

When developing a bisimilarity proof, there is often more to do to arrive at a bisimulation than just relate the pairs of states that one wants to prove bisimilar. Many more pairs of successor states are usually needed, and the hard part is often to come up with a simple description of a relation sufficiently large to cover them all. This work can sometimes be eased by considering relations that satisfy conditions weaker than being a bisimulation but strong enough for a general argument to show that the relation is expandable to some bisimulation. Such relations are often called *bisimulations up-to*. We will now show how one can get such conditions out of the framework presented in the previous section.

4.1. *Example: functional product and stream product*

In this section we would like to show that the multiplication of streams of real numbers defined in Section 3.1 does indeed satisfy equation (2). With the coinduction proof principle (Theorem 2.8) it suffices to prove that the streams on both sides of the equation are bisimilar. Ideally, this should be possible by considering the relation

$$R := \{\langle \mathcal{T}(f \cdot g), \mathcal{T}(f) \otimes \mathcal{T}(g)\rangle \mid f, g \in \mathscr{A}\} \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega. \tag{6}$$

But, unfortunately, the attempt to prove that it is a bisimulation fails: it is easy to show that all streams related have equal heads, but for the tails one obtains

$$\begin{array}{rcccccc} (\sigma' :=) & \mathtt{tail}(\underbrace{\mathcal{T}(f \cdot g)}_{=:\sigma}) & = & \underbrace{\mathcal{T}(f \cdot g')}_{=:\sigma'_l} & \oplus & \underbrace{\mathcal{T}(f' \cdot g)}_{=:\sigma'_r} & \\ (\tau' :=) & \mathtt{tail}(\underbrace{\mathcal{T}(f) \otimes \mathcal{T}(g)}_{=:\tau}) & = & (\underbrace{\mathcal{T}(f) \otimes \mathcal{T}(g')}_{=:\tau'_l}) & \oplus & (\underbrace{\mathcal{T}(f') \otimes \mathcal{T}(g)}_{=:\tau'_r}). \end{array} \tag{7}$$

This shows that for $\langle \sigma, \tau\rangle \in R$ instead of containing $\langle \sigma', \tau'\rangle$, as required for a bisimulation, R relates two pairs $\langle \sigma'_l, \tau'_l\rangle$ and $\langle \sigma'_r, \tau'_r\rangle$ with $\sigma' = \sigma'_l \oplus \sigma'_r$ and $\tau' = \tau'_l \oplus \tau'_r$.

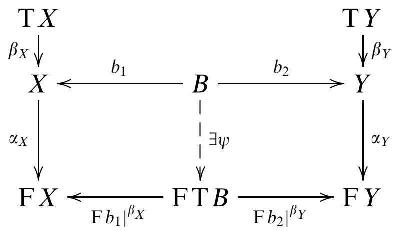
In the following we will show that the above condition is sufficient to conclude that B is contained in some larger bisimulation, which is what we need in order to prove our initial goal.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

4.2. λ-coinduction

In Section 3 we demonstrated how an FT-coalgebra $\langle X, \phi \rangle$ can be taken to construct a (larger) F-coalgebra, and how homomorphisms out of this coalgebra can be constructed from homomorphism up-to from $\langle X, \phi \rangle$. Here we will exploit the same idea for bisimilarity proofs: we will take an FT-coalgebra structure on a relation R to construct a larger relation containing R. The new relation will be a bisimulation, when R is a λ-*bisimulation*, a notion we introduce here.

Definition 4.1 (λ-bisimulation). Let λ be a distributive law of a functor T over a functor F. A span $\mathscr{B} = \langle B, b_1, b_2 \rangle$ is a λ-*bisimulation* between the λ-bialgebras $\langle X, \beta_X, \alpha_X \rangle$ and $\langle Y, \beta_Y, \alpha_Y \rangle$, if there exists an FT-operation ψ on B, such that b_1 and b_2 are homomorphisms up-to from $\langle B, \psi \rangle$ to $\langle X, \beta_X, \alpha_X \rangle$ and $\langle Y, \beta_Y, \alpha_Y \rangle$, respectively:



A λ-bisimulation between $\langle X, \beta, \alpha \rangle$ and itself will be called a λ-*bisimulation on* $\langle X, \beta, \alpha \rangle$.

A λ-bisimulation can be extended to a standard bisimulation using the construction from Lemma 3.7.

Theorem 4.2. Let the category C have countable coproducts, let λ be a distributive law of the functor T over the functor F, and let $\langle X, \beta_X, \alpha_X \rangle$ and $\langle Y, \beta_Y, \alpha_Y \rangle$ be λ-bialgebras. If $\mathscr{B} = \langle B, b_1, b_2 \rangle$ is a λ-bisimulation between $\langle X, \beta_X, \alpha_X \rangle$ and $\langle Y, \beta_Y, \alpha_Y \rangle$, then there exists a (conventional) bisimulation $\tilde{\mathscr{B}}$ between the F-coalgebras involved with $\mathscr{B} \preceq \tilde{\mathscr{B}}$.

Proof. Let $\psi : B \to \mathrm{FT}B$ be a witness for $\mathscr{B}$ being a λ-bisimulation (Definition 4.1). Lemma 3.7 says that there exist F-coalgebra homomorphisms $h_1 : \langle L_B, \alpha_\psi \rangle \to \langle X, \alpha_X \rangle$ and $h_2 : \langle L_B, \alpha_\psi \rangle \to \langle Y, \alpha_Y \rangle$ such that $b_i = h_i \circ \mathtt{in}_0$ for $i = 1, 2$. This makes α_ψ a witness that $\tilde{\mathscr{B}} := \langle L_B, h_1, h_2 \rangle$ is a bisimulation between $\langle X, \alpha_X \rangle$ and $\langle Y, \alpha_Y \rangle$, and $\mathtt{in}_0$ shows $\mathscr{B} \preceq \tilde{\mathscr{B}}$. □

Since Set has countable coproducts, Theorem 2.8 can be modified using Theorem 4.2.

Corollary 4.3 (λ-coinduction proof principle). Given a distributive law λ of a functor T over a functor F in Set, a final F-coalgebra $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$, and a λ-bisimulation relation R on $\langle \Omega_\mathrm{F}, \beta_\lambda, \omega_\mathrm{F} \rangle$ as in Lemma 3.6, we have that $\langle p, q \rangle \in R$ implies $p = q$.

The relation R from the example can be seen to be a λ-bisimulation on the final λ-bialgebra $\langle \mathbb{R}^\omega, \oplus, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ for λ as in (4). The operation $\psi : R \to \mathrm{FT}R$ in Definition 4.1 can be chosen as follows: for analytic functions $f, g \in \mathscr{A}$ it maps the pair $\langle \sigma, \tau \rangle$ to $\langle f(0) \cdot g(0), \langle \langle \sigma'_l, \tau'_l \rangle, \langle \sigma'_r, \tau'_r \rangle \rangle \rangle$, where we have again used the abbreviations from the equations (7). Corollary 4.3 now proves that $\otimes$ satisfies equation (2), as required.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

5. λ-coiteration and operators

In this section we are again looking for a schema for coinductive specifications involving auxiliary operators. But this time they should be more freely usable in the sense that the successor states can be described by multiple applications of them. Moreover, it should allow operators that would not have been available inside the schema we have considered so far. Our investigations will lead to a statement based on a variant of the λ-coiteration theorem living in the world of monads and copointed functors. Again we start with an example.

5.1. *The stream of Hamming Numbers*

Taking up an example from Dijkstra's (Dijkstra 1981), we consider the stream $\mathtt{ham} \in \mathbb{N}^\omega$ containing all natural numbers in increasing order with no prime factors other than 2 and 3. These numbers are often referred to as the *Hamming Numbers* (admittedly we have omitted the prime factor 5 for simplicity). As with the previous examples, the infinite streams of natural numbers $\mathbb{N}^\omega$ are equipped with the operation $\langle \mathtt{head}, \mathtt{tail} \rangle$ turning them into a final coalgebra, namely for the functor $\mathrm{F} := \mathbb{N} \times \mathrm{Id}$. Consider the specification

$$\mathtt{head}(\mathtt{ham}) = 1 \quad \text{and} \quad \mathtt{tail}(\mathtt{ham}) = \mathtt{merge}(\mathtt{map}_{\times 2}(\mathtt{ham}), \mathtt{map}_{\times 3}(\mathtt{ham})) \tag{8}$$

where $\mathtt{merge} : \mathbb{N}^\omega \times \mathbb{N}^\omega \to \mathbb{N}^\omega$ and $\mathtt{map}_g : \mathbb{N}^\omega \to \mathbb{N}^\omega$ (for $g : \mathbb{N} \to \mathbb{N}$) are the operators given coiteratively by declaring that for all $\sigma = \langle s_0 : \sigma' \rangle$ and $\tau = \langle t_0 : \tau' \rangle$,

$$\begin{aligned} \langle \mathtt{head}, \mathtt{tail} \rangle(\mathtt{merge}(\sigma, \tau)) &= \begin{cases} \langle s_0, \mathtt{merge}(\sigma', \tau) \rangle & \text{if } s_0 < t_0 \\ \langle s_0, \mathtt{merge}(\sigma', \tau') \rangle & \text{if } s_0 = t_0 \\ \langle t_0, \mathtt{merge}(\sigma, \tau') \rangle & \text{if } s_0 > t_0 \end{cases} \\ \langle \mathtt{head}, \mathtt{tail} \rangle(\mathtt{map}_g(\sigma)) &= \langle g(s_0), \mathtt{map}_g(\sigma') \rangle, \end{aligned}$$

and $\times 2, \times 3 : \mathbb{N} \to \mathbb{N}$ are the functions that double and triple their arguments.

To view this as a specification of a function, we treat the constant stream as an arrow $\mathtt{ham} : 1 \to \mathbb{N}^\omega$, where $1 = \{*\}$ is a singleton set. To be precise, the stream of Hamming Numbers would then arise as $\mathtt{ham}(*) \in \mathbb{N}^\omega$, but we still denote it by $\mathtt{ham}$ alone for simplicity. Again, the question is whether there exists a unique function satisfying (8). We are going to give an answer to this type of question in the remainder of this section.

5.2. *The problems posed by the example*

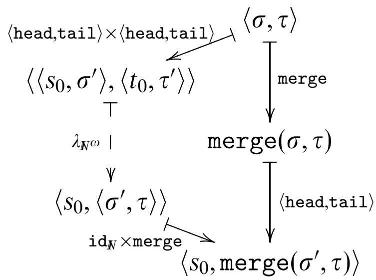
The example cannot be handled by the framework we have developed so far for two reasons.

First, the specification for $\mathtt{tail}(\mathtt{ham})$ involves the application of three different operators at the same time. If we work again with a functor T capturing the typing of all the auxiliary operators under consideration, this would not yield an arrow $\phi : 1 \to \mathrm{FT}1$. As a solution, we will take T to represent all terms that we can build. But without any extra precautions, we would not know whether the T-algebra structure β_λ appearing in the schema would evaluate these terms in the way we expect, namely by iteratively applying

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

suitable operators. The algebras doing this are the algebras of the *term monad*, and we can guarantee a reasonable outcome by working entirely with these.

Second, it turns out that the operation merge cannot be handled by the present framework, because for $\sigma = \langle s_0 : \sigma' \rangle, \tau = \langle t_0 : \tau' \rangle \in I\!N^\omega$ the tail of $\mathtt{merge}(\sigma, \tau)$ is not expressed in terms of s_0, t_0, σ', and τ' only, but also with reference to σ and τ themselves. To illustrate why this is a problem, assume we wanted to use merge instead of $\oplus$ within our first example from Section 3.1. This would require a distributive law λ such that $\langle I\!N^\omega, \mathtt{merge}, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ is a λ-bialgebra. Consider the instance of the pentagonal diagram from Definition 3.4 for σ and τ as above with $s_0 < t_0$:



It turns out that $\lambda_{I\!N^\omega}$ needs to produce a result involving τ, even though it is not among its arguments. One may be tempted to suggest that this could be solved by reconstructing τ from t_0 and τ', but λ should be a natural transformation and how would we generalise this approach to λ_X for other sets X?

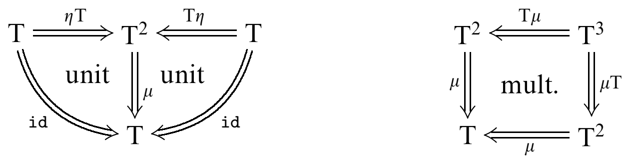
This limitation of the present framework is due to the fact that given only $\alpha(x)$ for some arbitrary (unknown) F-coalgebra $\langle X, \alpha \rangle$, there is no way to arrive back at $x \in X$. To overcome this problem, we turn towards a special class of coalgebras for which the application of the coalgebra structure α can be inverted (even without knowing α), namely, to coalgebras for a *copointed functor*. This will later allow us to handle operators like merge by transforming a given F-coalgebra into a coalgebra of the *cofree copointed functor* generated by F, which exists when C has binary products.

We are going to take the following approach. Starting with a signature Σ and a behaviour functor F, we will apply an adapted version of the λ-coiteration theorem to the term monad generated by Σ and the cofree copointed functor generated by F. Then we are going to reformulate the resulting statement in terms of the original ingredients Σ and F, which will yield a generalisation of the corresponding instance of the λ-coiteration schema.

5.3. *λ-coiteration for monads and copointed functors*

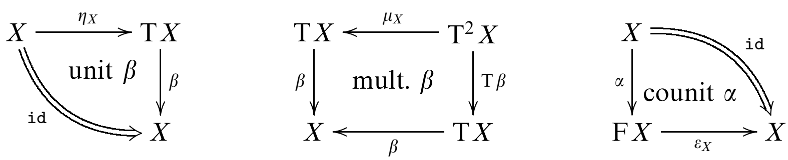
In this section we are going to restate the λ-coiteration framework for algebras for a monad and coalgebras for a copointed functor. We recall the definitions first.

Definition 5.1 (Monad, copointed functor). A *monad* is a triple $\langle \mathrm{T}, \eta, \mu \rangle$ of a functor $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ and two natural transformations $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$ and $\mu : \mathrm{T}^2 \Rightarrow \mathrm{T}$, called the *unit* and *multiplication* of the monad, such that the three parts of the diagrams below commute – we will call them the *unit* and *multiplication laws* of the monad.

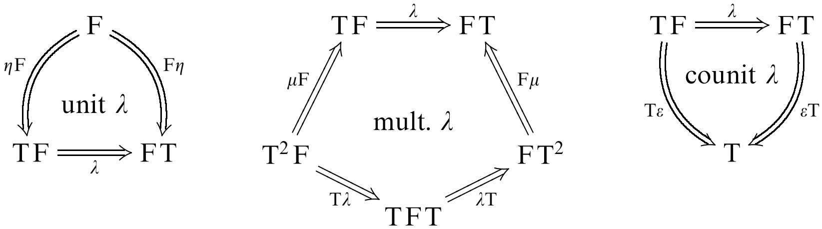


A *copointed functor* on a category C is a pair $\langle F, \varepsilon \rangle$ of a functor $F : C \to C$ and a natural transformation $\varepsilon : F \Rightarrow \mathrm{Id}$, called its *counit*.

A T-algebra $\langle X, \beta \rangle$ is called an *algebra for the monad* $\langle T, \eta, \mu \rangle$ if the left and middle diagram below commute – we will refer to them as the *unit* and *multiplication law* for β. The full subcategory of Alg^{T} containing all such algebras is denoted by $\mathsf{Alg}^{\langle T, \eta, \mu \rangle}$. A *coalgebra for a copointed functor* $\langle F, \varepsilon \rangle$ is an F-coalgebra $\langle X, \alpha \rangle$ such that α satisfies the *counit law* in right diagram below. The full subcategory of Coalg_{F} containing all such coalgebras is denoted by $\mathsf{Coalg}_{\langle F, \varepsilon \rangle}$.



A natural transformation $\lambda : TF \Rightarrow FT$ is called a *distributive law of the monad* $\langle T, \eta, \mu \rangle$ *over the copointed functor* $\langle F, \varepsilon \rangle$ if it satisfies the *unit*, *multiplication* and *counit laws for* λ depicted in the diagrams below. In this setting, a $\langle T, F \rangle$-bialgebra $\langle X, \beta, \alpha \rangle$ is called a *λ-bialgebra* if it makes the diagram in Definition 3.4 commute, $\langle X, \beta \rangle$ is an algebra for the monad, and $\langle X, \alpha \rangle$ is a coalgebra for the copointed functor.



Lemma 5.2 (λ-lifting (2)). Given a distributive law λ of a monad $\langle T, \eta, \mu \rangle$ over a copointed functor $\langle F, \varepsilon \rangle$, we can lift the monad $\langle T, \eta, \mu \rangle$ on C to a monad $\langle T_{\lambda}, \eta, \mu \rangle$ on $\mathsf{Coalg}_{\langle F, \varepsilon \rangle}$ by setting

$$T_{\lambda}\langle X, \alpha \rangle := \langle TX, \lambda_X \circ T\alpha \rangle, \quad T_{\lambda} h := Th, \quad \eta_{\langle X, \alpha \rangle} := \eta_X, \quad \text{and} \quad \mu_{\langle X, \alpha \rangle} := \mu_X,$$

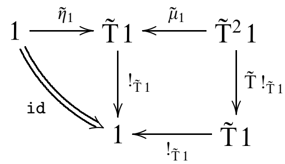
for any coalgebra for the copointed functor $\langle X, \alpha \rangle$ and homomorphism h.

Proof. For T_{λ}, in addition to the proof of Lemma 3.5, we need to show that $\lambda_X \circ T\alpha$ satisfies the counit law. This follows easily from the counit laws for α and λ. For η and μ one easily gets from the unit and multiplication law of λ that η_X and μ_X are homomorphisms from $\langle X, \alpha \rangle$ and ${T_{\lambda}}^2 \langle X, \alpha \rangle$, respectively, to $T_{\lambda}\langle X, \alpha \rangle$. □

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

Lemma 5.3. Let λ be a distributive law of the monad $\langle \mathrm{T}, \eta, \mu \rangle$ over the copointed functor $\langle \mathrm{F}, \varepsilon \rangle$. For a final coalgebra $\langle \Omega_{\langle \mathrm{F},\varepsilon \rangle}, \omega_{\langle \mathrm{F},\varepsilon \rangle} \rangle$ for the copointed functor there exists a unique T-algebra operation β_λ on $\Omega_{\langle \mathrm{F},\varepsilon \rangle}$ such that $\langle \Omega_{\langle \mathrm{F},\varepsilon \rangle}, \beta_\lambda, \omega_{\langle \mathrm{F},\varepsilon \rangle} \rangle$ is a λ-bialgebra. Furthermore, this λ-bialgebra is final.

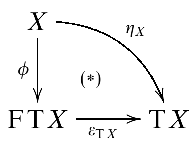
Proof. The algebra operation β_λ is set as in the proof of Lemma 3.6 (with $\mathsf{Coalg}_{\mathrm{F}}$ replaced by $\mathsf{Coalg}_{\langle \mathrm{F},\varepsilon \rangle}$). In addition, we need to show that this yields an algebra for the monad. Let $\langle \tilde{\mathrm{T}}, \tilde{\eta}, \tilde{\mu} \rangle$ be a monad in a category $\tilde{\mathsf{C}}$ with a final object 1. The unique $\tilde{\mathrm{T}}$-algebra operation $!_{\tilde{\mathrm{T}}1}$ on 1 is an algebra for the monad, since both parts of the following diagram commute by finality:



This argument yields the statement when instantiated with the monad $\langle \mathrm{T}_\lambda, \eta, \mu \rangle$ from Lemma 5.2 (modulo the application of the forgetful functor $\mathrm{U} : \mathsf{Coalg}_{\langle \mathrm{F},\varepsilon \rangle} \to \mathsf{C}$). $\square$

When the functor T comes as a monad $\langle \mathrm{T}, \eta, \mu \rangle$ and the distributive law λ interacts with η and μ as in the above definition, the construction of the F-coalgebra from an FT-coalgebra in Lemma 3.7 can be simplified as follows. For every $i \in \mathbb{N}$ the elements from $\mathrm{T}^i X$ can be represented within $\mathrm{T}X$, by applying η or (possibly several times) μ. The additional assumptions on λ ensure that these mappings 'preserve behaviours', so it suffices to take this second component of the countable coproduct considered previously as the carrier of the F-coalgebra that is to be constructed. This allows us to drop the assumption on C. Moreover, the counit law for λ together with an additional assumption on ϕ guarantees that the resulting coalgebra is a coalgebra for the copointed functor.

Lemma 5.4. Let λ be a distributive law of the monad $\langle \mathrm{T}, \eta, \mu \rangle$ over the copointed functor $\langle \mathrm{F}, \varepsilon \rangle$. Every FT-coalgebra $\langle X, \phi \rangle$ making the the diagram (*) below commute gives rise to a λ-bialgebra $\langle L_X, \mu_X, \alpha_\phi \rangle$ with $L_X := \mathrm{T}X$ and $\alpha_\phi := \mathrm{F}\mu_X \circ \phi_1$, where again $\phi_1 := \lambda_{\mathrm{T}X} \circ \mathrm{T}\phi$.

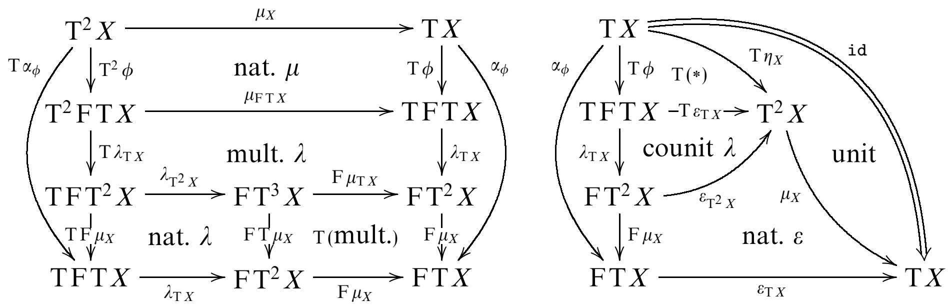


Proof. We need to check that

(i) $\langle L_X, \mu_X, \alpha_\phi \rangle$ satisfies the pentagonal law from Definition 3.4,
(ii) μ_X is an algebra for the monad, and
(iii) α_ϕ is a coalgebra for the copointed functor.

Item (ii) is obvious with the monad laws and for (i) and (iii) we do the left and right diagram chase below.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press



□

Lemma 5.5. Let λ, $\langle \mathrm{T}, \eta, \mu\rangle$, $\langle \mathrm{F}, \varepsilon\rangle$, and $\langle X, \phi\rangle$ be given as in Lemma 5.4. A homomorphism up-to f from $\langle X, \phi\rangle$ to any λ-bialgebra $\langle Y, \beta, \alpha\rangle$ factors as $h \circ \eta_X$ for a bialgebra homomorphism h from $\langle L_X, \mu_X, \alpha_\phi\rangle$ to $\langle Y, \beta, \alpha\rangle$.

Proof. By setting $h := f|^\beta$ with (a) the naturality of η and (b) the unit law of β, we indeed get

$$h \circ \eta_X = \beta \circ \mathrm{T}f \circ \eta_X \overset{(a)}{=} \beta \circ \eta_Y \circ f \overset{(b)}{=} f.$$

Furthermore, (a) the naturality of μ and (b) the multiplication law for β yields that h is a T-algebra homomorphism from $\langle L_X, \mu_X\rangle$ to $\langle Y, \beta\rangle$:

$$h \circ \mu_X = \beta \circ \mathrm{T}f \circ \mu_X \overset{(a)}{=} \beta \circ \mu_Y \circ \mathrm{T}^2 f \overset{(b)}{=} \beta \circ \mathrm{T}(\beta \circ \mathrm{T}f) = \beta \circ \mathrm{T}h. \tag{9}$$

From this and (a) the fact that h_f is a homomorphism up-to from $\langle L_X, \phi_1\rangle$ to $\langle Y, \beta, \alpha\rangle$, it finally follows that h is also a coalgebra homomorphism from $\langle L_X, \alpha_\phi\rangle$ to $\langle Y, \alpha\rangle$:

$$\alpha \circ h \overset{(a)}{=} \mathrm{F}h|^\beta \circ \phi_1 = \mathrm{F}(\beta \circ \mathrm{T}h) \circ \phi_1 \overset{(9)}{=} \mathrm{F}(h \circ \mu_X) \circ \phi_1 = \mathrm{F}h \circ \alpha_\phi.$$

To show (a), we instantiate Lemma 3.3 (i) with $\mathrm{T}f$, β and the intermediate bialgebra $\langle \mathrm{T}Y, \mathrm{T}\beta, \lambda_Y \circ \mathrm{T}\alpha\rangle$. From the assumption that $\langle Y, \beta, \alpha\rangle$ is a λ-bialgebra, it follows that β fits as a bialgebra homomorphism, and $\mathrm{T}f$ fits as a homomorphism up-to with (a) the assumption on f and (b) the naturality of λ:

$$\lambda_Y \circ \mathrm{T}(\alpha \circ f) \overset{(a)}{=} \lambda_Y \circ \mathrm{T}(\mathrm{F}f|^\beta \circ \phi) \overset{(b)}{=} \mathrm{FT}f|^\beta \circ \lambda_{\mathrm{T}X} \circ \mathrm{T}\phi = \mathrm{F}(\mathrm{T}f)|^{(\mathrm{T}\beta)} \circ \phi_1.$$ □

Theorem 5.6 (λ-coiteration (2)). Let λ be a distributive law of the monad $\langle \mathrm{T}, \eta, \mu\rangle$ over the copointed functor $\langle \mathrm{F}, \varepsilon\rangle$, and let $\langle \Omega_{\langle \mathrm{F},\varepsilon\rangle}, \omega_{\langle \mathrm{F},\varepsilon\rangle}\rangle$ be a final coalgebra for $\langle \mathrm{F}, \varepsilon\rangle$. For every FT-coalgebra $\langle X, \phi\rangle$ making diagram $(*)$ in Lemma 5.4 commute, there exists a unique homomorphism up-to f from $\langle X, \phi\rangle$ to $\langle \Omega_{\langle \mathrm{F},\varepsilon\rangle}, \beta_\lambda, \omega_{\langle \mathrm{F},\varepsilon\rangle}\rangle$ for β_λ as in Lemma 5.3, which we again call the λ-coiterative arrow coinduced by ϕ.

Proof. Instantiating Lemma 5.5 with $\langle \Omega_{\langle \mathrm{F},\varepsilon\rangle}, \beta_\lambda, \omega_{\Omega_{\langle \mathrm{F},\varepsilon\rangle}}\rangle$ shows that the only candidate for f is $h \circ \eta_X$, where h is the unique bialgebra homomorphism from $\langle L_X, \mu_X, \alpha_\phi\rangle$ to this final λ-bialgebra. That it is indeed a homomorphism up-to follows from Lemma 3.3 (i) because from (a) the naturality of η, (b) the unit law of λ, and (c) the unit laws of the

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

monad, one gets that η_X is a homomorphism up-to from $\langle X, \phi \rangle$ to $\langle L_X, \mu_X, \alpha_\phi \rangle$:

$$\begin{aligned} \alpha_\phi \circ \eta_X &= \mathrm{F}\mu_X \circ \lambda_{\mathrm{T}X} \circ \mathrm{T}\phi \circ \eta_X \stackrel{(a)}{=} \mathrm{F}\mu_X \circ \lambda_{\mathrm{T}X} \circ \eta_{\mathrm{FT}X} \circ \phi \\ &\stackrel{(b)}{=} \mathrm{F}(\mu_X \circ \eta_{\mathrm{T}X}) \circ \phi \stackrel{(c)}{=} \mathrm{F}\mathtt{id} \circ \phi \stackrel{(c)}{=} \mathrm{F}(\mu_X \circ \mathrm{T}\eta_X) \circ \phi \\ &= \mathrm{F}\eta_X|^{\mu_X} \circ \phi. \end{aligned}$$

□

Note that there is actually a range of λ-coiteration theorems involving more or less extra structure for T and F, of which Theorem 3.8 (plain functors) and Theorem 5.6 (monad and copointed functor) are actually two extreme representatives. One intermediate version uses distributive laws of a monad $\langle \mathrm{T}, \eta, \mu \rangle$ over a functor F. The proof is based on the same construction as that of Theorem 5.6, but requires fewer side conditions to be checked. This version is most appropriate for showing that a number of known schemata are covered by λ-coiteration, but for space limitations we have not presented it separately here.

As in Section 4, we can also develop a corresponding notion of a λ-bisimulation and a variant of the λ-coinduction proof principle (Corollary 4.3) in this setting involving a monad and a copointed functor.

5.4. *λ-coiteration for operators*

Let $\Sigma = (\Sigma_n)_{n \in \mathbb{N}}$ be a signature, which is to say a set of operator symbols, each with an associated arity ($\sigma \in \Sigma_n$ is an operator symbol with arity n). As usual, this gives rise to a *signature functor*, which we will again call Σ, namely,

$$\Sigma X := \coprod_{n \in \mathbb{N}} \Sigma_n \times X^n = \{\sigma(x_1, \dots, x_n) \mid n \in \mathbb{N},\ \sigma \in \Sigma_n,\ x_1, \dots, x_n \in X\},$$

where for readability we write the tuple $\langle \sigma, x_1, \dots, x_n \rangle$ like a function application.

The signature functor Σ freely generates a monad $\langle \mathrm{T}, \eta, \mu \rangle$, which we will call the *term monad*. The functor T maps a set X to the set of free Σ-terms over X, that is, $\mathrm{T}X$ is the carrier of the initial $(X + \Sigma)$-algebra (it can alternatively be characterised as the smallest set such that $X \subseteq \mathrm{T}X$ and $\Sigma\mathrm{T}X \subseteq \mathrm{T}X$). Calling X a set of variables in this context, the arrow part of T amounts to the renaming of variables. The unit $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$ yields the embedding of variables into terms (we will usually leave its application implicit) and $\mu : \mathrm{T}^2 \Rightarrow \mathrm{T}$ flattens terms having again terms as variables.

By induction on the term structure, any Σ-algebra operation $\Gamma : \Sigma X \to X$ can be extended to a T-algebra operation $[\![.]\!]_\Gamma : \mathrm{T}X \to X$. The T-algebra operations obtained in this way are precisely the *algebras for the term monad* (*c.f.* Definition 5.1). This gives us that Alg^Σ and $\mathsf{Alg}^{\langle \mathrm{T}, \eta, \mu \rangle}$ are isomorphic.

For the coalgebra part, note that in a category with binary products every functor F gives rise to the copointed functor $\langle \mathrm{Id} \times \mathrm{F}, \pi_1 \rangle$, which can be regarded as the *cofree copointed functor* generated by F (Lenisa *et al.* 2000). One aspect of the special relation between these two structures is that $\mathsf{Coalg}_\mathrm{F}$ is isomorphic to $\mathsf{Coalg}_{\langle \mathrm{Id} \times \mathrm{F}, \pi_1 \rangle}$: each coalgebra for the copointed functor can be written as $\langle X, \langle \mathtt{id}_X, \alpha \rangle \rangle$ for an F-coalgebra operation α. In particular, we have that a final F-coalgebra $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ yields a final coalgebra $\langle \Omega_\mathrm{F}, \langle \mathtt{id}_{\Omega_\mathrm{F}}, \omega_\mathrm{F} \rangle \rangle$ for $\langle \mathrm{Id} \times \mathrm{F}, \pi_1 \rangle$

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

With these two correspondences we can derive the following statement from Theorem 5.6.

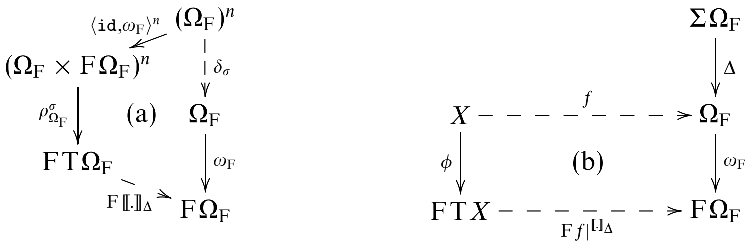
Corollary 5.7. Let F be a functor with a final coalgebra $\langle \Omega_F, \omega_F \rangle$ and let Σ be a signature (also regarded as a functor) generating the term monad $\langle T, \eta, \mu \rangle$. If for all $n \in I\!N$ each $\sigma \in \Sigma_n$ comes with a natural transformation

$$\rho^{\sigma} : (\mathrm{Id} \times F)^n \Rightarrow FT, \tag{10}$$

we can uniquely associate to each such σ an operation $\delta_\sigma : (\Omega_F)^n \to \Omega_F$ such that diagram (a) below commutes, where $\Delta : \Sigma\Omega_F \to \Omega_F$ is given by

$$\Delta(\sigma(p_1, \dots, p_n)) := \delta_\sigma(p_1, \dots, p_n).$$

Furthermore, for every FT-coalgebra $\langle X, \phi \rangle$ there is a unique arrow $f : X \to \Omega_F$ fitting into diagram (b).



Proof sketch. The ρ^σ can be combined with the natural transformation $\rho : \Sigma(\mathrm{Id} \times F) \Rightarrow FT$, which can be inductively extended to a distributive law $\lambda^\rho : T(\mathrm{Id} \times F) \Rightarrow (\mathrm{Id} \times F)T$ of the term monad $\langle T, \eta, \mu \rangle$ over the copointed functor $\langle \mathrm{Id} \times F, \pi_1 \rangle$ (*cf.* Theorem 5.4 in Lenisa *et al.* (2000)). Lemma 5.3 now yields a unique algebra for the monad β_λ such that $\langle \Omega_F, \beta_\lambda, \langle \mathtt{id}, \omega_F \rangle \rangle$ is a final λ^ρ-bialgebra. The isomorphism $\mathsf{Alg}^{\Sigma} \cong \mathsf{Alg}^{\langle T, \eta, \mu \rangle}$ gives us a unique Σ-algebra operation Δ on Ω_F such that $\beta_\lambda = [\![.]\!]_\Delta$, and this Δ is equivalent to a set of operators δ_σ for $n \in I\!N$ and $\sigma \in \Sigma_n$ as mentioned in the statement. The diagram in Definition 3.4 for the bialgebra now decomposes into the collection of diagrams (a) for each δ_σ.

The second part follows by applying Theorem 5.6 to the $(\mathrm{Id} \times F)T$-coalgebra $\langle \eta_X, \phi \rangle : X \to TX \times FTX$, which trivially satisfies condition $(*)$ from Lemma 5.4. The characterisation of the λ-coiterative arrow obtained from this theorem can be simplified (using naturality of η and the unit law for $[\![.]\!]_\Delta$) to diagram (b). □

We will now use this statement to conclude the Hamming Number example. The signature $\Sigma = (\Sigma_n)_{n \in N}$ will be set as $\Sigma_1 := \{\underline{\mathtt{map}}_g \mid g : I\!N \to I\!N\}$, $\Sigma_2 := \{\underline{\mathtt{merge}}\}$, and $\Sigma_i := \emptyset$ for $i \in I\!N \setminus \{1, 2\}$. For the ρ^σ we take

$$\rho_X^{\mathtt{merge}}(\langle x, \langle x_0, x' \rangle \rangle, \langle y, \langle y_0, y' \rangle \rangle) := \begin{cases} \langle x_0, \underline{\mathtt{merge}}(x', y) \rangle & \text{if } x_0 < y_0 \\ \langle x_0, \underline{\mathtt{merge}}(x', y') \rangle & \text{if } x_0 = y_0 \\ \langle y_0, \underline{\mathtt{merge}}(x, y') \rangle & \text{if } x_0 > y_0 \end{cases}$$

$$\rho_X^{\mathtt{map}_g}(\langle x, \langle x_0, x' \rangle \rangle) := \langle g(x_0), \underline{\mathtt{map}}_g(x') \rangle.$$

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

It is easy to check that the definitions of merge and $\mathtt{map}_g$ are equivalent to the corresponding instances of diagram (a) in the above corollary, so we find $\delta_{\underline{\mathtt{merge}}} = \mathtt{merge}$ and $\delta_{\underline{\mathtt{map}}_g} = \mathtt{map}_g$. The specification (8) is translated into the function $\phi : 1 \to I\!N \times \mathrm{T}1$ defined by

$$\phi(*) := \langle 1, \underline{\mathtt{merge}}(\underline{\mathtt{map}}_{\times 2}(*), \underline{\mathtt{map}}_{\times 3}(*))\rangle. \tag{11}$$

Instantiating Corollary 5.7 with this yields a unique arrow $\mathtt{ham} : 1 \to I\!N^{\omega}$ satisfying diagram (b). With

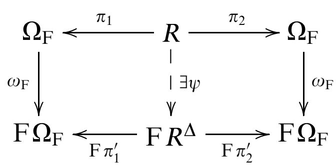
$$\begin{aligned}(\mathrm{F}\mathtt{ham}|^{[\![.]\!]_\Delta} \circ \phi)(*) &= (\mathrm{F}([\![.]\!]_\Delta \circ \mathrm{T}\mathtt{ham}))(\langle 1, \underline{\mathtt{merge}}(\underline{\mathtt{map}}_{\times 2}(*), \underline{\mathtt{map}}_{\times 3}(*))\rangle) \\ &= \langle 1, [\![\underline{\mathtt{merge}}(\underline{\mathtt{map}}_{\times 2}(\mathtt{ham}), \underline{\mathtt{map}}_{\times 3}(\mathtt{ham}))]\!]_\Delta\rangle \\ &= \langle 1, \mathtt{merge}(\mathtt{map}_{\times 2}(\mathtt{ham}), \mathtt{map}_{\times 3}(\mathtt{ham}))\rangle\end{aligned}$$

we have that this is equivalent to ham satisfying Equation (8). So the latter has a unique solution, as required.

5.5. *Bisimulation up-to-context*

Along the same lines we can also adapt the λ-coinduction proof principle based on Theorem 4.2. For simplicity we will concentrate on λ-bisimulation relations on the final λ-bialgebra. The basic observation here is that for a relation $R \subseteq \Omega_\mathrm{F} \times \Omega_\mathrm{F}$, a span of the shape $\langle \mathrm{T}R, \pi_1|^{[\![.]\!]_\Delta}, \pi_2|^{[\![.]\!]_\Delta}\rangle$, as it appears inside the corresponding definition of a λ-bisimulation, describes the *congruence closure* R^Δ of R under Δ. By this we mean the smallest relation containing R such that for all components δ of Δ with arity n we have that $\langle \delta(p_1, \ldots, p_n), \delta(q_1, \ldots, q_n)\rangle \in R^\Delta$ if $\langle p_i, q_i\rangle \in R^\Delta$ for $1 \leqslant i \leqslant n$.

Corollary 5.8. Let F, Σ and ρ^σ be given as in Corollary 5.7, and take Δ to be the set of operators provided by the first part of its statement. We call a relation $\langle R, \pi_1, \pi_2\rangle$ on the carrier of the final F-coalgebra $\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ a *bisimulation up-to-context* if there exists a function $\psi : R \to \mathrm{F}R^\Delta$ (where $\langle R^\Delta, \pi_1', \pi_2'\rangle$ is the congruence closure of R under the operators δ_σ in Δ) making both parts of the following diagram commute:



For such a relation R, $\langle p, q\rangle \in R$ implies $p = q$ for all $p, q \in \Omega_\mathrm{F}$.

The successors of two states related by a bisimulation up-to-context R need not be related by R themselves, but they need to be obtainable by plugging related states into the 'holes' of the same *context*, which is an open term built with the operators under consideration (see, for example, Sangiorgi (1998)).

5.6. *Some instances of the format*

The Hamming Numbers example does not exploit all the power given by the framework: in the definition of the ρ^σ the tails are specified by applying the same operator once. The

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

framework allows several applications – possibly also including the other operators. In order to see a bit better which sets of operators Δ can be captured by this approach, we will look more closely at its instances for two functors F.

First, we consider again the setting of infinite streams A^ω of elements of some set A (that is, $\mathrm{F} := A \times \mathrm{Id}$). The principle allows us to work with sets Δ of operators such that for every $\delta \in \Delta$ with arity n there are functions $h^\delta : A^n \to A$ and $t^\delta : A^n \to \mathrm{T}\{x_1, \ldots, x_n, x_1', \ldots, x_n'\}$ such that for $\tau_i = \langle a_i : \tau_i' \rangle \in A^\omega$ we have

$$\begin{aligned}\mathtt{head}(\delta(\tau_1, \ldots, \tau_n)) &= h^\delta(a_1, \ldots, a_n),\\ \mathtt{tail}(\delta(\tau_1, \ldots, \tau_n)) &= [\![t^\delta(a_1, \ldots, a_n)[x_i := \tau_i, x_i' := \tau_i']]\!]_\Delta.\end{aligned}$$

To mimic the famous rule notation for non-deterministic transition systems, we could write $\tau \xrightarrow{t_0} \tau'$ for $\tau = \langle t_0 : \tau' \rangle$ and denote such a definition by giving for every $a_1, \ldots, a_n \in A$ a rule of the shape

$$\frac{x_i \xrightarrow{a_i} x_i' \qquad 1 \leqslant i \leqslant n}{\delta(x_1, \ldots, x_n) \xrightarrow{h^\delta(a_1, \ldots, a_n)} t^\delta(a_1, \ldots, a_n)}$$

The two binary operators $\oplus$ and $\otimes$ on streams of real numbers, for example, arise by declaring for all $r_1, r_2 \in I\!R$ the rules

$$\frac{x_1 \xrightarrow{r_1} x_1' \quad x_2 \xrightarrow{r_2} x_2'}{x_1 \oplus x_2 \xrightarrow{r_1 + r_2} x_1' \oplus x_2'} \qquad \frac{x_1 \xrightarrow{r_1} x_1' \quad x_2 \xrightarrow{r_2} x_2'}{x_1 \otimes x_2 \xrightarrow{r_1 \cdot r_2} (x_1' \otimes x_2) \oplus (x_1 \otimes x_2')}$$

Like these, most of the operators defined by Rutten (Rutten 2000a) fit into this format.

Consider a system of *guarded recursive equations*, which consists of two equations $\mathtt{head}(x) = h_x$ and $\mathtt{tail}(x) = t_x$ for all x in a (not necessarily finite) set of variables X, where $h_x \in A$ and $t_x \in \mathrm{T}X$. Corollary 5.7 yields a unique solution for such a system, which is an assignment of streams to the variables making the equations hold. (We use the term 'guarded' to express the fact that for every variable $x \in X$ the equations immediately provide the first element h_x of the stream to be assigned to x.)

As an example of the use of bisimulations up-to-context in this setting, we show that $\otimes$ distributes over $\oplus$: this follows from Corollary 5.8 instantiated with the set $\Delta := \{\oplus, \otimes\}$ containing the two operators under consideration and the relation

$$R := \{\langle \sigma \otimes (\tau \oplus \rho), (\sigma \otimes \tau) \oplus (\sigma \otimes \rho) \rangle \mid \sigma, \tau, \rho \in I\!R^\omega\}.$$

To see that it is a bisimulation up-to-context, one first easily checks that all related states have equal heads. Next, for streams σ, τ, and ρ with tails σ', τ' and ρ', we further compute (using associativity and commutativity of $\oplus$)

$$\begin{aligned}\mathtt{tail}(\sigma \otimes (\tau \oplus \rho)) &= \underbrace{(\sigma \otimes (\tau' \oplus \rho'))}_{=:x_1} \oplus \underbrace{(\sigma' \otimes (\tau \oplus \rho))}_{=:x_2},\\ \mathtt{tail}((\sigma \otimes \tau) \oplus (\sigma \otimes \rho)) &= \underbrace{((\sigma \otimes \tau') \oplus (\sigma \otimes \rho'))}_{=:y_1} \oplus \underbrace{((\sigma' \otimes \tau) \oplus (\sigma' \otimes \rho))}_{=:y_2},\end{aligned}$$

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

We need to show $\langle x_1 \oplus x_2, y_1 \oplus y_2 \rangle \in R^\Delta$. This follows easily by applying the closure condition for $\oplus$ once, since $\langle x_i, y_i \rangle \in R \subseteq R^\Delta$ for $i = 1, 2$.

The above identity is one of those Rutten proves for his stream calculus (Rutten 2000a). In a similar way, the up-to-context principle enables simpler proofs for many of the equations he states, for example, in Theorem 4.1 in *loc. cit.*

As a second example for a functor F, consider (image finite) labelled transition systems with label set A. They can be modelled as coalgebras of the functor $\mathrm{F} := (\mathscr{P}_\mathrm{f})^A$, where $\mathscr{P}_\mathrm{f}$ denotes the finite power set functor (a state of such a system is often called a *process*). Turi and Plotkin (Turi and Plotkin 1997) have shown a close connection between sets of natural transformations of the type (10) instantiated by this functor and structural operational rules in GSOS format (Bloom *et al.* 1995). In this format, for $\delta \in \Delta$ with arity n and processes p_i, the outgoing transitions of $\delta(p_1, \ldots, p_n)$ are determined by the labels for which the p_i have transitions, and they lead to states that can be composed of the p_i and their immediate successors by the operators in Δ.

6. Further properties and instances

As an interesting, though trivial, observation, note that by taking T to be the identity functor (monad) and λ to be the identity natural transformation, the coiteration schema itself arises as the λ-coiteration schema and λ-bisimulations are ordinary bisimulations.

Furthermore, many single instances of the framework are extensions of these basic schemata, in particular, in the following situation. When there is a unit natural transformation $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$ for which λ satisfies the unit law from Definition 5.1, a coiterative arrow f from the F-coalgebra $\langle X, \alpha \rangle$ can be obtained as the λ-coiterative arrow from the FT-coalgebra $\langle X, \mathrm{F}\eta_X \circ \alpha \rangle$, and, with a similar construction, every ordinary bisimulation on the final coalgebra is a λ-bisimulation. This is because, using an argument similar to that for Lemma 5.3, the assumption on λ makes β_λ satisfy the unit law from Definition 5.1 which yields $f|^{\beta_\lambda} \circ \eta_X = f$.

One simple extension of the coiteration schema arises as the dual of primitive recursion and is therefore sometimes called *primitive corecursion* (see, for example, Uustalu and Vene (1999)). It states that in a category with binary coproducts, every arrow $\phi : X \to \mathrm{F}(X + \Omega_\mathrm{F})$ uniquely determines a morphism $f : X \to \Omega_\mathrm{F}$ making the diagram below commute:

$$\begin{array}{ccc} X & \overset{\exists! f}{\dashrightarrow} & \Omega_\mathrm{F} \\ {\scriptstyle \forall\phi}\downarrow & \text{primitive corecursion} & \downarrow{\scriptstyle \omega_\mathrm{F}} \\ \mathrm{F}(X + \Omega_\mathrm{F}) & \underset{\mathrm{F}[f, \mathtt{id}]}{\dashrightarrow} & \mathrm{F}\Omega_\mathrm{F} \end{array}$$

This characterisation can be obtained from the λ-coiteration schema for the functor $\mathrm{T} := \mathrm{Id} + \Omega_\mathrm{F}$ (which can be extended to a monad) and the distributive law

$$\lambda : \mathrm{TF} \Rightarrow \mathrm{FT} \quad \text{defined as} \quad \lambda_X := [\mathrm{F}\,\mathtt{in}_l, \mathrm{F}\,\mathtt{in}_r \circ \omega_\mathrm{F}] : \mathrm{F}X + \Omega_\mathrm{F} \to \mathrm{F}(X + \Omega_\mathrm{F}).$$

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

On the final F-coalgebra the corresponding λ-bisimulations could be called *bisimulations up-to-equality*, since the larger bisimulation constructed corresponds to the reflexive closure of the original relation.

The schema that arises as the dual of (a categorical presentation of) course-of-value iteration (as given also in Uustalu and Vene (1999)) can also be obtained as an instance of λ-coiteration. For space limitations, we do not present details here, but we claim that for this second example the use of the λ-coiteration framework simplifies the justification of the schema considerably when compared to a proof from scratch.

For a more concrete instance consider the functor $\mathrm{F} := 2 \times (\mathrm{Id})^A$ capturing deterministic automata with alphabet A. The set of languages over A, $\mathscr{L} := \mathscr{P}(A^*)$, carries a final F-coalgebra structure (Rutten 1998). The coiterative arrow from any deterministic automata (that is, F-coalgebra) to this final one assigns to each state the language it accepts. Non-deterministic automata can be described as $\mathrm{F}\mathscr{P}$-coalgebras. A distributive law λ of the power set monad over F can be given such that the λ-coiterative arrow from an $\mathrm{F}\mathscr{P}$-coalgebra to the final F-coalgebra above captures the classical definition of the language accepted by a state in a non-deterministic automaton.

7. Related and future work

Our use of a second functor T to generalise coinductive definition schemata was inspired by work of Lenisa (Lenisa 1999), who was probably the first to give a framework for extended formats on the categorical level. She introduced the principle of *coiteration up-to-*$\mathscr{T}$ for a *pointed functor* $\mathscr{T} = \langle \mathrm{T}, \eta \rangle$ – that is, a functor with a natural transformation $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$. Distributive laws and FT-coalgebras are not mentioned in her schema in the first place, but they appear later in a proof principle for certain arrows coiterative up-to-$\mathscr{T}$ (although they appear as a technical prerequisite rather then being viewed as specifications determining the resulting schema).

Lemma 5.5 shows that in the setting involving a monad $\langle \mathrm{T}, \eta, \mu \rangle$ whose structure is respected by the distributive law λ, the λ-coiterative arrows from a set X factor as $h \circ \eta_X$ for a coiterative arrow h, which is to say they form a special class of arrows coiterative up-to-$\langle \mathrm{T}, \eta \rangle$ in the sense of Lenisa. Her framework yields a statement about the equivalences induced by such arrows, a subject that we have not devoted much attention to here admittedly. The technical assumption in her main theorem (stated in a revised version in Lenisa *et al.* (2000) as Theorem 6.9 based on Theorem 6.6) is almost equivalent to the statement in Lemma 5.4, and thus satisfied, but unfortunately the resulting principle in this case does not go beyond Corollary 3.9.

For the mentioned statement, Lenisa introduced the notion of a *bisimulation up-to-*$\mathscr{T}$ between coalgebras of a special shape as a proof tool. We found that the concept is useful for other types of bisimilarity proofs as well. A generalisation to arbitrary λ-bialgebras resulted in our notion of a λ-bisimulation.

For the functional programming community, Pardo, Uustalu and Vene (Uustalu *et al.* 2001) have recently, but independently from us, introduced a framework for generalised inductive definitions parametric in a comonad over which the algebra functor distributes. Their schema turns out to be the dual of the mentioned version of the λ-coiteration

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

schema involving a monad and a plain functor. They show that standard iteration, primitive recursion, and course-of-value iteration arise as instances of their format, and they give an implementation in the functional language Haskell. Their work does not contain a dual to our treatment of λ-bisimulations and to our schema involving auxiliary operators (Corollary 5.7). Interestingly, the latter does dualise to their setting, which yields formats going beyond the instances considered in *loc. cit.* in that they allow the value of $f(t)$ to depend on function values $f(t')$ for certain t' that are not necessarily sub-terms of t. We plan to present the details of this instance in a forthcoming paper.

Various other individual definition schemata for arrows into a final coalgebra that are developed from coiteration appear in the literature. As recent examples we mention the *Flattening Lemma* by Moss (Moss 2001, Lemma 2.1) and the *Solution Theorem* by Aczel, Adámek and Velebil (Aczel *et al.* 2001, Theorem 3.3), which can be seen as variants of primitive corecursion and the dual of course-of-value iteration, respectively.

We should stress that our framework (and its dual above) needs to be distinguished from another approach to generalise coinductive definition principles, which also involves a monad and a distributive law, but this time of the behavioural functor F over the monad (Power and Turi 1999). The aim there is to define arrows with a codomain constructed from a final coalgebra, like sets of streams or the choice between a standard behaviour and an exception (or, for the dual case, arrows from an initial datatype plus parameters (Pardo 2000)).

Our work on bisimilarity proofs is related to that of Sangiorgi (Sangiorgi 1998). He works more concretely with labelled transition systems in Set and develops a framework yielding sound conditions for bisimulations up-to. Our more global approach cannot handle all aspects he covers, like bisimulation up-to-bisimilarity, because it refers to the (local) bisimilarity relation on one specific coalgebra, but the theories overlap, for example, for the important instance of bisimulations up-to-context. Sangiorgi proves that the principle is sound for operators given by (unary) De Simone rules. Besides generalising the setting to arbitrary types of systems, we improve this result by showing that one can move to the more powerful format of GSOS rules.

We based this result on a categorical formulation of GSOS rules given by Turi and Plotkin (Turi and Plotkin 1997). They have shown that such specifications lead to distributive laws of a free monad over a cofree comonad. Lenisa *et al.* (2000) sharpened this result by showing that they correspond precisely to the class of distributive laws of the free monad over the cofree copointed functor. With the definition and proof principles from Corollaries 5.7 and 5.8, we can now justify the fact that this narrower class is interesting in its own right, because the statements do not hold in the larger class considered by Turi and Plotkin. This follows from an example of Sangiorgi's (see the end of Section 2 in Sangiorgi (1998)) demonstrating that the up-to-context proof technique is not sound for all operators. It turns out that the operators used in this example are definable by safe tree rules, a format covered by distributive laws of a monad over a comonad as well (Turi and Plotkin 1997).

Our framework yields yet another indication of the importance of the categorical treatment of GSOS rules. This makes it even more interesting to spell out this formulation

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

for further types of systems. As an example, we have made preliminary steps towards applying it in the area of probabilistic transition systems.

We have also left for future work a strengthening of the statement about equivalences induced by λ-coiterative arrows in Corollary 3.9. Another direction might be a study of invariance in this context.

Acknowledgements

Special thanks go to the anonymous referees for a number of valuable comments and hints. I would further like to thank my colleagues for discussions, in particular, Jan Rutten, Alexandru Baltag and Alexander Kurz.

References

Aczel, P., Adamek, J. and Velebil, J. (2001) A coalgebraic view of infinite trees and iteration. In: Corradini, A., Lenisa, M. and Montanari, U. (eds.) Proc. CMCS 2001. *Electronic Notes in Theoretical Computer Science* **44**.

Aczel, P. and Mendler, N. (1989) A final coalgebra theorem. In: Pitt, D., Rydeheard, D., Dybjer, P., Pitts, A. and Poigné, A. (eds.) Proc. 3rd CTCS. *Springer-Verlag Lecture Notes in Computer Science* **389** 357–365.

Bartels, F. (2001) Generalised coinduction. In: Corradini, A., Lenisa, M. and Montanari, U. (eds.) Proc. CMCS 2001. *Electronic Notes in Theoretical Computer Science* **44**.

Bloom, B., Istrail, S. and Meyer, A. R. (1995) Bisimulation can't be traced. *Journal of the ACM* **42** (1) 232–268.

Dijkstra, E. W. (1981) Hamming's exercise in SASL. Personal Note EWD792, see `http://www.cs.utexas.edu/users/EWD/`.

Jacobs, B. and Rutten, J. (1996) A tutorial on (co)algebras and (co)induction. *Bulletin of the EATCS* **62** 222–259.

Lenisa, M. (1999) From set-theoretic coinduction to coalgebraic coinduction: some results, some problems. In: Jacobs, B. and Rutten, J. (eds.) Proc. CMCS 1999. *Electronic Notes in Theoretical Computer Science* **19** 1–21.

Lenisa, M., Power, J. and Watanabe, H. (2000) Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads. In: Reichel, H. (ed.) Proc. CMCS 2000 *Electronic Notes in Theoretical Computer Science* **33** 233–263.

Milner, R. (1989) *Communication and Concurrency*, International Series in Computer Science, Prentice Hall.

Moss, L. S. (2001) Parametric corecursion. *Theoretical Computer Science* **260** (1-2) 139–163.

Pardo, A. (2000) Towards merging recursion and comonads. In: Jeuring, J. (ed.) Proc. WGP'2000. Tech. Report UU-CS-2000-19, University of Utrecht 50–68.

Power, J. and Turi, D. (1999) A coalgebraic foundation for linear time semantics. In: Hofmann, M., Pavlović, D. and Rosolini, G. (eds.) Proc. 8th CTCS Conf. *Electronic Notes in Theoretical Computer Science* **29**.

Power, J. and Watanabe, H. (1999) Distributivity for a monad and a comonad. In: Jacobs, B. and Rutten, J. (eds.) Proc. CMCS 1999. *Electronic Notes in Theoretical Computer Science* **19** 119–132.

Rutten, J. (1998) Automata and coinduction (an exercise in coalgebra). In: CONCUR'98. *Springer-Verlag Lecture Notes in Computer Science* **1466** 194–218.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press

Rutten, J. (2000a) Behavioural differential equations: a coinductive calculus of streams, automata, and power series. Technical Report SEN-R0023, CWI, Amsterdam.

Rutten, J. (2000b) Universal coalgebra: A theory of systems. *Theoretical Computer Science* **249** (1) 3–80.

Sangiorgi, D. (1998) On the bisimulation proof method. *Mathematical Structures in Computer Science* **8** 447–479.

Turi, D. and Plotkin, G. D. (1997) Towards a mathematical operational semantics. In: *Proc. 12th LICS Conf.*, IEEE, Computer Society Press 280–291.

Uustalu, T. and Vene, V. (1999) Primitive (co)recursion and course-of-value (co)iteration, categorically. *Informatica* (*IMI, Lithuania*) **10** (1) 5–26.

Uustalu, T., Vene, V. and Pardo, A. (2001) Recursion schemes from comonads. *Nordic Journal of Computing* **8** (3) 366–390.

https://doi.org/10.1017/S0960129502003900 Published online by Cambridge University Press