Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Generalised Coinduction

F. Bartels

Software Engineering (SEN)

SEN-R0043 December 31, 2000

Report SEN-R0043
ISSN 1386-369X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Generalised Coinduction

F. Bartels
Email: `falk.bartels@cwi.nl`,
URL: `www.cwi.nl/~bartels/`

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT
Final coalgebras of a functor F are suited for an abstract description of infinite datatypes and dynamical systems. Functions into such a domain are specified by coinductive definitions. The format these specifications take when their justification is directly based on finality is called the *coiteration schema* here. In applications it often turns out to be too rigid to allow for a convenient description of the functions under consideration. Thus, generalisations or variations are desired.

We introduce a generic λ-*coiteration* schema that can be instantiated by a distributive law λ of some functor T over F and show that – under mild assumptions on the underlying category – one obtains principles which uniquely characterise arrows into the carrier of a final F-coalgebra as well. Certain instances of λ-coiteration can be shown to specify arrows that fail to be coiterative. Examples are the duals of primitive recursion and course-of-value iteration, which are known extensions of coiteration. One can furthermore obtain schemata justifying recursive specifications that involve operators such as arithmetic operations on power series, regular operators for languages, or parallel and sequential composition of processes.

Next, the same type of distributive law λ is used to generalise coinductive proof techniques. To this end, we introduce the notion of a λ-*bisimulation* relation, many instances of which are weaker than the conventional definition of a bisimulation. It specialises e.g. to what could be called bisimulation up-to-equality or bisimulation up-to-context for contexts built from operators of the type mentioned above. We give a proof showing that every λ-bisimulation only contains pairs of bisimilar states. This principle leads to simpler proofs through the use of less complex relations.

2000 Mathematics Subject Classification: 68Q65, 68Q85
Keywords & Phrases: Coalgebra, coinduction, bisimulation, bisimulation up-to-context, distributive law, bialgebra
Note: The research reported here was part of the NWO project ProMACS.

Contents

1 Introduction

In theoretical computer science, *initial algebras* are commonly used to formally specify finitely constructed datatypes like the natural numbers $\mathbb{N}$ or finite lists A^* over elements from a given set A. Around the early nineties, the dual concept of a *final coalgebra* was found useful for the abstract description of possibly infinite objects. These include datatypes such as infinite streams or dynamical systems like processes or automata.

Initial algebras come with the definition and proof principle of induction for functions having their carriers as the domain. Dually, final coalgebras provide the so called coinduction definition and proof principle for functions into their carrier. To distinguish the very basic formats of these definition schemata as given by initiality and finality from generalised variants, we will call them *iteration* and *coiteration* in this paper.

Since they offer only limited flexibility, it is reasonable to look for generalisations. In the algebra world those are familiar – to such an extent that the reader may not even recognise our presentation of the iteration schema as the basic one. Examples of generalised induction schemata range from primitive recursion and course-of-value iteration to well-founded recursion.

However, the situation with coalgebras seems less advanced. Dualisations of definition by primitive recursion and course-of-value iteration have been stated on an abstract level (see e.g. Vene/Uustalu [UV99]), but for many frequently encountered situations solutions are known for special cases only. One example is the use of operators in the specification of dynamical systems like sequential or parallel composition, as in process algebra. Similarly, one uses operators like addition and multiplication to combine formal power series [Rut00a] or regular operators on languages [Rut98a]. But these problems are usually studied in isolation.

When it comes to proof principles, again often the canonical tool for proving behavioural equivalence appears too rigid, namely that of *bisimulation relations*. These are relations on the state spaces of two coalgebras that are suitably closed under the coalgebra operations. The closure condition often forces the relations to be larger than desired in the following sense: starting with a relation containing all pairs of states to be shown bisimilar one needs to iteratively add new pairs encountered by applying the coalgebra operations to states already included. The bisimilarity of the new pairs needs to be established in order to conclude the bisimilarity one was initially interested in. But technically the process of enlarging the relation has to be continued even with pairs that can immediately be seen to be bisimilar. The simplest example arises when one is forced to relate a state to itself. Or one may need to add a pair of states the bisimilarity of which can be suitably derived from that of other pairs already present. It is interesting to look for weaker sufficient conditions for a relation to still only contain bisimilar states. The goal is to reduce the complexity of the relations exhibited and hence the amount of work needed for a proof. In the literature one can find some of these conditions for individual types of systems, which are sometimes called bisimulations up-to. In the two cases above one could talk about bisimulation up-to-equality or bisimulation up-to-context [San98].

The aim of the research presented here is to develop a categorical framework justifying generalised coinductive definition and proof schemata. In this setting one talks about F-coalgebras for a functor $\mathrm{F} : \mathsf{C} \to \mathsf{C}$, which describes the type of behaviour one is interested in. The approach we take is parametric in a distributive law λ of another functor T over F. Although we state our theory for some category C, all our explanations and examples work with the category of sets and total functions, Set.

The coiteration schema assigns infinite behaviours to the states of a set X by specifying for each element a direct observation and successor states determining the next layer of the behaviour. Since these successors are taken from X again, the observation can be continued with the same specification. A different approach is taken by the *λ-coiteration schema* that we introduce here. It allows these successors to be taken from $\mathrm{T}X$ for a functor T. This increases the expressiveness of the format in case $\mathrm{T}X$ can be regarded as being richer than X. For the observations to be continued with these successors, the distributive law λ of T over F is used to lift the specification for X to $\mathrm{T}X$.

We give two different sufficient conditions for the λ-coiteration schema to uniquely characterise behaviours. The first is that the underlying category has countable coproducts. The second assumes the functor T to be taken from a monad and λ to be a distributive law of this monad over F. The proof for the first condition is given in detail.

Making similar use of T and λ, we define the notion of a λ-*bisimulation* and show that under the assumptions from above it is a sufficient condition for all related states to be bisimilar. A small example demonstrates that the technique enables simpler proofs involving less complex relations.

We show that primitive corecursion and the dual of course-of-value iteration as presented by Vene and Uustalu [UV99] can be obtained from the λ-coiteration schema for suitable instantiations of T and λ. Moreover, we briefly explain how it can be used to justify the validity of specifications involving operators of a certain type. The same operators were considered by Turi and Plotkin [TP97] and shown to be closely related to those definable by structured transition rules in GSOS format [BIM95].

Showing that the schemata from above are instances of our framework at the same time produces the corresponding variants of the λ-bisimulation proof technique. In case of primitive corecursion and the dual of course-of-value iteration the conditions on the relations amount to what one could call bisimulation up-to-equality and multi-step bisimulation, the latter we do not know from the literature (see Section 6). For the case of definitions via operators one obtains a notion of bisimulation up-to-context [San98] for multivariate contexts (i.e. context with several "holes") built from operators of the type mentioned above.

We view our theory as an advancement of Lenisa's coiteration up-to-$\mathcal{T}$ definition and proof schema [Len99a], who presented the first categorical framework for generalised coinduction schemata. Furthermore, our notion of λ-bisimulation can be seen as a starting point for a categorical reformulation of Sangiorgi's set-theoretical bisimulation proof method for labeled transition systems [San98] (see Section 8 for a detailed comparison with related work).

1.1 Organisation of the Paper

In Section 2 we recall the definition of initial T-algebras and final F-coalgebras for functors T and F and demonstrate the use of the coiteration principle in an example. In Section 3 we consider two functions that cannot be handled by this basic format directly. We take them as a motivation to develop our λ-coiteration schema and give sufficient conditions for it to indeed uniquely define arrows into the final F-coalgebra. The following Section 4 is devoted to proof principles. After explaining the technique based on bisimulations, we consider again two problematic cases. They lead to the introduction of the notion of a λ-bisimulation and a corresponding proof principle. Section 5 extends the framework developed in the two preceding ones to allow for a more general type of distributive law. The following two Sections 6 and 7 treat slightly more advanced examples yielding a principle dual to course-of-value iteration and a format for recursive definitions using operators. Both make use of our most general version of the schema from Section 5. In the last two sections we treat the related work mentioned above in some more detail and give conclusions.

1.2 Preliminaries

The theory that we are going to present assumes some abstract category C to be given. Of course, the most important situation is $\mathsf{C} = \mathsf{Set}$, the category of sets and total functions. This was the starting point of our investigation and is our source of examples and motivation. We will also use sets and elements within informal explanations. Still, the theory is abstract enough to allow for other choices of C.

Most of the proofs in this paper are given in the diagrammatic proof style. For readability, usually the subparts of the diagrams are inscribed with a justification for them to commute.

By functoriality, the application of a functor F to every object and arrow of a commuting diagram yields a commuting diagram again. If this is used in a proof, we will inscribe $\mathrm{F}(< s >)$ in the latter diagram in case $< s >$ was the argument for the first diagram to commute. Here is an example:

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[d, "f"'] \arrow[dr, phantom, "\text{nat. } \eta"] & \mathrm{T}X \arrow[d, "\mathrm{T}f"] \\ Y \arrow[r, "\eta_Y"'] & \mathrm{T}Y \end{tikzcd} \quad \text{implies} \quad \begin{tikzcd} \mathrm{F}X \arrow[r, "\mathrm{F}\,\eta_X"] \arrow[d, "\mathrm{F}f"'] \arrow[dr, phantom, "\mathrm{F}(\text{nat. } \eta)"] & \mathrm{F}\mathrm{T}X \arrow[d, "\mathrm{F}\mathrm{T}f"] \\ \mathrm{F}Y \arrow[r, "\mathrm{F}\,\eta_Y"'] & \mathrm{F}\mathrm{T}Y \end{tikzcd}$$

The (co)projections and universal arrows (i.e. the pairing and case analysis) for a product $X \times Y$ and a coproduct $X + Y$ are denoted as in the following diagrams:

$$\begin{tikzcd} & Z \arrow[dl, "f"'] \arrow[d, "\langle f,g \rangle"] \arrow[dr, "g"] & \\ X & X \times Y \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & Y \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "\mathtt{in}_l"] \arrow[dr, "f"'] & X + Y \arrow[d, "{[f,g]}"] & Y \arrow[l, "\mathtt{in}_r"'] \arrow[dl, "g"] \\ & Z & \end{tikzcd}$$

The final object of a category will be denoted by 1. In Set we usually write $1 = \{*\}$ for an arbitrary singleton set.

2 (Co)Algebras and (Co)Iteration

In this section we will briefly present a categorical view on algebras and coalgebras. More detailed expositions can be found e.g. among the papers of Jacobs and Rutten [JR96, Rut00b]. We take C to denote a category and $\mathrm{T}, \mathrm{F} : \mathsf{C} \to \mathsf{C}$ for two functors on it.

Definition 2.1 (T-algebra, F-coalgebra) *A* T-algebra *is a pair $\langle X, \beta \rangle$ where X is an object of C and $\beta : \mathrm{T}X \to X$ is an arrow in C. 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 $\alpha : X \to \mathrm{F}X$ is an arrow going into the reversed direction.*

Example 2.2 *Consider the Set functors* N *and* S *defined as*

$$\mathrm{N}X := 1 + X \qquad \mathrm{N}f := \mathtt{Id}_1 + f$$

and

$$\mathrm{S}X := I\!R \times X \qquad \mathrm{S}f := \mathtt{Id}_{I\!R} \times f$$

for a set X, a function f, and the singleton set $1 := \{*\}$.

Given a set X, a constant $z \in X$ and an operation $s : X \to X$, we get an N*-algebra $\langle X, [z, s] \rangle$. One concrete instance of this is the algebra of natural numbers $\langle I\!N, [0, . + 1] \rangle$.*

Given again a set X and this time two functions $o : X \to I\!R$ and $s : X \to X$, we get an S*-coalgebra $\langle X, \langle h, t \rangle \rangle$. The infinite streams of real numbers for example form the* S*-coalgebra $\langle I\!R^\omega, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$, where for $\sigma = \langle \sigma_0, \sigma_1, \ldots \rangle \in I\!R^\omega$ we set*

$$\mathtt{head}(\sigma) := \sigma_0 \quad \textit{and} \quad \mathtt{tail}(\sigma) := \langle \sigma_1, \sigma_2, \ldots \rangle.$$

We will use these streams for most of our examples. They are often denoted by σ, τ, or ρ. We often write $\sigma_i \in I\!R$ for the i-th element of a stream $\sigma \in I\!R^\omega$, in particular σ_0 for $\mathtt{head}(\sigma)$, and $\sigma' \in I\!R^\omega$ for $\mathtt{tail}(\sigma)$.

Generally, algebra operations can be seen as a means for *constructing* elements of their carrier. In the case of the N-algebra $\langle X, [z, s] \rangle$ from above either as the constant z or as the successor $s(x)$ of another element $x \in X$. On the other hand, the operation of a coalgebra – also called *destruction* or *unfolding* elsewhere – gives us information about its states, either in terms of attributes or (potential) successor states. In the first case we will talk about the *observation* a state allows, in the second about its *dynamics*. In the concrete example of an S-coalgebra $\langle X, \langle o, s \rangle \rangle$ from the example above, for every element $x \in X$ the observation is given by the attribute $o(x) \in I\!R$ and the dynamics by a successor state $s(x) \in X$.

If the dynamics of a state $x \in X$ inside the F-coalgebra $\langle X, \alpha \rangle$ leads to other states, like with $s(x)$ above, then we can repeatedly apply the operation α to those successors. If we assume that we have no other way of inspecting the states in X, the observation of all these successively reachable states may still give us quite some information about the element x that we started out with, and we will call it its *behaviour*. For example, an element $x \in X$ in some S-coalgebra $\langle X, \langle o, s \rangle \rangle$ gives rise to the infinite sequence of successive states

$$\langle x, s(x), s(s(x)), \ldots \rangle \in X^\omega$$

for each of which all we can observe is its attribute in $I\!R$. Thus, the behaviour of x is described by the infinite stream

$$\langle o(x), o(s(x)), o(s(s(x))), \ldots \rangle \in I\!R^\omega.$$

This is why we will call S-coalgebras *stream systems* in the following.

Definition 2.3 (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 the following diagram commute:*

$$\begin{tikzcd}
\mathrm{T}X \arrow[r, "\mathrm{T}h"] \arrow[d, "\beta_X"'] & \mathrm{T}Y \arrow[d, "\beta_Y"] \\
X \arrow[r, "h"'] & Y
\end{tikzcd}$$

Similarly, an F-coalgebra homomorphism *from one* F*-coalgebra $\langle X, \alpha_X \rangle$ to another* F*-coalgebra $\langle Y, \alpha_Y \rangle$ is an arrow $h : X \to Y$ making this diagram commute:*

$$\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "\alpha_X"'] & Y \arrow[d, "\alpha_Y"] \\
\mathrm{F}X \arrow[r, "\mathrm{F}h"'] & \mathrm{F}Y
\end{tikzcd}$$

We will often just talk about homomorphisms when it is clear from the context whether T-algebra or F-coalgebra homomorphisms are meant.

Example 2.4 *Consider again the functors* N *and* S *form Example 2.2.*

Given two N*-algebras $\langle X, [z_X, s_X] \rangle$ and $\langle Y, [z_Y, s_Y] \rangle$, a function $h : X \to Y$ is an* N*-algebra homomorphism if we have*

$$h(z_X) = z_Y \quad \textit{and} \quad h \circ s_X = s_Y \circ h.$$

Similarly, a function $h : X \to Y$ is a homomorphism from one S*-coalgebra $\langle X, \langle o_X, s_X \rangle \rangle$ to another* S*-coalgebra $\langle Y, \langle o_Y, s_Y \rangle \rangle$ if we have*

$$o_X = o_Y \circ h \quad \textit{and} \quad h \circ s_X = s_Y \circ h.$$

Since identities can easily be seen to form homomorphisms and homomorphisms compose, we get two categories: $\mathsf{Alg}^{\mathrm{T}}$ and $\mathsf{Coalg}_{\mathrm{F}}$, categories having as objects T-algebras and F-coalgebras respectively and as arrows homomorphisms of the appropriate type. Both categories come with a forgetful functor $\mathrm{U}^{\mathrm{T}} : \mathsf{Alg}^{\mathrm{T}} \to \mathsf{C}$ and $\mathrm{U}_{\mathrm{F}} : \mathsf{Coalg}_{\mathrm{F}} \to \mathsf{C}$. They forget about the algebra and coalgebra operations, i.e. they map algebras and coalgebras onto their carriers and homomorphisms to the underlying C arrows.

Definition 2.5 (initial T-algebra, final F-coalgebra) *An* initial T-algebra *is an initial object in the category of* T*-algebras* $\mathsf{Alg}^{\mathrm{T}}$*, that is an algebra* $\langle \Delta_{\mathrm{T}}, \delta_{\mathrm{T}} \rangle$ *such that there exists exactly one homomorphism from it to every other* T*-algebra.*
Dually, a final F-coalgebra *is a final object in the category of* F*-coalgebras* $\mathsf{Coalg}_{\mathrm{F}}$*, that is a coalgebra – usually denote here by* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ *– such that there exists exactly one homomorphism from every other* F*-coalgebra to it.*

Example 2.6 *Consider again the functors* N *and* S *from Example 2.2.*

For an arbitrary N*-algebra* $\langle X, [z, s] \rangle$ *we get a homomorphism*

$$f : \langle \mathbb{N}, [0, . + 1] \rangle \to \langle X, [z, s] \rangle$$

by setting $h(n) := s^n(z)$*.*

Given an S*-coalgebra* $\langle X, \langle o, s \rangle \rangle$*, we get a homomorphism*

$$h : \langle X, \langle o, s \rangle \rangle \to \langle \mathbb{R}^{\omega}, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle,$$

mapping every element $x \in X$ *to the stream* $\langle o(x), o(s(x)), o(s(s(x))), \ldots \rangle$*.*

One can actually show that these homomorphisms are the only ones between the respective algebras and coalgebras. Thus, $\langle \mathbb{N}, [0, . + 1] \rangle$ *is an initial* N*-algebra and* $\langle \mathbb{R}^{\omega}, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ *is a final* S*-coalgebra (see e.g. Rutten's work on stream calculus [Rut00a] for the latter statement).*

In the above example the carrier of the final coalgebra contains exactly those elements that we used above to informally describe the behaviour of a state in a stream system. This is an instance of the general observation that the states of a final coalgebra – if it exists – represent abstract behaviours: Since it can be argued that the behaviour is preserved by homomorphisms, the existence of a homomorphism in the definition of a final coalgebra says that any behaviour exhibited by some state in some coalgebra can be matched by an element of Ω_{F}. The uniqueness of homomorphisms on the other hand says that there is only one such candidate. In that sense, the coiterative morphism maps a state of a coalgebra to the abstract behaviour it exhibits.

We will now give a name to the use of the existence aspect of initiality or finality of an algebra or coalgebra as a definition principle:

The initiality of the T-algebra $\langle \Delta_{\mathrm{T}}, \delta_{\mathrm{T}} \rangle$ provides us with a unique arrow $h : \Delta_{\mathrm{T}} \to X$ for every T-algebra operation $\beta : \mathrm{T}X \to X$ that is a homomorphism from $\langle \Delta_{\mathrm{T}}, \delta_{\mathrm{T}} \rangle$ to $\langle X, \beta \rangle$. We call h the *iterative arrow defined (or induced) by* β, since – as in Example 2.6 – the function value cana usually be obtained by iteratively applying the constructors of the algebra:

$$\begin{tikzcd}
\mathrm{T}\,\Delta_{\mathrm{T}} \arrow[r, dashed, "\mathrm{T}\,h"] \arrow[d, "\delta_{\mathrm{T}}"'] & \mathrm{T}\,X \arrow[d, "\forall \beta"] \\
\Delta_{\mathrm{T}} \arrow[r, dashed, "\exists! h"'] & X
\end{tikzcd}$$

Dually, the finality of an F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ yields an arrow $h : X \to \Omega_{\mathrm{F}}$ for every F-coalgebra operation α on X by assuming it to be 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 induced) by* α:

$$\begin{tikzcd}
X \arrow[r, dashed, "\exists! h"] \arrow[d, "\forall \alpha"'] & \Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}X \arrow[r, dashed, "\mathrm{F}\,h"'] & \mathrm{F}\,\Omega_{\mathrm{F}}
\end{tikzcd}$$

Example 2.7 (Coiteration for Streams) *The coiteration schema for stream systems from Example 2.2 states that for every* S*-coalgebra* $\langle X, \langle o, s \rangle \rangle$ *there is a unique arrow* $h : X \to \mathbb{R}^{\omega}$ *making the diagram below commute:*

$$\begin{tikzcd}
X \arrow[r, dashed, "\exists! h"] \arrow[d, "\forall \langle o,s \rangle"'] & \mathbb{R}^\omega \arrow[d, "\langle \mathtt{head},\mathtt{tail} \rangle"] \\
\mathbb{R} \times X \arrow[r, dashed, "\mathtt{Id}_{\mathbb{R}} \times h"'] & \mathbb{R} \times \mathbb{R}^\omega
\end{tikzcd}$$

In other words, every pair of functions $o : X \to \mathbb{R}$ and $s : X \to X$ defines a function $h : X \to \mathbb{R}^\omega$ by assuming it to satisfy the equations

$$\begin{aligned}
\mathtt{head}(h(x)) &= o(x), \\
\mathtt{tail}(h(x)) &= h(s(x)).
\end{aligned}$$

Example 2.8 *As an example for a function into the streams of reals we take a look at the elementwise addition of two such streams. By the coiteration schema there is a unique function $\oplus$: $\mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ satisfying the following two equations:*

$$\begin{aligned}
\mathtt{head}(\sigma \oplus \tau) &= \mathtt{head}(\sigma) + \mathtt{head}(\tau) \\
\mathtt{tail}(\sigma \oplus \tau) &= \mathtt{tail}(\sigma) \oplus \mathtt{tail}(\tau)
\end{aligned}$$

Whereas the existence part of initiality and finality of an algebra or coalgebra provides arrows in a definition principle, the uniqueness aspect brings about a principle to prove two given arrows equal:

To show that for an initial T-algebra $\langle \Delta_{\mathrm{T}}, \delta_{\mathrm{T}} \rangle$ two arrows $h_1, h_2 : \Delta_{\mathrm{T}} \to X$ are equal, it suffices to come up with a T-operation β on X such that both, h_1 and h_2, are homomorphisms from $\langle \Delta_{\mathrm{T}}, \delta_{\mathrm{T}} \rangle$ to $\langle X, \beta \rangle$. We will call this the *induction proof principle.* The reader probably associates with this name a principle that in the example of the natural numbers for a predicate $P \subseteq \mathbb{N}$ looks as follows:

$$(0 \in P \wedge \forall n \in \mathbb{N} : n \in P \Rightarrow n + 1 \in P) \Rightarrow \forall n \in \mathbb{N} : n \in P.$$

But as argued by Jacobs and Rutten [JR96, Section 5], this way of stating the principle with predicates is equivalent to our formulation based on initiality.

In a dual manner, one can show that for a final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ two arrows $h_1, h_2 : X \to \Omega_{\mathrm{F}}$ are equal by providing an F-coalgebra operation on X for which both arrows are homomorphisms. But this schema is usually not applied directly. Instead, one uses a technique derived from it which is based on the notion of *bisimulation.* We will present it in detail in Section 4.

3 Definition by λ-Coiteration

As already mentioned, the states of a final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ represent abstract behaviours of the type F. Thus, an arrow $f : X \to \Omega_{\mathrm{F}}$ assigns such behaviours to the elements of X. The coiteration schema allows us to define this function such that it maps every element of X to the behaviour it exhibits as a state in a given F-coalgebra $\langle X, \alpha \rangle$. Unfortunately, for many functions $f : X \to \Omega_{\mathrm{F}}$ that one wants to specify there is no F-coalgebra operation α on X itself 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 two examples of this kind. As it turns out both share a common underlying pattern which will then lead to the definition of a framework that we called λ-*coiteration.* As the main contribution of this paper we later establish sufficient conditions for the instances of it to be valid definition schemata in the sense that they uniquely characterise arrows.

3.1 Example 1: Multiplication of Formal Series

Like Mc Ilroy [McI99] or Rutten [Rut00a], we will this time consider infinite streams of real numbers $\sigma = \langle \sigma_0, \sigma_1, \ldots \rangle \in \mathbb{R}^\omega$ as representations of formal power series $\sum_{i=0}^{\infty} \sigma_i X^i$. The operation $\oplus$ from Example 2.8 turns out to be such that $\sigma \oplus \tau$ represents the sum of the two series represented by σ and τ, i.e.

$$\sum_{i=0}^{\infty} \sigma_i X^i + \sum_{i=0}^{\infty} \tau_i X^i = \sum_{i=0}^{\infty} (\sigma_i + \tau_i) X^i.$$

We would like to define an operation $\otimes$ on $\mathbb{R}^\omega$ such that $\sigma \otimes \tau$ represents the convolution product of the two series represented by σ and τ, i.e.

$$\sum_{i=0}^{\infty} \sigma_i X^i \cdot \sum_{i=0}^{\infty} \tau_i X^i = \sum_{i=0}^{\infty} (\sum_{j=0}^{i} \sigma_j \tau_{i-j}) X^i.$$

The expression on the right hand side of the equation can be rewritten to

$$\sigma_0 \tau_0 + X \cdot (\sigma_0 \cdot \sum_{i=0}^{\infty} \tau_{i+1} X^i + \sum_{i=0}^{\infty} \sigma_{i+1} X^i \cdot \sum_{i=0}^{\infty} \tau_i X^i).$$

The single constant $\sigma_0 = \mathtt{head}(\sigma)$ can be represented by the series $[\mathtt{head}(\sigma)]$ where for $r \in \mathbb{R}$ we set $[r] = \langle r, 0, 0, \ldots \rangle \in \mathbb{R}^\omega$ (or coiteratively: $\mathtt{head}([r]) = r$ and $\mathtt{tail}([r]) = [0]$). Furthermore the series $\sum_{i=0}^{\infty} \sigma_{i+1} X^i$ is represented by $\mathtt{tail}(\sigma)$ and the same for τ. Thus, we arrive at the following equations for the multiplication of streams (see also Rutten [Rut00a]):

$$\begin{aligned} \mathtt{head}(\sigma \otimes \tau) &= \mathtt{head}(\sigma) \cdot \mathtt{head}(\tau), \\ \mathtt{tail}(\sigma \otimes \tau) &= ([\mathtt{head}(\sigma)] \otimes \mathtt{tail}(\tau)) \ \oplus \ (\mathtt{tail}(\sigma) \otimes \tau). \end{aligned}$$

These two equations do not form a coiterative definition of a stream 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 $s_1, s_2 : X \to X, o : X \to \mathbb{R}$ with

$$\begin{aligned} o(\sigma, \tau) &:= \mathtt{head}(\sigma) \cdot \mathtt{head}(\tau), \\ s_1(\sigma, \tau) &:= \langle [\mathtt{head}(\sigma)], \mathtt{tail}(\tau) \rangle, \\ s_2(\sigma, \tau) &:= \langle \mathtt{tail}(\sigma), \tau \rangle. \end{aligned}$$

Now the question is whether there exists a unique function $\otimes : X \to \mathbb{R}^\omega$ satisfying for $x \in X$

$$\begin{aligned} \mathtt{head}(\otimes(x)) &= o(x), \\ \mathtt{tail}(\otimes(x)) &= \otimes(s_1(x)) \ \oplus \ \otimes(s_2(x)), \end{aligned}$$

or, diagrammatically, a unique arrow $\otimes$ fitting into the following diagram:

$$\begin{tikzcd} X \arrow[r, dashed, "\otimes"] \arrow[d, "{\langle o, \langle s_1, s_2 \rangle \rangle}"'] & \mathbb{R}^\omega \arrow[d, "{\langle \mathtt{head}, \mathtt{tail} \rangle}"] \\ \mathbb{R} \times (X \times X) \arrow[r, dashed, "\mathtt{Id} \times (\oplus \circ (\otimes \times \otimes))"'] & \mathbb{R} \times \mathbb{R}^\omega \end{tikzcd}$$

3.2 Example 2: Sorted Insertion into Streams

Assume we need a function that inserts a new element into a sorted stream of real numbers at such a position that the resulting stream is again sorted, if possible. Practically, we specify a function $\mathtt{insert} : \mathbb{R} \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ that inserts an element $r \in \mathbb{R}$ into a given stream $\sigma = \langle \sigma_0, \sigma_1, \ldots \rangle \in \mathbb{R}^\omega$

just in front of the first σ_i such that $r \leq \sigma_i$. The head and tail of the result of $\texttt{insert}(r, \sigma)$ can be characterised as follows:

$$\begin{aligned}
\texttt{head}(\texttt{insert}(r,\sigma)) &= \begin{cases} r & \text{if } r \leq \texttt{head}(\sigma), \\ \texttt{head}(\sigma) & \text{otherwise,} \end{cases} \\
\texttt{tail}(\texttt{insert}(r,\sigma)) &= \begin{cases} \sigma & \text{if } r \leq \texttt{head}(\sigma), \\ \texttt{insert}(r, \texttt{tail}(\sigma)) & \text{otherwise.} \end{cases}
\end{aligned}$$

Since in the case $r \leq \texttt{head}(\sigma)$ the tail of $\texttt{insert}(r,\sigma)$ is not specified as $\texttt{insert}(\tilde{r}, \tilde{\sigma})$ for new arguments $\tilde{R} \in \mathbb{R}$ and $\tilde{\sigma} \in \mathbb{R}^\omega$, the equations do not lead to an S-coalgebra on $X := \mathbb{R} \times \mathbb{R}^\omega$, the domain of `insert`. Instead, they give rise to operations $o : X \to A$ and $\tilde{s} : X \to X + \mathbb{R}^\omega$, where

$$\begin{aligned}
o(r,\sigma) &= \begin{cases} r & \text{if } r \leq \texttt{head}(\sigma), \\ \texttt{head}(\sigma) & \text{otherwise,} \end{cases} \\
\tilde{s}(r,\sigma) &= \begin{cases} \texttt{in}_r(\sigma) & \text{if } r \leq \texttt{head}(\sigma), \\ \texttt{in}_l(r, \texttt{tail}(\sigma)) & \text{otherwise.} \end{cases}
\end{aligned}$$

That `insert` satisfies the above equations is then equivalent to saying that it fits as the function into the diagram below:

$$\begin{tikzcd}
X \arrow[r, dashed, "\texttt{insert}"] \arrow[d, "\langle o, \tilde{s} \rangle"'] & \mathbb{R}^\omega \arrow[d, "\langle \texttt{head}, \texttt{tail} \rangle"] \\
\mathbb{R} \times (X + \mathbb{R}^\omega) \arrow[r, dashed, "\texttt{Id}_{\mathbb{R}} \times [\texttt{insert}, \texttt{Id}_{\mathbb{R}^\omega}]"'] & \mathbb{R} \times \mathbb{R}^\omega
\end{tikzcd}$$

This example fits into the known schema that arises as the dual of primitive recursion and is thus called *primitive corecursion* e.g. by Vene and Uustalu [VU97, UV99]. They give the following answer:

Theorem 3.1 (primitive corecursion) *Assume that the category* C *has binary coproducts and the functor* $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ *has the final coalgebra* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$. *Then for every object* X *and operation* $\phi : X \to \mathrm{F}(X + \Omega_{\mathrm{F}})$ *there is a unique arrow* $f : X \to \Omega_{\mathrm{F}}$ *making the diagram below commute, called the* corecursive arrow induced by ϕ*:*

$$\begin{tikzcd}
X \arrow[r, dashed, "f"] \arrow[d, "\phi"'] & \Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}(X + \Omega_{\mathrm{F}}) \arrow[r, dashed, "\mathrm{F}[f, \texttt{Id}_{\Omega_{\mathrm{F}}}]"'] & \mathrm{F}\Omega_{\mathrm{F}}
\end{tikzcd}$$

3.3 The Common Pattern

We have seen specifications of two functions assigning behaviours to elements of a given set that do not fit into the original format of coiteration. And we have seen an extension of the coiteration schema that could be used to define the desired function in the second case, but it is still not sufficient for the first. Now we are going to explain that both specifications can be seen to fall into a similar pattern after all. This will pave the way for a generalisation of Theorem 3.1 covering both examples that we are going to develop in the next section.

To formulate this common pattern, we make the following two definitions:

Definition 3.2 (T-extension) *Let* $\beta : \mathrm{T}Y \to Y$ *be a* T*-algebra. For a given arrow* $f : X \to Y$, *we call* $f|^{\beta} := \beta \circ \mathrm{T}f : \mathrm{T}X \to Y$ *the* T-extension of f along β*:*

$$\begin{tikzcd}
\mathrm{T}X \arrow[r, "\mathrm{T}f"] \arrow[dr, dashed, "f|^{\beta}"'] & \mathrm{T}Y \arrow[d, "\beta"] \\
X \arrow[r, "f"'] & Y
\end{tikzcd}$$

Definition 3.3 (homomorphism up-to-β) *Let $\langle X, \phi\rangle$ be an FT-coalgebra and $\langle Y, \alpha\rangle$ an F-coalgebra with a T-algebra operation $\beta : \mathrm{T}Y \to Y$ on its carrier. An arrow $f : X \to Y$ is called a* homomorphism up-to-β *from $\langle X, \phi\rangle$ to $\langle Y, \alpha\rangle$, if it makes the following diagram commute (note that in the picture we added an arrow for β to visualise its typing, though it does not contribute to the commutativity expressed):*

$$\begin{tikzcd}
 & \mathrm{T}Y \arrow[d, "\beta"] \\
X \arrow[r, "f"] \arrow[d, "\phi"'] & Y \arrow[d, "\alpha"] \\
\mathrm{FT}X \arrow[r, "\mathrm{F}f|^{\beta}"'] & \mathrm{F}Y
\end{tikzcd}$$

For the two previous examples we can find functors T and T-algebra operations β on the final F-coalgebra such that the arrows under consideration become homomorphisms up-to-β:

- For the specification of $\otimes$ from Section 3.1 we set

$$\begin{array}{rcll} \mathrm{T}X & := & X \times X, & \text{for a set } X, \\ \mathrm{T}f & := & f \times f, & \text{for a function } f : X \to Y. \end{array}$$

  Now we can rewrite the arrow at the bottom of the diagram covering its specification as follows:

$$\begin{array}{rcl} \mathtt{Id}_{I\!R} \times (\oplus \circ (\otimes \times \otimes)) & = & \mathrm{S}(\oplus \circ (\otimes \times \otimes)) \\ & = & \mathrm{S}(\oplus \circ \mathrm{T}\otimes) \\ & = & \mathrm{S}\otimes|^{\oplus} \end{array}$$

  Thus, it turns out that we were looking for is a homomorphism up-to-$\oplus$ from the ST-coalgebra $\langle X, \langle o, \langle s_1, s_2\rangle\rangle\rangle$ to the final S-coalgebra $\langle I\!R^{\omega}, \langle \mathtt{head}, \mathtt{tail}\rangle\rangle$.

- In the case of primitive corecursion from Theorem 3.1 we set

$$\begin{array}{rcll} \mathrm{T}X & := & X + \Omega_{\mathrm{F}}, & \text{for an object } X, \\ \mathrm{T}f & := & f + \mathtt{Id}_{\Omega_{\mathrm{F}}}, & \text{for a function } f : X \to Y, \end{array}$$

  Again, the bottom arrow in the corresponding diagram can be rewritten:

$$\begin{array}{rcl} \mathrm{F}[f, \mathtt{Id}_{\Omega_{\mathrm{F}}}] & = & \mathrm{F}([\mathtt{Id}_{\Omega_{\mathrm{F}}}, \mathtt{Id}_{\Omega_{\mathrm{F}}}] \circ (f + \mathtt{Id}_{\Omega_{\mathrm{F}}})) \\ & = & \mathrm{F}([\mathtt{Id}_{\Omega_{\mathrm{F}}}, \mathtt{Id}_{\Omega_{\mathrm{F}}}] \circ \mathrm{T}f) \\ & = & \mathrm{F}f|^{\beta}, \end{array}$$

  where $\beta := [\mathtt{Id}_{\Omega_{\mathrm{F}}}, \mathtt{Id}_{\Omega_{\mathrm{F}}}] : \mathrm{T}\Omega_{\mathrm{F}} \to \Omega_{\mathrm{F}}$. Thus, the corecursive arrow induced by $\phi : X \to \mathrm{F}(X + \Omega_{\mathrm{F}})$ turns out to be a homomorphism up-to-β from the FT-coalgebra $\langle X, \phi\rangle$ to the final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$.

3.4 Distributive Laws

In the following section we will develop a framework in which a homomorphism up-to-β_{F} into a final F-coalgebra uniquely exists for a certain T-algebra operation β_{F} on its carrier. This result means that an FT-operation on an object X specifies an arrow $h : X \to \Omega_{\mathrm{F}}$, i.e. an assignment of abstract F-behaviours to the elements in X. Not surprisingly, this requires some general information about the behavioural effect of T. In our setting it is given as a *distributive law* of T over F defined below. The T-algebra operation β_{F} is a unique one that respects this behavioural meaning of T in a sense to be made precise shortly. It will turn out that in the case of our two driving examples, these distributive laws can be given such that the resulting T-algebra operations β_{F} are the ones considered above, namely $\oplus$ and $[\mathtt{Id}_{\Omega_{\mathrm{F}}}, \mathtt{Id}_{\Omega_{\mathrm{F}}}]$.

Definition 3.4 (distributive law) *Let* $\mathrm{T}, \mathrm{F} : \mathsf{C} \to \mathsf{C}$ *be two functors. A natural transformation* $\lambda : \mathrm{TF} \Rightarrow \mathrm{FT}$ *is called a* distributive law *of* T *over* F*. We will sometimes alternatively use the phrase that* T distributes over F via λ.

We call λ a *distributive law* to stress the analogy with the use of a natural transformation of the same type by Turi and Plotkin [TP97] and Lenisa [Len99b]. There the name is explained by the additional assumptions made on the interaction of λ with the monad from which T is taken (and similarly with the additional comonad structure for F in the first paper). Here we have neither monad nor comonad and hence λ is just a natural transformation, but it will still serve a similar purpose.

Regarding the elements from $\mathrm{T}X$ as structured entities containing elements from X as arguments, a distributive law tells us how to assign an F-step to such an entity given the steps the arguments can do. With this information, we can derive an F-coalgebra operation for $\mathrm{T}X$ from the F-coalgebra $\langle X, \alpha\rangle$: First we unfold the arguments inside an element form $\mathrm{T}X$ (i.e. we apply $\mathrm{T}\alpha$) and then we use λ_X to derive an F-step of the whole entity from those of the arguments.

Definition 3.5 (λ-lifting) *Given a distributive law* $\lambda : \mathrm{TF} \Rightarrow \mathrm{FT}$ *of a functor* T *over a functor* F*, we can lift* $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ *to the functor* $\mathrm{T}_\lambda : \mathsf{Coalg}_{\mathrm{F}} \to \mathsf{Coalg}_{\mathrm{F}}$ *on the* F*-coalgebras by setting*

$$\begin{aligned} \mathrm{T}_\lambda \langle X, \alpha\rangle &:= \langle \mathrm{T}X, \mathit{lift}^\lambda_\alpha\rangle \\ \mathrm{T}_\lambda h &:= \mathrm{T}h, \end{aligned}$$

for an F*-coalgebra* $\langle X, \alpha\rangle$ *and an* F*-homomorphism* h*, where*

$$\mathit{lift}^\lambda_\alpha := \lambda_X \circ \mathrm{T}\alpha : \mathrm{T}X \to \mathrm{FT}X.$$

For this definition to make sense we have to check that T indeed maps homomorphisms to homomorphisms for the respective coalgebras. This is where the assumption on λ being natural is needed:

Lemma 3.6 (see also [Rut00b, Theorem 15.3]) *Let* h *be an* F*-coalgebra homomorphism from* $\langle X, \alpha_X\rangle$ *to* $\langle Y, \alpha_Y\rangle$*, then* $\mathrm{T}h$ *is a homomorphism from* $\mathrm{T}_\lambda\langle X, \alpha_X\rangle$ *to* $\mathrm{T}_\lambda\langle Y, \alpha_Y\rangle$*.*

Proof: The statement is easily proved using the naturality of λ and the T-image of the assumption on h:

$$\begin{tikzcd}
\mathrm{T}X \arrow[r, "\mathrm{T}h"] \arrow[d, "\mathrm{T}\alpha_X"] \arrow[dd, bend right=60, "\mathit{lift}^\lambda_{\alpha_X}"'] \arrow[dr, phantom, "\mathrm{T}(\text{ass. } h)"] & \mathrm{T}Y \arrow[d, "\mathrm{T}\alpha_Y"'] \arrow[dd, bend left=60, "\mathit{lift}^\lambda_{\alpha_Y}"] \\
\mathrm{TF}X \arrow[r, "\mathrm{TF}h"] \arrow[d, "\lambda_X"] \arrow[dr, phantom, "\text{nat.} \lambda"] & \mathrm{TF}Y \arrow[d, "\lambda_Y"'] \\
\mathrm{FT}X \arrow[r, "\mathrm{FT}h"'] & \mathrm{FT}Y
\end{tikzcd}$$

□

Again, we will check what these new notions mean for our two examples:

- Consider again the specifications of $\otimes$ in Section 3.1. It involved the functor $\mathrm{T}X = X \times X$. A pair of states from a stream system here is intended to behave like the addition of the two streams. Thus our distributive law, which has to be of the shape $\lambda_X : (\mathbb{R} \times X) \times (\mathbb{R} \times X) \to \mathbb{R} \times (X \times X)$, will be defined as

  $$\lambda_X(\langle h_x, x' \rangle, \langle h_y, y' \rangle) := \langle h_x + h_y, \langle x', y' \rangle\rangle. \tag{1}$$

  It is easy to verify that we get a natural transformation indeed.

  Given a stream system $\langle X, \langle o, s \rangle\rangle$, the system $\mathrm{T}_\lambda \langle X, \langle o, s \rangle\rangle$ has pairs of states from X as states. Unfolding such a pair $\langle x, y \rangle$ yields the sum of the two observations $o(x) + o(y)$ and the pair of the successors $\langle s(x), s(y) \rangle$, that is the behaviour of $\langle x, y \rangle$ in $\mathrm{T}_\lambda \langle X, \langle o, s \rangle\rangle$ is the (stream) sum of the behaviours of x and y in $\langle X, \langle o, s \rangle\rangle$ as wanted.

- In the case of primitive corecursion (Theorem 3.1) we considered the functor $\mathrm{T}X := X + \Omega_\mathrm{F}$ where $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ is a final F-coalgebra. A distributive law of T over F is of the shape $\lambda_X : \mathrm{F}X + \Omega_\mathrm{F} \to \mathrm{F}(X + \Omega_\mathrm{F})$. We set it as

  $$\lambda_X := [\mathrm{F}\,\mathtt{in}_l, \mathrm{F}\,\mathtt{in}_r \circ \omega_\mathrm{F}]. \tag{2}$$

  For an F-coalgebra $\langle X, \alpha \rangle$, the states of the system $\mathrm{T}_\lambda \langle X, \alpha \rangle$ are either states from X or from Ω_F. They behave as the states in the corresponding systems would do. With other words, $\mathrm{T}_\lambda \langle X, \alpha \rangle$ is the coproduct of $\langle X, \alpha \rangle$ and $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ (see [Rut00b, Section 4.1]).

With the lifting from above, a distributive law λ assigns F-behaviours to the set $\mathrm{T}X$, given the behaviours of the elements from X appearing as arguments inside the elements form $\mathrm{T}X$. The generalised coinduction schema we are about to state and justify makes use of T-algebra operations β which preserve these behaviours. More precisely, for a given F-coalgebra $\langle X, \alpha \rangle$ we consider homomorphisms β from $\mathrm{T}_\lambda \langle X, \alpha \rangle$ to $\langle X, \alpha \rangle$. It turns out that this situation is captured by the notion of a λ-bialgebra occurring in the literature, thus we recall its definition here:

Definition 3.7 (T,F-bialgebra, λ-bialgebra) *A* T,F-bialgebra *is a triple $\langle X, \beta, \alpha \rangle$ of an object X and two arrows $\beta : \mathrm{T}X \to X$ and $\alpha : X \to \mathrm{F}X$, i.e. a* T*-algebra and an* F*-coalgebra operation on a common carrier.*
Given two T,F*-bialgebras $\langle X, \beta_X, \alpha_X \rangle$ and $\langle Y, \beta_Y, \alpha_Y \rangle$, a* T,F-bialgebra homomorphism *from $\langle X, \beta_X, \alpha_X \rangle$ to $\langle Y, \beta_Y, \alpha_Y \rangle$ is an arrow $h : X \to Y$ which makes the following diagram commute:*

$$\begin{tikzcd}
\mathrm{T}X \arrow[r, "\mathrm{T}h"] \arrow[d, "\beta_X"'] & \mathrm{T}Y \arrow[d, "\beta_Y"] \\
X \arrow[r, "h"] \arrow[d, "\alpha_X"'] & Y \arrow[d, "\alpha_Y"] \\
\mathrm{F}X \arrow[r, "\mathrm{F}h"'] & \mathrm{F}Y
\end{tikzcd}$$

I.e. it is both, a T*-algebra homomorphism and an* F*-coalgebra homomorphism. Like with* T*-algebras and* F*-coalgebras,* T,F*-bialgebras and their homomorphisms form a category, denoted by $\mathsf{Bialg}^\mathrm{T}_\mathrm{F}$.*
Given a distributive law $\lambda : \mathrm{TF} \Rightarrow \mathrm{FT}$ of T *over* F*, a* λ-bialgebra *is a* T,F*-bialgebra $\langle X, \beta, \alpha \rangle$ such that the following diagram commutes:*

$$\begin{tikzcd}
 & \mathrm{T}X \arrow[dl, "\mathrm{T}\alpha"'] \arrow[dd, "\beta"] \\
\mathrm{TF}X \arrow[dd, "\lambda_X"', "\lambda\text{-}\mathit{bialg.}"] & \\
 & X \arrow[dd, "\alpha"] \\
\mathrm{FT}X \arrow[dr, "\mathrm{F}\beta"'] & \\
 & \mathrm{F}X
\end{tikzcd}$$

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

Note that indeed the definition of $\langle X, \beta, \alpha\rangle$ being a λ-bialgebra is equivalent to saying that β is an F-coalgebra homomorphism from $\mathrm{T}_\lambda\langle X, \alpha\rangle$ to $\langle X, \alpha\rangle$ as wanted above:

$$\begin{tikzcd}
\mathrm{T}X \arrow[r, "\beta"] \arrow[d, "\mathrm{T}\,\alpha"] \arrow[dd, bend right=60, dashed, "lift_\beta^\lambda"'] & X \arrow[dd, "\alpha"] \\
\mathrm{T}\mathrm{F}X \arrow[d, "\lambda_X"] & \\
\mathrm{F}\mathrm{T}X \arrow[r, "\mathrm{F}\,\beta"'] & \mathrm{F}X
\end{tikzcd}$$

With this remark, for a final F-coalgebra $\langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ there is exactly one choice for a T-algebra operation $\beta_{\mathrm{F}} : \mathrm{T}\,\Omega_{\mathrm{F}} \to \Omega_{\mathrm{F}}$ such that $\langle\Omega_{\mathrm{F}}, \beta_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ is a λ-bialgebra, namely the coiterative morphism from $\mathrm{T}_\lambda\langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ to $\langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$.

Bialgebras for a distributive law play an important role in the paper by Turi and Plotkin [TP97] as well. There the functors T and F are taken from a monad and a comonad respectively. As already mentioned, this leads to extra assumptions on λ. Similarly, the algebras and coalgebras they consider are those for the monad and comonad. This carries over to the algebra and coalgebra operations allowed in their notion of a λ-bialgebra.

For the examples form Sections 3.1 and 3.2 the T-algebra operations considered in Section 3.3 are such that they turn the final coalgebra into a λ-bialgebra for the distributive laws considered above:

- It turns out that $\langle \mathbb{R}^\omega, \oplus, \langle\mathtt{head}, \mathtt{tail}\rangle\rangle$ is a λ-bialgebra for λ as in (1) on page 13, since for all $\sigma = \langle\sigma_0 : \sigma'\rangle, \tau = \langle\tau_0 : \tau'\rangle \in \mathbb{R}^\omega$ we have:

$$\begin{aligned}
((\mathrm{S}\,\oplus) \circ \lambda_{\mathbb{R}^\omega} \circ \mathrm{T}\langle\mathtt{head}, \mathtt{tail}\rangle)(\sigma, \tau) &= ((\mathrm{S}\,\oplus) \circ \lambda_{\mathbb{R}^\omega})(\langle\sigma_0, \sigma'\rangle, \langle\tau_0, \tau'\rangle) \\
&= (\mathrm{S}\,\oplus)(\sigma_0 + \tau_0, \langle\sigma', \tau'\rangle) \\
&= \langle\sigma_0 + \tau_0, \sigma' \oplus \tau'\rangle \\
&= \langle\mathtt{head}(\sigma \oplus \tau), \mathtt{tail}(\sigma \oplus \tau)\rangle \\
&= (\langle\mathtt{head}, \mathtt{tail}\rangle \circ \oplus)(\sigma, \tau)
\end{aligned}$$

- In the case of primitive corecursion we get as well that $\langle\Omega_{\mathrm{F}}, [\mathtt{Id}, \mathtt{Id}], \omega_{\mathrm{F}}\rangle$ is a λ-bialgebra for λ as in (2) on page 13. Since $\mathrm{T}_\lambda\langle X, \alpha\rangle = \langle X, \alpha\rangle + \langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$, the coiterative arrow from it to $\langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ is given by $[h_X, \mathtt{Id}]$ where h_X is the coiterative arrows from $\langle X, \alpha\rangle$ to $\langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$, which is the identity as well in case $\langle X, \alpha\rangle = \langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$.

3.5 The λ-Coiteration Definition Schema

We are now ready to define our new schema called λ-*coiteration* for a distributive law λ of a functor T over F. It is a generalisation of the standard coiteration schema capturing other known extensions, like the corecursion schema from Theorem 3.1.

Definition 3.8 (λ-Coiterative Arrow) *Assume we are given a functor* F *with a final coalgebra* $\langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ *and a distributive law* λ *of another functor* T *over* F*. For an* FT*-coalgebra* $\langle X, \phi\rangle$ *we will call an arrow* $f : X \to \Omega_{\mathrm{F}}$ *a* λ-coiterative arrow induced by ϕ *if it is a homomorphisms up-to-*β_{F} *from* $\langle X, \phi\rangle$ *to* $\langle\Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ *for the unique arrow* β_{F} *such that* $\langle\Omega_{\mathrm{F}}, \beta_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ *is a* λ*-bialgebra (c.f. remark following Definition 3.7):*

$$\begin{tikzcd}
 & \mathrm{T}\,\Omega_{\mathrm{F}} \arrow[d, "\beta_{\mathrm{F}}"] \\
X \arrow[r, "f"] \arrow[d, "\phi"'] & \Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}\mathrm{T}X \arrow[r, "\mathrm{F}\,f|^{\beta_{\mathrm{F}}}"'] & \mathrm{F}\,\Omega_{\mathrm{F}}
\end{tikzcd}$$

The aim now is to prove the unique existence of a λ-coiterative arrow for any given $\mathrm{F}\,\mathrm{T}$-coalgebra $\langle X, \phi\rangle$. We have not found a way to do so without additional assumptions that allow us to construct an intermediate F-coalgebra and to establish a correspondence between the coiterative arrow from this coalgebra to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ and a λ-coiterative arrow induced by ϕ.

In a first approach, the carrier of this intermediate coalgebra is a countable coproduct. For this to exist, we need the additional assumption that the category C we are working in has countable coproducts, as it is the case for $\mathsf{C} = \mathsf{Set}$:

Theorem 3.9 (λ-Coiteration (1)) *Assume the category C has countable coproducts and we are given a functor $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ with a final coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ and another functor $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ that distributes over F via λ. Then, for every $\mathrm{F}\,\mathrm{T}$-coalgebra $\langle X, \phi\rangle$ there exists a unique λ-coiterative arrow induced by ϕ.*

In a second version, the carrier of the intermediate F-coalgebra is $\mathrm{T}X$. To construct the operation on it, we need a natural transformation $\mu : \mathrm{T}^2 \Rightarrow \mathrm{T}$. To relate coiterative and λ-coiterative arrows we furthermore use a natural transformation $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$. The construction can be shown to work in case some assumptions on the interaction of the three natural transformations η, μ, and λ hold, which say that $\langle \mathrm{T}, \eta, \mu\rangle$ is a monad and λ is a distributive law of this monad over F:

Theorem 3.10 (λ-Coiteration (2)) *Assume we are given a functor $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ with a final coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$, a monad $\langle \mathrm{T}, \eta, \mu\rangle$ in C and a distributive law λ of this monad over F, i.e. a distributive law of T over F such that the following diagrams commute:*

$$\begin{tikzcd}
& \mathrm{F} \arrow[dl, Rightarrow, "\eta\mathrm{F}"'] \arrow[dr, Rightarrow, "\mathrm{F}\eta"] & \\
\mathrm{T}\,\mathrm{F} \arrow[rr, Rightarrow, "\lambda"'] & & \mathrm{F}\,\mathrm{T}
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathrm{T}\,\mathrm{F} \arrow[rr, Rightarrow, "\lambda"] & & \mathrm{F}\,\mathrm{T} & \\
\mathrm{T}^2\mathrm{F} \arrow[ur, Rightarrow, "\mu\mathrm{F}"] \arrow[drr, Rightarrow, "\mathrm{T}\lambda"'] & & & & \mathrm{F}\,\mathrm{T}^2 \arrow[ul, Rightarrow, "\mathrm{F}\mu"'] \\
& & \mathrm{T}\,\mathrm{F}\,\mathrm{T} \arrow[urr, Rightarrow, "\lambda\mathrm{T}"'] & &
\end{tikzcd}$$

Then, for every $\mathrm{F}\,\mathrm{T}$-coalgebra $\langle X, \phi\rangle$ there exists a unique λ-coiterative arrow induced by ϕ.

An interesting property of the instances of the λ-coiteration schema fitting in the latter theorem is that they individually generalise the coiteration schema because one can show that a coiterative arrow induced by some F-coalgebra operation α on X is a λ-coiterative arrow induced by $\mathrm{F}\eta_X \circ \alpha$.

Another trivial observation in this direction is that the coiteration schema itself arises as the λ-coiteration schema for the identity functor distributing over F via the natural transformation $\mathrm{F}\,\mathtt{Id} : \mathrm{Id}\mathrm{F} \Rightarrow \mathrm{F}\mathrm{Id}$.

3.6 Proof of the λ-Coiteration Theorem (1)

Since a proof of the second version of the theorem requires some more technical overhead, we will leave it out and only give a proof for the first one, namely Theorem 3.9, in this section. For now, we consider its assumptions to hold, including the existence of countable coproducts in the category C. We will denote such a coproduct of the objects X_i for $i \in I\!N$ by $\sum_{i=0}^{\infty} X_i$ with the canonical injections $\mathtt{in}_j : X_j \to \sum_{i=0}^{\infty} X_i$ for $j \in I\!N$. Given another object Y and arrows $f_i : X_i \to Y$ for $i \in I\!N$, we will write $[f_i]_{i=0}^{\infty} : \sum_{i=0}^{\infty} X_i \to Y$ to denote the countable case analysis, i.e. the unique arrow satisfying $[f_i]_{i=0}^{\infty} \circ \mathtt{in}_j = f_j$ as given by the universal property of the countable coproduct. That is, we get the following picture:

$$\begin{tikzcd}
 & X_0 \arrow[dl, "\mathtt{in}_0"'] \arrow[ddl, "f_0"] & X_1 \arrow[dll, "\mathtt{in}_1"'] \arrow[ddll, "f_1"] & X_2 \arrow[dlll, "\mathtt{in}_2"'] \arrow[ddlll, "f_2"] & \cdots \\
\sum_{i=0}^{\infty} X_i \arrow[d, dashed, "{[f_i]_{i=0}^{\infty}}"'] & & & & \\
Y & & & &
\end{tikzcd}$$

To construct an F-coalgebra from $\phi : X \to \mathrm{F}\mathrm{T}X$, imagine we start with X as the carrier and ϕ as the operation. But this operation would not be of the right type, since after making an F-step we end up in $\mathrm{T}X$ rather than in X again. As a remedy, we just add $\mathrm{T}X$ to the carrier and consider $X + \mathrm{T}X$. But now we need to extend the operation as well and assign observations to $\mathrm{T}X$. A suitable candidate for this is [1]

$$lift_{\phi}^{\lambda} := \lambda_{\mathrm{T}X} \circ \mathrm{T}\phi : \ \mathrm{T}X \to \mathrm{F}\mathrm{T}^2 X.$$

Then the procedure needs to be repeated for $\mathrm{T}^2 X$, and so on. The idea can be put to work if we jump to the limit of this process and take the F-coalgebra $\langle L_X, \alpha_\phi \rangle$ with

$$\begin{aligned} L_X &:= \sum_{i=0}^{\infty} \mathrm{T}^{\mathrm{i}} X \\ \alpha_\phi &:= [\mathrm{F}\,\mathtt{in}_{i+1} \circ \phi_i]_{i=0}^{\infty} \end{aligned}$$

where the $\phi_i : \mathrm{T}^{\mathrm{i}} X \to \mathrm{F}\mathrm{T}^{\mathrm{i}+1} X$ are set inductively as $\phi_0 := \phi$ and $\phi_{i+1} := lift_{\phi_i}^{\lambda}$:

$$\begin{tikzcd}
L_X \arrow[dd, "\alpha_\phi"'] & = & X \arrow[d, "\phi_0"'] & + & \mathrm{T}X \arrow[d, "\phi_1"'] & + & \mathrm{T}^2 X \arrow[d, "\phi_2"'] & + & \cdots \\
 & & \mathrm{F}\mathrm{T}X \arrow[drr, "\mathrm{F}\,\mathtt{in}_1"'] & & \mathrm{F}\mathrm{T}^2 X \arrow[drr, "\mathrm{F}\,\mathtt{in}_2"'] & & \mathrm{F}\mathrm{T}^3 X \arrow[drr, "\mathrm{F}\,\mathtt{in}_2"'] & & \cdots \\
\mathrm{F}L_X & = & \mathrm{F}(X & + & \mathrm{T}X & + & \mathrm{T}^2 X & + & \ldots)
\end{tikzcd}$$

Since we will need the following statement about this F-coalgebra several times, we make it into a lemma. It directly follows from the definition:

Lemma 3.11 *Let $\langle X, \phi \rangle$ be an $\mathrm{F}\mathrm{T}$-coalgebra and $\langle Y, \alpha \rangle$ an F-coalgebra. With the definition of L_X and α_ϕ from above we have that $h : L_X \to Y$ is a homomorphisms from $\langle L_X, \alpha_\phi \rangle$ to $\langle Y, \alpha \rangle$ if and only if for each $i \in I\!N$ the following diagram commutes, where we set $h_j := h \circ \mathtt{in}_j$ (note that this gives $h = [h_j]_{j=0}^{\infty}$):*

$$\begin{tikzcd}
\mathrm{T}^{\mathrm{i}} X \arrow[r, "h_i"] \arrow[d, "\phi_i"'] & Y \arrow[d, "\alpha"] \\
\mathrm{F}\mathrm{T}^{\mathrm{i}+1} X \arrow[r, "\mathrm{F}\,h_{i+1}"'] & \mathrm{F}\Omega_{\mathrm{F}}
\end{tikzcd}$$

Let $h : \langle L_X, \alpha_\phi \rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ be the coiterative morphism induced by α_ϕ. In the following we will show that $h_0 : X \to \Omega_{\mathrm{F}}$ as in Lemma 3.11 fits as a unique λ-coiterative morphism induced by ϕ. From Lemma 3.11 and $i = 0$ we get that h_0 fits as a λ-coiterative morphism in case we can show that $h_1 = h_0|^{\beta_{\mathrm{F}}}$:

[1] Note that this in not precisely the same as in Definition 3.5, because ϕ is of a different shape.

$$\begin{tikzcd}
 & \mathrm{T}\,\Omega_{\mathrm{F}} \arrow[d, "\beta_{\mathrm{F}}"] \\
X \arrow[r, "h_0"] \arrow[d, "\phi_0=\phi"'] & \Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}\,\mathrm{T}\,X \arrow[r, "\mathrm{F}\,h_1 \overset{!}{=} \mathrm{F}\,h_0|^{\beta_{\mathrm{F}}}"'] & \mathrm{F}\,\Omega_{\mathrm{F}}
\end{tikzcd}$$

This follows from the next lemma:

Lemma 3.12 *Let* $h = [h_i]_{i=0}^{\infty}$ *be the coiterative morphism from the* F*-coalgebra* $\langle L_X, \alpha_\phi\rangle$ *(constructed as above from the* FT*-coalgebra* $\langle X, \phi\rangle$*) to the final* F*-coalgebra* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$*, then we have*

$$h_{i+1} = h_i|^{\beta_{\mathrm{F}}} \quad (:= \beta_{\mathrm{F}} \circ \mathrm{T}\,h_i)$$

for all $i \in I\!N$.

Proof: The statement follows from the fact that the following diagram commutes for every $i \in I\!N$:

$$\begin{tikzcd}
 & L_{\mathrm{T}\,X} \arrow[r, "{[\mathrm{T}\,h_j]_{j=0}^{\infty}}"'] \arrow[dd, "{[\mathtt{in}_{j+1}]_{j=0}^{\infty}}"] & \mathrm{T}\,\Omega_{\mathrm{F}} \arrow[dd, "\beta_{\mathrm{F}}"] \\
\mathrm{T}^{\mathrm{i+1}}\,X \arrow[ur, "\mathtt{in}_i"'] \arrow[dr, "\mathtt{in}_{i+1}"] \arrow[urr, dashed, bend left, "\mathrm{T}\,h_i"] \arrow[drr, dashed, bend right, "h_{i+1}"'] & & \\
 & L_X \arrow[r, "{h=[h_j]_{j=0}^{\infty}}"'] & \Omega_{\mathrm{F}}
\end{tikzcd}$$

The triangle commutes by the characterisation of the countable case analysis. The rectangle commutes because it is the image under the forgetful functor U_{F} of the following diagram in $\mathsf{Coalg}_{\mathrm{F}}$, commuting by the finality of $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$:

$$\begin{tikzcd}
\langle L_{\mathrm{T}\,X}, \alpha'_\phi\rangle \arrow[r, "{[\mathrm{T}\,h_j]_{j=0}^{\infty}}"] \arrow[d, "{[\mathtt{in}_{j+1}]_{j=0}^{\infty}}"'] \arrow[dr, phantom, "\text{finality}"] & \mathrm{T}_\lambda\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle \arrow[d, "\beta_{\mathrm{F}}"] \\
\langle L_X, \alpha_\phi\rangle \arrow[r, "h"'] & \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle
\end{tikzcd}$$

where $\langle L_{\mathrm{T}\,X}, \alpha'_\phi\rangle$ is just $\langle L_X, \alpha_\phi\rangle$ with the summand X missing, i.e. $\alpha'_\phi := [\mathrm{F}\,\mathtt{in}_{j+1} \circ \phi_{j+1}]_{j=0}^{\infty}$. All arrows in the above diagram are homomorphisms between the corresponding F-coalgebras indeed: h and β_{F} are so by assumption. For $[\mathtt{in}_{j+1}]_{j=0}^{\infty}$ we easily compute that the two outer paths in the following diagram equal the diagonal arrow:

$$\begin{tikzcd}
L_{\mathrm{T}\,X} \arrow[r, "{[\mathtt{in}_{j+1}]_{j=0}^{\infty}}"] \arrow[d, "\alpha'_\phi"'] \arrow[dr, dashed, "{[\mathrm{F}\,\mathtt{in}_{j+2} \circ \phi_{j+1}]_{j=0}^{\infty}}" description] & L_X \arrow[d, "\alpha_\phi"] \\
\mathrm{F}\,L_{\mathrm{T}\,X} \arrow[r, "{\mathrm{F}\,[\mathtt{in}_{j+1}]_{j=0}^{\infty}}"'] & \mathrm{F}\,L_X
\end{tikzcd}$$

For $[\mathrm{T}h_j]_{j=0}^{\infty}$ we need to show that the following diagram commutes:

$$\begin{tikzcd}
L_{\mathrm{T}X} \arrow[r, "{[\mathrm{T}h_j]_{j=0}^{\infty}}"] \arrow[d, "\alpha'_\phi"'] & \mathrm{T}\Omega_{\mathrm{F}} \arrow[d, "lift^{\lambda}_{\omega_{\mathrm{F}}}"] \\
\mathrm{F}L_{\mathrm{T}X} \arrow[r, "{\mathrm{F}[\mathrm{T}h_j]_{j=0}^{\infty}}"'] & \mathrm{F}\mathrm{T}\Omega_{\mathrm{F}}
\end{tikzcd}$$

By case analysis on $L_{\mathrm{T}X}$ and simplification this boils down to showing that for each $j \in I\!N$ the outer part of the diagram below commutes, where we abbreviated $\mathrm{T}^{\mathrm{j}}X$ to Y:

$$\begin{tikzcd}
\mathrm{T}Y \arrow[r, "\mathrm{T}h_j"] \arrow[d, "\mathrm{T}\phi_j"] \arrow[dd, bend right=60, "\phi_{j+1}"'] \arrow[dr, phantom, "\mathrm{T}(\text{ass.}h)"] & \mathrm{T}\Omega_{\mathrm{F}} \arrow[d, "\mathrm{T}\omega_{\mathrm{F}}"'] \arrow[dd, bend left=60, "lift^{\lambda}_{\omega_{\mathrm{F}}}"] \\
\mathrm{T}\mathrm{F}\mathrm{T}Y \arrow[r, "\mathrm{T}\mathrm{F}h_{j+1}"] \arrow[d, "\lambda_{\mathrm{T}Y}"] \arrow[dr, phantom, "\text{nat. }\lambda"] & \mathrm{T}\mathrm{F}\Omega_{\mathrm{F}} \arrow[d, "\lambda_{\Omega_{\mathrm{F}}}"'] \\
\mathrm{F}\mathrm{T}^2Y \arrow[r, "\mathrm{F}\mathrm{T}h_{j+1}"'] & \mathrm{F}\mathrm{T}\Omega_{\mathrm{F}}
\end{tikzcd}$$

The lower one of the two rectangles inscribed commutes by the naturality of λ, the upper one is the T-image of one following from the assumption on h by Lemma 3.11. □

This concludes the first part of the argument, namely the construction of the λ-coiterative morphism h_0 from the coiterative one h. It remains to be shown that h_0 is the unique arrow with this property. We do so by showing that if $f : X \to \Omega_{\mathrm{F}}$ fits as a λ-coiterative morphisms induced by ϕ, then we can extend it to a homomorphism $h' : \langle L_X, \alpha_\phi \rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ (with $f = h' \circ \mathtt{in}_0$). By finality we then get $h' = h$ and thus $f = h_0 = h \circ \mathtt{in}_0$.

The main part of the argument is stated in the following lemma in a slightly more general setting, because we will need it in this format later:

Lemma 3.13 *Let $\langle X, \phi \rangle$ be an* FT*-coalgebra and $\langle Z, \beta, \alpha \rangle$ be a λ-bialgebra. If $f : X \to Z$ is a homomorphism up-to-β from $\langle X, \phi \rangle$ to $\langle Z, \alpha \rangle$, then $[f_i]_{i=0}^{\infty} : L_X \to Z$ with $f_i : \mathrm{T}^{\mathrm{i}}X \to Z$ set inductively as $f_0 := f$ and $f_{i+1} := f_i|^{\beta}$ is a homomorphism from $\langle L_X, \alpha_\phi \rangle$ to $\langle Z, \alpha \rangle$.*

Proof: According to Lemma 3.11 what we need to check is that for all $i \in I\!N$ the diagram below commutes, which is in this case equivalent to saying that f_i is a homomorphism up-to-β from $\langle \mathrm{T}^{\mathrm{i}}X, \phi_i \rangle$ to $\langle Z, \alpha \rangle$ for all $i \in I\!N$ (abbreviating again $\mathrm{T}^{\mathrm{i}}X$ to Y):

$$\begin{tikzcd}
 & \mathrm{T}Z \arrow[d, "\beta"] \\
Y \arrow[r, "f_i"] \arrow[d, "\phi_i"'] & Z \arrow[d, "\alpha"] \\
\mathrm{F}\mathrm{T}Y \arrow[r, "\mathrm{F}f_{i+1}=\mathrm{F}f_i|^{\beta_{\mathrm{F}}}"'] & \mathrm{F}Z
\end{tikzcd}$$

This can be obtained by induction on i: For $i = 0$ we encounter the assumption on $f = f_0$. For $i + 1$ we have the diagram below:

$$\begin{tikzcd}
 & & \mathrm{T}Z \arrow[d, "\beta"] \\
\mathrm{T}Y \arrow[r, "\mathrm{T}f_i"'] \arrow[d, "\mathrm{T}\phi_i"] \arrow[rr, dashed, bend left, "f_{i+1}=f_i|^{\beta}"] \arrow[dd, dashed, bend right, "\phi_{i+1}"'] \arrow[dr, phantom, "\mathrm{T}(\text{ind. hyp.})"] & \mathrm{T}Z \arrow[r, "\beta"'] \arrow[d, "\mathrm{T}\alpha"] \arrow[dr, phantom, "\lambda\text{-bialg}"] & Z \arrow[dd, "\alpha"] \\
\mathrm{TFT}Y \arrow[r, "\mathrm{TF}f_i|^{\beta}", "=\mathrm{TF}f_{i+1}"'] \arrow[d, "\lambda_{\mathrm{T}Y}"] \arrow[dr, phantom, "\text{nat.}\,\lambda"] & \mathrm{TF}Z \arrow[d, "\lambda_Z"] & \\
\mathrm{FT}^2Y \arrow[r, "\mathrm{FT}f_{i+1}"] \arrow[rr, dashed, bend right, "\mathrm{F}f_{i+1}|^{\beta}"'] & \mathrm{FT}Z \arrow[r, "\mathrm{F}\beta"] & \mathrm{F}Z
\end{tikzcd}$$

□

4 Proof by λ-Coinduction

In this Section we show how the setting from above can be used to derive generalised proof principles for behavioural equivalence. First we recall the proof technique based on the categorical definition of a *bisimulation*. Next we give examples of proof goals for which the method is difficult to apply because relations covering them become unpleasantly complex while trying to make them satisfy the full definition of a bisimulation. This observation motivates us to look for weaker sufficient conditions for a relation to only contain bisimilar states. We introduce the notion of a λ-*bisimulation* for a distributive law λ and demonstrate that the corresponding proof principle helps to overcome the problems indicated by the given examples.

4.1 Bisimulation Proofs

Accepting the carrier of the final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ as a domain for behaviours of the appropriate type, for two F-coalgebras $\langle X, \alpha_X \rangle$ and $\langle Y, \alpha_Y \rangle$, to show that two states $\langle x, y \rangle \in X \times Y$ behave the same way, we need to show that x and y are mapped onto the same element of Ω_{F} by the respective coiterative morphisms $h_X : \langle X, \alpha_X \rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ and $h_Y : \langle Y, \alpha_Y \rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$, i.e. $h_X(x) = h_Y(y)$.

One technique to prove these equations is to come up with a relation $R \subseteq X \times Y$ containing $\langle x, y \rangle$ – or rather $\langle x_i, y_i \rangle$ for $i \in I$ since one is often interested in proving equivalence of behaviour for more pairs in one go – and on which there exists an F-coalgebra operation $\gamma : R \to \mathrm{F}R$ embodying the common behaviour of the states related. Technically one requires the two projections $\pi_1 : R \to X$ and $\pi_2 : R \to Y$ to form homomorphisms from $\langle R, \gamma \rangle$ to $\langle X, \alpha_X \rangle$ and $\langle Y, \alpha_Y \rangle$ respectively. This principle works, because it gives us the arrows shown in the following diagram in $\mathsf{Coalg}_{\mathrm{F}}$, commuting by finality:

$$\begin{tikzcd}
 & \langle R, \gamma \rangle \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] & \\
\langle X, \alpha_X \rangle \arrow[dr, "h_X"'] & & \langle Y, \alpha_Y \rangle \arrow[dl, "h_Y"] \\
 & \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle &
\end{tikzcd}$$

Hence we get $h_X(x_i) = h_X(\pi_1(\langle x_i, y_i \rangle)) = h_Y(\pi_2(\langle x_i, y_i \rangle)) = h_Y(y_i)$ for all $i \in I$ as wanted.

A relation R with the above property that there is an F-operation $\gamma : R \to \mathrm{F}R$ turning the two projections into homomorphisms to the given coalgebras is called an F-*bisimulation*. This notion was introduced by Aczel and Mendler [AM89] as a generalisation of notions of bisimulation in use for concrete systems, like strong bisimulation in process algebra. It became a central ingredient of the theory of coalgebra (see again e.g. Jacobs and Rutten [JR96, Rut00b]).

Working in an abstract category C instead of Set, it turns out that it is easier to deal with a slightly more general definition where one considers *spans* instead of relations:

Definition 4.1 (span) *In a category C, a* span $\mathcal{R} = \langle R, r_1, r_2\rangle$ *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 the following diagram commutes:

$$\begin{tikzcd}
& R \arrow[dl, "r_1"'] \arrow[dr, "r_2"] \arrow[dd, dashed, "\exists f"] & \\
X & & Y \\
& S \arrow[ul, "s_1"] \arrow[ur, "s_2"'] &
\end{tikzcd}$$

Example 4.2 *In Set, we can view every relation $R \subseteq X \times Y$ as the span $\langle R, \pi_1, \pi_2\rangle$. On the other hand, for every span $\mathcal{R} = \langle R, r_1, r_2\rangle$ between X and Y we consider its* image $\langle r_1, r_2\rangle[R] = \{\langle r_1(z), r_2(z)\rangle \mid z \in R\} \subseteq X \times Y$. *We get $\langle R, r_1, r_2\rangle \preceq \langle S, s_1, s_2\rangle$ if and only if $\langle r_1, r_2\rangle[R] \subseteq \langle s_1, s_2\rangle[S]$.*

Definition 4.3 (Bisimulation) *For a functor $\mathrm{F} : \mathsf{C} \to \mathsf{C}$, an* F-bisimulation *between two F-coalgebras $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$ is a span $\mathcal{B} = \langle B, b_1, b_2\rangle$ between the carriers X and Y, such that there is an F-operation $\gamma : B \to \mathrm{F}B$ turning b_1 and b_2 into homomorphisms:*

$$\begin{tikzcd}
X \arrow[d, "\alpha_X"'] & B \arrow[l, "b_1"'] \arrow[r, "b_2"] \arrow[d, dashed, "\exists\gamma"] & Y \arrow[d, "\alpha_Y"] \\
\mathrm{F}X & \mathrm{F}B \arrow[l, "\mathrm{F}\,b_1"] \arrow[r, "\mathrm{F}\,b_2"'] & \mathrm{F}Y
\end{tikzcd}$$

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

With other words, a bisimulation is a span $\langle B, b_1, b_2\rangle$ between the carriers of two coalgebras such that there is an operation γ that turns it into the span $\langle\langle B, \gamma\rangle, b_1, b_2\rangle$ in $\mathsf{Coalg}_{\mathrm{F}}$ between the coalgebras themselves. We will use the latter notation if we want to specify the operation γ involved.

In Set we call two states x and y of two coalgebras *bisimilar* if there is a bisimulation $\langle B, b_1, b_2\rangle$ relating the two, i.e. if there is a $z \in B$ such that $x = b_1(z)$ and $y = b_2(z)$.

Note that this definition of a bisimulation, which also appears e.g. in the work of Turi and Plotkin [TP97] and Lenisa [Len99b], generalises the one by Aczel and Mendler [AM89] for relations in Set in the following way: A relation is a bisimulation if and only if the span it gives rise to according to Example 4.2 is a bisimulation in our sense. On the other hand, if a span is a bisimulation, then its image is a relational bisimulation:

Lemma 4.4 (c.f. [Rut00b, Lemma 5.3]) *Let $\mathrm{F} : \mathsf{Set} \to \mathsf{Set}$ be a functor and $\mathcal{B} = \langle B, b_1, b_2\rangle$ be a bisimulation between the given F-coalgebras $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$. Then the image $\langle b_1, b_2\rangle[B] := \{\langle b_1(z), b_2(z)\rangle \mid z \in B\} \subseteq X \times Y$ is a (relational) bisimulation as well.*

Example 4.5 (bisimulations on streams) *In the case of the stream systems from example 2.2, the condition for a bisimulation relation can be spelled out like this: A relation $B \subseteq X \times Y$ gives rise to a bisimulation $\langle B, \pi_1, \pi_2\rangle$ between the stream coalgebras $\langle X, \langle o_X, s_X\rangle\rangle$ and $\langle Y, \langle o_Y, s_Y\rangle\rangle$, if and only if there are $o_B : B \to \mathbb{R}$ and $s_B : B \to B$ making the diagram below commute, which is equivalent to saying that for all $\langle x, y\rangle \in B$ we have $o_X(x) = o_Y(y)$ and $\langle s_X(x), s_Y(y)\rangle \in B$:*

$$\begin{tikzcd}
X \arrow[d, "\langle o_X, s_X \rangle"'] & B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "\exists \langle o_B, s_B \rangle"] & Y \arrow[d, "\langle o_Y, s_Y \rangle"] \\
\mathbb{R} \times X & \mathbb{R} \times B \arrow[l, "\mathtt{Id} \times \pi_1"] \arrow[r, "\mathtt{Id} \times \pi_2"'] & \mathbb{R} \times Y
\end{tikzcd}$$

The abovementioned proof principle can now be formulated as follows:

Theorem 4.6 *Given a functor* F *with a final coalgebra* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$*. If* $\langle B, b_1, b_2 \rangle$ *is a bisimulation between the* F*-coalgebras* $\langle X, \alpha_X \rangle$ *and* $\langle Y, \alpha_Y \rangle$*, then we have the following order of spans on* Ω_{F}*:*

$$\langle B, h_X \circ b_1, h_Y \circ b_2 \rangle \preceq \langle \Omega_{\mathrm{F}}, \mathtt{Id}, \mathtt{Id} \rangle,$$

where $h_X : \langle X, \alpha_X \rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ *and* $h_Y : \langle Y, \alpha_Y \rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ *are the coiterative morphisms.*

Proof: Let γ be an F-operation on B witnessing the bisimulation property. Then the coiterative morphism $h_B : \langle B, \gamma \rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ is a witness for the order of spans on Ω_{F} claimed. The equations from the definition of the ordering (Definition 4.1) to be shown easily follow by applying the forgetful functor U_{F} to the diagram below. Both parts of it commute by finality:

$$\begin{tikzcd}
\langle X, \alpha_X \rangle \arrow[d, "h_X"'] & \langle B, \gamma \rangle \arrow[l, "b_1"'] \arrow[r, "b_2"] \arrow[d, "h_B"] & \langle Y, \alpha_Y \rangle \arrow[d, "h_Y"] \\
\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle & \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle \arrow[l, "\mathtt{Id}"] \arrow[r, "\mathtt{Id}"'] & \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle
\end{tikzcd}$$

□

Example 4.7 *Via the translation between spans and relations in* Set *from Example 4.2, we have that* $\langle \Omega_{\mathrm{F}}, \mathtt{Id}, \mathtt{Id} \rangle$ *represents the diagonal relation* $\triangle_{\Omega_{\mathrm{F}}} := \{\langle \sigma, \sigma \rangle \mid \sigma \in \Omega_{\mathrm{F}}\}$ *and thus Theorem 4.6 says that for a bisimulation* $B \subseteq X \times Y$ *between the* F*-coalgebras* $\langle X, \alpha_X \rangle$ *and* $\langle Y, \alpha_Y \rangle$ *we have that* $\langle x, y \rangle \in B$ *implies* $h_X(x) = h_Y(y)$ *for the coiterative morphisms* h_X *and* h_Y*.*

In the case of a bisimulation B *on the final coalgebra* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ *we have that the coiterative morphism is the identity on* Ω_{F}*, and we get that* $\langle x, y \rangle \in B$ *implies* $x = y$*.*

Example 4.8 *We want to show that the operation* $\oplus$ *from Example 2.8 is commutative, i.e. that for all* $\sigma, \tau \in \mathbb{R}^\omega$ *we have* $\sigma \oplus \tau = \tau \oplus \sigma$*. According to Example 4.7, it is sufficient to show that the relation*

$$B = \{\langle \sigma \oplus \tau, \tau \oplus \sigma \rangle \mid \sigma, \tau \in \mathbb{R}^\omega\}$$

is a bisimulation on $\langle \mathbb{R}^\omega, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$*. This follows by Example 4.5, since for* $\sigma = \langle \sigma_0 : \sigma' \rangle, \tau = \langle \tau_0 : \tau' \rangle \in \mathbb{R}^\omega$ *we have*

$$\mathtt{head}(\sigma \oplus \tau) = \sigma_0 + \tau_0 = \tau_0 + \sigma_0 = \mathtt{head}(\tau \oplus \sigma)$$

and

$$\langle \mathtt{tail}(\sigma \oplus \tau), \mathtt{tail}(\tau \oplus \sigma) \rangle = \langle \sigma' \oplus \tau', \tau' \oplus \sigma' \rangle \in B.$$

We now give two examples where a technique that requires (standard) bisimulations does not seem to be ideal.

4.2 Example 1: Distributivity of $\oplus$ over $\otimes$

We would like to prove that the addition of streams of real numbers defined in the Example 2.8 distributes over the multiplication specified in Section 3.1:

$$\forall \sigma, \tau, \rho \in \mathbb{R}^\omega : \quad \sigma \otimes (\tau \oplus \rho) = (\sigma \otimes \tau) \oplus (\sigma \otimes \rho)$$

To obtain a bisimulation, the starting point would be to consider all the pairs needed for the statement:

$$B := \{\langle \sigma \otimes (\tau \oplus \rho), (\sigma \otimes \tau) \oplus (\sigma \otimes \rho)\rangle \mid \sigma, \tau, \rho \in I\!R^\omega\} \subseteq I\!R^\omega \times I\!R^\omega \tag{3}$$

For this relation to be contained in a bisimulation, we first need to check that all streams related have equal heads. This is the case, since for all $\sigma, \tau, \rho \in I\!R^\omega$ we have (writing again $\sigma = \langle \sigma_0 : \sigma' \rangle$ and similar for τ and ρ)

$$\begin{aligned} \mathtt{head}(\sigma \otimes (\tau \oplus \rho)) &= \sigma_0 \cdot \mathtt{head}(\tau \oplus \rho) \\ &= \sigma_0 \cdot (\tau_0 + \rho_0) \\ &= \sigma_0 \cdot \tau_0 + \sigma_0 \cdot \rho_0 \\ &= \mathtt{head}(\sigma \otimes \tau) + \mathtt{head}(\sigma \otimes \rho) \\ &= \mathtt{head}((\sigma \otimes \tau) \oplus (\sigma \otimes \rho)). \end{aligned}$$

Furthermore, the bisimulation to be found has to relate the tails of every two streams related by B:

$$\begin{aligned} &\mathtt{tail}(\sigma \otimes (\tau \oplus \rho)) \\ &\quad = \{\text{def. } \otimes\} \\ &\qquad ([\sigma_0] \otimes \mathtt{tail}(\tau \oplus \rho)) \oplus (\sigma' \otimes (\tau \oplus \rho)) \\ &\quad = \{\text{def. } \oplus\} \\ &\qquad \underbrace{([\sigma_0] \otimes (\tau' \oplus \rho'))}_{=:x_1} \oplus \underbrace{(\sigma' \otimes (\tau \oplus \rho))}_{=:x_2} \end{aligned}$$

and

$$\begin{aligned} &\mathtt{tail}((\sigma \otimes \tau) \oplus (\sigma \otimes \rho)) \\ &\quad = \{\text{def. } \oplus\} \\ &\qquad \mathtt{tail}(\sigma \otimes \tau) \oplus \mathtt{tail}(\sigma \otimes \rho) \\ &\quad = \{\text{def. } \otimes\} \\ &\qquad (([\sigma_0] \otimes \tau') \oplus (\sigma' \otimes \tau)) \oplus (([\sigma_0] \otimes \rho') \oplus (\sigma' \otimes \rho)) \\ &\quad = \{\text{associativity and commutativity of } \oplus\} \\ &\qquad \underbrace{(([\sigma_0] \otimes \tau') \oplus ([\sigma_0] \otimes \rho'))}_{=:y_1} \oplus \underbrace{((\sigma' \otimes \tau) \oplus (\sigma' \otimes \rho))}_{=:y_2} \end{aligned}$$

We observe that we obtain sums of streams related by B, namely $\langle x_1, y_1\rangle, \langle x_2, y_2\rangle \in B$. Thus, the bisimulation should also contain $T(B)$ where T acts on relations $R \subseteq I\!R^\omega \times I\!R^\omega$ as

$$T(R) := \{\langle x_1 \oplus x_2, y_1 \oplus y_2\rangle \mid \langle x_1, y_1\rangle, \langle x_2, y_2\rangle \in R\} \subseteq I\!R^\omega \times I\!R^\omega.$$

But if this is the case, we need to check on the pairs contained in $T(B)$ as well. The continuation of this procedure would lead to the relation

$$\tilde{B} := \bigcup_{i=0}^{\infty} T^i(B).$$

To show that it is a bisimulation one could prove by induction on $i \in I\!N$ that for $\langle x, y\rangle \in T^i(B)$ one finds

$$\mathtt{head}(x) = \mathtt{head}(y) \quad \text{and} \quad \langle \mathtt{tail}(x), \mathtt{tail}(y)\rangle \in T^{i+1}(B).$$

The base case is given by the calculation from above, the induction step is easy and independent of B.[2] It proves the following principle: Given a relation $B \subseteq I\!R^\omega \times I\!R^\omega$ such that for $\langle x, y\rangle \in B$ one has

$$\mathtt{head}(x) = \mathtt{head}(y) \quad \text{and} \quad \langle \mathtt{tail}(x), \mathtt{tail}(y)\rangle \in T(B)$$

[2] The induction step basically states that T is a *respectful function* in the terminology of Sangiorgi [San98] (with a translation of this notion from labeled transition systems to stream systems).

then $\tilde{B} := \bigcup_{i=0}^{\infty} T^i(B)$ is a bisimulation extending B.

With this principle, one can work directly with the relation B as in

4.3 Example 2: Sorted Insertion

In the context of Section 3.2 consider another function $\texttt{insert}_l : \mathbb{R}^* \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ that inserts several real numbers given as a list $l \in \mathbb{R}^*$ into a stream $\sigma = \langle \sigma_0 : \sigma' \rangle \in \mathbb{R}^\omega$ in one go, specified by the following coiterative definition:

$$\langle \texttt{head}, \texttt{tail} \rangle (\texttt{insert}_l(l, \sigma)) = \begin{cases} \langle m_l, \texttt{insert}_l(l - m_l, \sigma) \rangle & \text{if } l \neq \langle\rangle \text{ and } m_l \leq \sigma_0, \\ \langle \sigma_0, \texttt{insert}_l(l, \sigma') \rangle & \text{otherwise,} \end{cases}$$

where $m_l \in \mathbb{R}$ is the least element of a nonempty list l and $l - m_l$ denotes the list l with the first occurrence of m_l deleted.

Now we want to prove that we can use this function to implement multiple applications of the previous insertion function that inserted one number only, i.e.

$$\begin{aligned} \texttt{insert}_l(\langle\rangle, \sigma) &= \sigma, \\ \texttt{insert}_l(\langle r : l \rangle, \sigma) &= \texttt{insert}(r, \texttt{insert}_l(l, \sigma)). \end{aligned}$$

The first equation is easily proved using the bisimulation

$$\{ \langle \texttt{insert}_l(\langle\rangle, \sigma), \sigma \rangle \mid \sigma \in \mathbb{R}^\omega \},$$

for the second one the relation

$$B = \{ \langle \texttt{insert}_l(\langle r : l \rangle, \sigma), \texttt{insert}(r, \texttt{insert}_l(l, \sigma)) \rangle \mid r \in \mathbb{R}, l \in \mathbb{R}^*, \sigma \in \mathbb{R}^\omega \}$$

alone does not do the job, because in the case $r \leq m_l \wedge r \leq \sigma_0$ we have

$$\begin{aligned} \texttt{tail}(\texttt{insert}_l(\langle r : l \rangle, \sigma)) &= \texttt{insert}_l(l, \sigma) \\ &= \texttt{tail}(\texttt{insert}(r, \texttt{insert}_l(l, \sigma))), \end{aligned}$$

but the pair $\langle \texttt{insert}_l(l, \sigma), \texttt{insert}_l(l, \sigma) \rangle$ may not be in B. As a remedy, one would take $\tilde{B} = B \cup \triangle_{\mathbb{R}^\omega}$. The pairs added via $\triangle_{\mathbb{R}^\omega}$ do not impose new proof obligations, since for $\langle \sigma, \tau \rangle \in \triangle_{\mathbb{R}^\omega}$ one trivially has

$$\texttt{head}(\sigma) = \texttt{head}(\tau) \quad \text{and} \quad \langle \texttt{tail}(\sigma), \texttt{tail}(\tau) \rangle \in \triangle_{\mathbb{R}^\omega} \subseteq \tilde{B}.$$

The underlying principle can be stated more generally using the following notion:

Definition 4.9 (Bisimulation up-to-equality) *Let* $\mathrm{F} : \mathsf{Set} \to \mathsf{Set}$ *be a functor with a final coalgebra* $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$. *A relation* $B \subseteq \Omega_\mathrm{F} \times \Omega_\mathrm{F}$ *is called a* bisimulation up-to-equality*, if there exists an operation* $\chi : B \to \mathrm{F}(B \cup \triangle_{\Omega_\mathrm{F}})$ *making the following diagram commute (note that the upper arrows use the projections* $\pi_i : B \to \Omega_\mathrm{F}$ *whereas the lower ones are* $\pi_i : B \cup \triangle_{\Omega_\mathrm{F}} \to \Omega_\mathrm{F}$*):*

$$\begin{tikzcd} \Omega_\mathrm{F} \arrow[d, "\omega_\mathrm{F}"'] & B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "\exists\chi"] & \Omega_\mathrm{F} \arrow[d, "\omega_\mathrm{F}"] \\ \mathrm{F}\,\Omega_\mathrm{F} & \mathrm{F}(B \cup \triangle_{\Omega_\mathrm{F}}) \arrow[l, "\mathrm{F}\,\pi_1"] \arrow[r, "\mathrm{F}\,\pi_2"'] & \mathrm{F}\,\Omega_\mathrm{F} \end{tikzcd}$$

For every bisimulation up-to-equality B one can easily find a larger bisimulation $\tilde{B}$ (namely $B \cup \triangle_{\Omega_\mathrm{F}}$, using the argument from above). By Example 4.7 this means that it suffices to show that B is a bisimulation up-to-equality in order to prove that all elements related are bisimilar.

4.4 λ-Coinduction

The aim of this section is to give a framework that generalises the proof principle based on standard bisimulations such that the two cases mentioned above arise as instances of it. We are going to use the same setting as for the λ-coiteration schema.

Given a functor T that distributes over F via λ, the idea is to ask for an FT-coalgebra operation χ on the relation $B \subseteq X \times Y$ instead of an F-coalgebra operation. But then we need to find a suitable notion of a behaviour preserving arrow to be imposed on the projections $\pi_1 : B \to X$ and $\pi_2 : B \to Y$, replacing the assumption of being homomorphisms in the definition of a bisimulation.

For a general consideration, assume we are given an FT-coalgebra $\langle X, \phi \rangle$, an F-coalgebra $\langle Y, \alpha \rangle$, and the final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$. The λ-coiterative morphism $h_X : X \to \Omega_{\mathrm{F}}$ induced by ϕ assigns F-behaviours to the states in X, those of the states in Y are given by the coiterative morphism $h_Y : Y \to \Omega_{\mathrm{F}}$. The function f preserves these behaviours if we have $h_X = h_Y \circ f$. To establish this equation, we could look for conditions guaranteeing that $h_Y \circ f$ fits as a λ-coiterative morphism induced by ϕ. Then the statement would follow by its uniqueness, stated in Theorem 3.9.

The following two lemmata show that $h_Y \circ f$ fits as a λ-coiterative arrow in case f is a homomorphism up-to-β as introduced in Definition 3.3 from $\langle X, \phi \rangle$ to $\langle Y, \alpha \rangle$ for a T-algebra operation β on Y such that $\langle Y, \beta, \alpha \rangle$ is a λ-bialgebra.

Lemma 4.10 *Let two functors* $\mathrm{F}, \mathrm{T} : \mathsf{C} \to \mathsf{C}$, *an* FT-*coalgebra* $\langle X, \phi \rangle$, *and two* T,F-*bialgebras* $\langle Y, \beta_Y, \alpha_Y \rangle$ *and* $\langle Z, \beta_Z, \alpha_Z \rangle$ *be given. If* $f : X \to Y$ *is a homomorphism up-to-*β_Y *from* $\langle X, \phi \rangle$ *to* $\langle Y, \alpha_Y \rangle$ *and* $h : Y \to Z$ *is a bialgebra homomorphism from* $\langle Y, \beta_Y, \alpha_Y \rangle$ *to* $\langle Z, \beta_Z, \alpha_Z \rangle$, *then* $h \circ f : X \to Z$ *is a homomorphism up-to-*β_Z *from* $\langle X, \phi \rangle$ *to* $\langle Z, \alpha_Z \rangle$.

Proof: From the assumptions we have that both inner rectangles in the diagram below commute:

$$\begin{tikzcd}
 & \mathrm{T}Y \arrow[d, dashed, "\beta_Y"'] & \mathrm{T}Z \arrow[d, dashed, "\beta_Z"] \\
X \arrow[r, "f"] \arrow[d, "\phi"'] \arrow[dr, phantom, "\text{ass. } f"] & Y \arrow[r, "h"] \arrow[d, "\alpha_Y"'] \arrow[dr, phantom, "\text{ass. } h"] & Z \arrow[d, "\alpha_Z"] \\
\mathrm{F}\mathrm{T}X \arrow[r, "\mathrm{F}\, f|^{\beta_Y}"] \arrow[rr, dashed, bend right, "\mathrm{F}\,(*)", "\mathrm{F}\,(h \circ f)|^{\beta_Z}"'] & \mathrm{F}Y \arrow[r, "\mathrm{F}\, h"] & \mathrm{F}Z
\end{tikzcd}$$

It remains to be shown that the dashed arrow at the bottom can be taken for the composition of the two above. It follows from the equation

$$h \circ f|^{\beta_Y} \overset{(*)}{=} (h \circ f)|^{\beta_Z},$$

which holds since h was assumed to be an algebra homomorphism as well:

$$\begin{tikzcd}
\mathrm{T}X \arrow[r, "\mathrm{T}\, f"] \arrow[rr, dashed, bend left, "\mathrm{T}\,(h \circ f)"] \arrow[dr, dashed, "f|^{\beta_Y}"'] & \mathrm{T}Y \arrow[r, "\mathrm{T}\, h"] \arrow[d, "\beta_Y"'] \arrow[dr, phantom, "\text{ass. } h"] & \mathrm{T}Z \arrow[d, "\beta_Z"] \\
 & Y \arrow[r, "h"'] & Z
\end{tikzcd}$$

□

Let β_{F} denote again the unique T-algebra operation such that $\langle \Omega_{\mathrm{F}}, \beta_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ is a λ-bialgebra. By definition, a homomorphism up-to-β_{F} from $\langle X, \phi \rangle$ to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ is a λ-coiterative morphism induced by ϕ. Thus, to use the above lemma, we need to guarantee that the coiterative morphism h_Y induced by α is a bialgebra homomorphism from $\langle Y, \beta, \alpha\rangle$ to $\langle \Omega_\mathrm{F}, \beta_\mathrm{F}, \omega_\mathrm{F}\rangle$. One sufficient condition is that the first bialgebra is a λ-bialgebra as well:

Lemma 4.11 *Assume a functor $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ with a final coalgebra $\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ and another functor $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ that distributes over F via λ to be given. Let $\beta_\mathrm{F} : \mathrm{T}\Omega_\mathrm{F} \to \Omega_\mathrm{F}$ denote the unique arrow such that $\langle \Omega_\mathrm{F}, \beta_\mathrm{F}, \omega_\mathrm{F}\rangle$ is a λ-bialgebra (i.e. the coiterative morphism from $\mathrm{T}_\lambda\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ to $\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$). For every λ-bialgebra $\langle Y, \beta, \alpha\rangle$ the coiterative morphism $h_Y : \langle Y, \alpha\rangle \to \langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ is at the same time a bialgebra homomorphism from $\langle Y, \alpha, \beta\rangle$ to $\langle \Omega_\mathrm{F}, \beta_\mathrm{F}, \omega_\mathrm{F}\rangle$.*

Note that this statement is the heart of the proof that β_F extends the final F-coalgebra to a final object in λ-Bialg.

Proof: We need to show that h_Y is a T-algebra homomorphism from $\langle Y, \beta\rangle$ to $\langle \Omega_\mathrm{F}, \beta_\mathrm{F}\rangle$. We can see this by applying the forgetful functor to this diagram in $\mathsf{Coalg}_\mathrm{F}$, which commutes by finality:

$$\begin{tikzcd}
\mathrm{T}_\lambda\langle Y, \alpha\rangle \arrow[r, "\mathrm{T}_\lambda h_Y"] \arrow[d, "\beta"'] \arrow[dr, phantom, "\text{finality}"] & \mathrm{T}_\lambda\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle \arrow[d, "\beta_\mathrm{F}"] \\
\langle Y, \alpha\rangle \arrow[r, "h_Y"'] & \langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle
\end{tikzcd}$$

The arrows exist in $\mathsf{Coalg}_\mathrm{F}$, i.e. they are homomorphisms for the corresponding coalgebras: β and β_F are since they extend $\langle Y, \alpha\rangle$ and $\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ respectively to λ-bialgebras (c.f. remark following Definition 3.7), h_Y is so by assumption, which further implies that $\mathrm{T}_\lambda h_Y$ exists in $\mathsf{Coalg}_\mathrm{F}$, since T_λ is a functor in it (c.f. Lemma 3.6). □

Taken together, we get the following corollary:

Corollary 4.12 *Assume a functor $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ with a final coalgebra $\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ and another functor $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ that distributes over F via λ to be given. If $\langle Y, \beta, \alpha\rangle$ is a λ-bialgebra and $f : X \to Y$ is a homomorphism up-to-β from the $\mathrm{F}\mathrm{T}$-coalgebra $\langle X, \phi\rangle$ to $\langle Y, \alpha\rangle$, then for the coiterative morphism h_Y induced by α the composition $h_Y \circ f$ is a λ-coiterative morphism induced by ϕ.*

This corollary forms the basis of our new λ-coinduction proof principle for bisimilarity, in which the notion of bisimulation is replaced by the following one:

Definition 4.13 (λ-bisimulation) *Let $\mathrm{F}, \mathrm{T} : \mathsf{C} \to \mathsf{C}$ be functors. A span $\mathcal{B} = \langle B, b_1, b_2\rangle$ is a* λ-bisimulation *between the F-coalgebras $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$, if there exist an $\mathrm{F}\mathrm{T}$-operation χ on B and T-algebra operations β_X and β_Y on X and Y, such that $\langle X, \beta_X, \alpha_X\rangle$ and $\langle Y, \beta_Y, \alpha_Y\rangle$ are λ-bialgebras and b_1 and b_2 are homomorphisms up-to-β_X and -β_Y respectively:*

$$\begin{tikzcd}
\mathrm{T}X \arrow[d, dashed, "\exists\beta_X"'] & & \mathrm{T}Y \arrow[d, dashed, "\exists\beta_Y"] \\
X \arrow[d, "\alpha_X"'] & B \arrow[l, "b_1"'] \arrow[r, "b_2"] \arrow[d, dashed, "\exists\chi"] & Y \arrow[d, "\alpha_Y"] \\
\mathrm{F}X & \mathrm{F}\mathrm{T}B \arrow[l, "\mathrm{F}\,b_1|^{\beta_X}"] \arrow[r, "\mathrm{F}\,b_2|^{\beta_Y}"'] & \mathrm{F}Y
\end{tikzcd}$$

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

As in the case of bisimulation from Definition 4.3, we sometimes want to fix the operation χ involved. We will then again use the notation $\langle\langle B, \chi\rangle, b_1, b_2\rangle$, but note that this time we do not get a span in a category of coalgebras. If we want to make the T-algebra operations β_X and β_Y explicit, we talk about a λ-bisimulation *with respect to β_X and β_Y*.

From Corollary 4.12 we can easily derive a correspondent of Theorem 4.6 for λ-bisimulation:

Theorem 4.14 (λ-coinduction) *Let the category* C *have countable coproducts. Given a functor* $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ *with a final coalgebra* $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ *and another functor* $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ *that distributes over* F *via* λ.

If $\mathcal{B} = \langle B, b_1, b_2 \rangle$ *is a* λ*-bisimulation between the* F*-coalgebras* $\langle X, \alpha_X \rangle$ *and* $\langle Y, \alpha_Y \rangle$*, then we have* $\langle B, h_X \circ b_1, h_Y \circ b_2 \rangle \preceq \langle \Omega_\mathrm{F}, \mathtt{Id}, \mathtt{Id} \rangle$*, where* $h_X : \langle X, \alpha_X \rangle \to \langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ *and* $h_Y : \langle Y, \alpha_Y \rangle \to \langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ *are the coiterative morphisms.*

Again, in the world of sets ($\mathsf{C} = \mathsf{Set}$) this means that for $B \subseteq X \times Y$ such that $\langle B, \pi_1, \pi_2 \rangle$ is a λ-bisimulation we have that $h_X(x) = h_Y(y)$ for all $\langle x, y \rangle \in B$.

Proof: Let $\chi : B \to \mathrm{FT}B$, $\beta_X : \mathrm{T}X \to X$ and $\beta_Y : \mathrm{T}Y \to Y$ witness the λ-bisimulation property for $\mathcal{B}$. Then the by Theorem 3.9 existing λ-coiterative morphism $h_B : B \to \Omega_\mathrm{F}$ induced by χ is a witness for the order of spans on Ω_F claimed:

$$\begin{tikzcd}
X \arrow[d, "h_X"'] & B \arrow[l, "b_1"'] \arrow[r, "b_2"] \arrow[d, "h_B"] & Y \arrow[d, "h_Y"] \\
\Omega_\mathrm{F} & \Omega_\mathrm{F} \arrow[l, "\mathtt{Id}"] \arrow[r, "\mathtt{Id}"'] & \Omega_\mathrm{F}
\end{tikzcd}$$

Applying Corollary 4.12 to both λ-bialgebras – $\langle X, \beta_X, \alpha_X \rangle$ and $\langle Y, \beta_Y, \alpha_Y \rangle$ – yields that $h_X \circ b_1$ and $h_Y \circ b_2$ are λ-coiterative morphisms induced by χ as well, and thus all three are equal by the uniqueness part of Theorem 3.9. □

One immediately gets a version of the above theorem with the condition from Theorem 3.9 replaced by the one from 3.10 by also replacing the one Theorem by the other inside the proof.

Instead of applying Theorem 4.14 directly to prove that the states related by a λ-bisimulation show equivalent behaviours, we could try to reduce the new concept to the standard one and still use Theorem 4.6. This requires a construction that enlarges a λ-bisimulation such that a standard bisimulation is encountered, similar to what we have done in the examples. The advantage would be that with this construction we could use λ-bisimulations instead of bisimulations in other contexts as well. These include methods that construct larger bisimulations from smaller ones or notions of system equivalence in a setting where no final coalgebra is known to exist.

Theorem 4.15 *Assume the category* C *has countable coproducts and we are given two functors* $\mathrm{T}, \mathrm{F} : \mathsf{C} \to \mathsf{C}$*, such that* T *distributes over* F *via* λ*, and two* F*-coalgebras* $\langle X, \alpha_X \rangle$ *and* $\langle Y, \alpha_Y \rangle$*. If* $\mathcal{B}$ *is a* λ*-bisimulation between* $\langle X, \alpha_X \rangle$ *and* $\langle Y, \alpha_Y \rangle$ *then there exists a (standard) bisimulation* $\tilde{\mathcal{B}}$ *between them with* $\mathcal{B} \preceq \tilde{\mathcal{B}}$.

Proof: Let χ be an FT-operation on B and β_X, β_Y be T-algebra operations on X, Y witnessing the λ-bisimulation property on $\mathcal{B} =: \langle B, b_1, b_2 \rangle$. For $\tilde{\mathcal{B}}$ we will then take $\langle \langle L_B, \alpha_\chi \rangle, \tilde{b_1}, \tilde{b_2} \rangle$ where L_B and α_χ are set as in Section 3.5 and for $i \in \{1, 2\}$ we set $\tilde{b_i} := [(b_i)_j]_{j=0}^\infty$ as in Lemma 3.13, i.e. $(b_i)_0 = b_i$, $(b_1)_{j+1} = (b_1)_j|^{\beta_X}$ and $(b_2)_{j+1} = (b_2)_j|^{\beta_Y}$.

By applying Lemma 3.13 on both sides, we get that $\tilde{b_1}$ and $\tilde{b_2}$ are homomorphisms from $\langle L_B, \alpha_\chi \rangle$ to $\langle X, \alpha_X \rangle$ and $\langle Y, \alpha_Y \rangle$ respectively. The order of spans is witnessed by $\mathtt{in}_0 : B \to L_B$, which immediately follows from the property of the countable case analysis and the definition of the $(b_i)_0$:

$$\begin{tikzcd}
 & B \arrow[dl, "b_1=(b_1)_0"'] \arrow[d, "\mathtt{in}_0"] \arrow[dr, "b_2=(b_2)_0"] & \\
X \arrow[d, "\alpha_X"'] & L_B \arrow[l, "\tilde{b_1}=[(b_1)_j]_{j=0}^\infty"] \arrow[r, "\tilde{b_2}=[(b_2)_j]_{j=0}^\infty"'] \arrow[d, "\alpha_\chi"'] & Y \arrow[d, "\alpha_Y"'] \\
\mathrm{F}X & \mathrm{F}L_B \arrow[l, "\mathrm{F}\,\tilde{b_1}"] \arrow[r, "\mathrm{F}\,\tilde{b_2}"'] & \mathrm{F}Y
\end{tikzcd}$$

□

Note that with the help of Lemma 4.4 this result still applies in a setting where a definition of bisimulation based on relations is used.

As above, a version of the Theorem using the condition from Theorem 3.10 can be given as well. This time it takes more effort to change the proof, because a different intermediate F-coalgebra has to be considered.

4.5 Example 1 revisited

Consider again the example from Section 4.2 where we had $\mathrm{T}X := X \times X$ and λ as given in Equation (1) on page 13. In this setting the definition of a λ-bisimulation can be spelled out as folllows:

Let $\langle X, \langle o_X, s_X \rangle\rangle$ and $\langle Y, \langle o_Y, s_Y \rangle\rangle$ be two stream systems, both coming with T-algebra operations $\oplus_X$ and $\oplus_Y$ turning them into λ-bialgebras. Literally, we get that a relation $B \subseteq X \times Y$ gives rise to a λ-bisimulation between $\langle X, \langle o_X, s_X \rangle\rangle$ and $\langle Y, \langle o_Y, s_Y \rangle\rangle$ with respect to $\oplus_X$ and $\oplus_Y$, if there exist $o : B \to \mathbb{R}$ and $\tilde{s}_1, \tilde{s}_2 : B \to B$ such that the diagram below commutes:

$$\begin{tikzcd}
X \times X \arrow[d, "\oplus_X"'] & & Y \times Y \arrow[d, "\oplus_Y"] \\
X \arrow[d, "{\langle o_X, s_X \rangle}"'] & B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "{\exists \langle o, \langle \tilde{s}_1, \tilde{s}_2 \rangle\rangle}"] & Y \arrow[d, "{\langle o_Y, s_Y \rangle}"] \\
\mathbb{R} \times X & \mathbb{R} \times (B \times B) \arrow[l, "{\mathtt{Id}_{\mathbb{R}} \times \pi_1|^{\oplus_X}}"] \arrow[r, "{\mathtt{Id}_{\mathbb{R}} \times \pi_2|^{\oplus_Y}}"'] & \mathbb{R} \times Y
\end{tikzcd}$$

This is equivalent to saying that for all $\langle x, y \rangle \in B$ we have that $o_X(x) = o_Y(y)$ and $\langle s_X(x), s_Y(y) \rangle \in T(B)$ for

$$T(B) := \{\langle x_1 \otimes_X x_2, y_1 \otimes_Y y_2 \rangle \mid \langle x_i, y_i \rangle \in B; i = 1, 2\} \in X \times Y.$$

By taking $\langle X, \langle o_X, s_X \rangle\rangle = \langle Y, \langle o_Y, s_Y \rangle\rangle = \langle \mathbb{R}^\omega, \langle \mathtt{head}, \mathtt{tail} \rangle\rangle$ and $\otimes_X = \otimes_Y = \otimes$ [3] we find that the assumption on the relation B in the principle given in Section 4.2 is just to be a λ-bisimulation on the final stream system.

Theorem 4.15 together with Lemma 4.4 tell us that this is enough to conclude that there is a bisimulation $\tilde{B} \subseteq \mathbb{R}^\omega \times \mathbb{R}^\omega$ with $B \subseteq \tilde{B}$. Looking into the proof of Theorem 4.15, we see that the concrete relation constructed is the same as $\tilde{B}$ from Section 4.2, since the span $\langle \mathrm{T}B, \pi_1|^{\oplus}, \pi_2|^{\oplus} \rangle$ represents the relation $T(B)$.

4.6 Example 2 revisited

While proving a statement about the interaction of the two insert functions in Section 4.3 we came across the notion of a bisimulation up-to-equality for coalgebras of a functor F in a category with binary coproducts. We are going to show that it arises as a λ-bisimulation on a final coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ with the setting from Section 3.2, i.e. $\mathrm{T}X := X + \Omega_{\mathrm{F}}$ and λ as in equation (2) on page 13.

The only choice for the T-algebra operation on Ω_{F} to yield a λ-bialgebra is $\langle \mathtt{Id}, \mathtt{Id} \rangle$, as shown earlier. For $b_i : B \to \Omega_{\mathrm{F}}$ we have $b_i|^{\langle \mathtt{Id}, \mathtt{Id} \rangle} = [b_i, \mathtt{Id}]$. Thus, a span $\langle B, b_1, b_2 \rangle$ on Ω_{F} is a λ-bisimulation if there exists an operation $\chi : B \to \mathrm{F}(B + \Omega_{\mathrm{F}})$ making the following diagram commute:

[3] The latter setting is possible as shown in subsection 3.4. Moreover, it is the only one by finality of $\langle \mathbb{R}^\omega, \langle \mathtt{head}, \mathtt{tail} \rangle\rangle$.

$$\begin{tikzcd}
\Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"'] & B \arrow[l, "b_1"'] \arrow[r, "b_2"] \arrow[d, dashed, "\exists\chi"] & \Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}\,\Omega_{\mathrm{F}} & \mathrm{F}(B+\Omega_{\mathrm{F}}) \arrow[l, "{\mathrm{F}\,[b_1,\mathtt{Id}]}"] \arrow[r, "{\mathrm{F}\,[b_2,\mathtt{Id}]}"'] & \mathrm{F}\,\Omega_{\mathrm{F}}
\end{tikzcd}$$

This can easily be seen to correspond to Definition 4.9 in case the span arises from a relation $B \subseteq \Omega_{\mathrm{F}} \times \Omega_{\mathrm{F}}$ as $\langle B, \pi_1, \pi_2 \rangle$.

With this correspondence, Theorem 4.15 says that every bisimulation up-to-equality is contained in some larger (standard) bisimulation, as claimed in Section 4.3. This time the span constructed in the proof of the theorem looks more complicated, but nevertheless is corresponds to the relation $B \cup \triangle_{\Omega_{\mathrm{F}}}$ that one would use in a "hand-made" proof. [4]

5 A more general Type of Distributive Law

In the previous sections we presented a definition schema and a proof principle which incorporated operations $\beta_{\mathrm{F}} : \mathrm{T}\,\Omega_{\mathrm{F}} \to \Omega_{\mathrm{F}}$ arising from a distributive law $\lambda : \mathrm{TF} \Rightarrow \mathrm{FT}$. As an example for such an operation we introduced the addition of two streams $\sigma, \tau \in I\!\!R^{\omega}$, specified by

$$\begin{aligned}
\mathtt{head}(\sigma \oplus \tau) &= \mathtt{head}(\sigma) + \mathtt{head}(\tau) \\
\mathtt{tail}(\sigma \oplus \tau) &= \mathtt{tail}(\sigma) \oplus \mathtt{tail}(\tau).
\end{aligned}$$

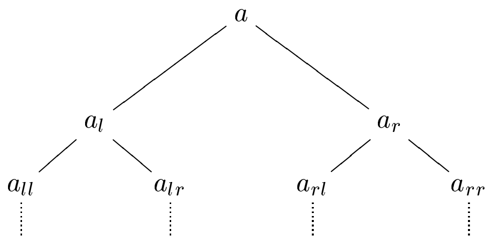
On the right hand side of these definitions, σ and τ are not mentioned themselves, but only their heads and tails.

Up to now we can only treat operations β_{F} of this kind, since they are assumed to form homomorphisms from $\mathrm{T}_{\lambda}\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle := \langle \mathrm{T}\,\Omega_{\mathrm{F}}, \mathit{lift}^{\lambda}_{\omega_{\mathrm{F}}} \rangle$ to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$, where in $\mathit{lift}^{\lambda}_{\omega_{\mathrm{F}}} := \lambda_{\Omega_{\mathrm{F}}} \circ \mathrm{T}\omega_{\mathrm{F}}$ all arguments from Ω_{F} in the elements of $\mathrm{T}\,\Omega_{\mathrm{F}}$ are first replaced by the F-step they can do according to the final operation ω_{F}.

Not all operations one is interested in are of that kind. One often meets specifications where both are used, the original arguments as well as the F-steps they can do. We will first present an example where this is the case. Then we show how to generalise our framework to this more powerful type of specification.

5.1 Example: Coding of infinite binary Trees into Streams

Consider infinite binary trees with the nodes labeled by real numbers. These do arise as the final coalgebra of the functor $\mathrm{B}X := I\!\!R \times X \times X$ denoted by $\langle \mathit{Btree}, \langle \mathtt{label}, \mathtt{left}, \mathtt{right} \rangle \rangle$. Let us name the labels of such a tree as follows:



The function $\mathtt{encode} : \mathit{Btree} \to I\!\!R^{\omega}$ should map such a tree onto the stream

$$\langle a, a_l, a_r, a_{ll}, a_{rl}, a_{lr}, a_{rr}, \ldots \rangle.$$

[4] Actually in this case the manual proof is more closely mimicked if one follows the approach based on monads as mentioned after Theorem 4.15.

To explain the procedure formally, we introduce the function that interleaves two streams $\sigma = \langle \sigma_0, \sigma_1, \ldots \rangle$ and $\tau = \langle \tau_0, \tau_1, \ldots \rangle$ yielding

$$\mathtt{interleave}(\sigma, \tau) := \langle \sigma_0, \tau_0, \sigma_1, \tau_1, \ldots \rangle,$$

defined by coiteration from the equations

$$\begin{aligned} \mathtt{head}(\mathtt{interleave}(\sigma, \tau)) &= \mathtt{head}(\sigma), \\ \mathtt{tail}(\mathtt{interleave}(\sigma, \tau)) &= \mathtt{interleave}(\tau, \mathtt{tail}(\sigma)). \end{aligned}$$

Now for $t \in \mathit{Btree}$ the function `encode` satisfies

$$\begin{aligned} \mathtt{head}(\mathtt{encode}(t)) &= \mathtt{label}(t) \\ \mathtt{tail}(\mathtt{encode}(t)) &= \mathtt{interleave}(\mathtt{encode}(\mathtt{left}(t)), \mathtt{encode}(\mathtt{right}(t))). \end{aligned}$$

The equations are of the same shape as those for the multiplication of formal power series in Section 3.1. So we could again use the functor $\mathrm{T}X := X \times X$ and try to find a distributive law $\lambda : \mathrm{TS} \Rightarrow \mathrm{S\,T}$ such that the resulting T-algebra operation β_{S} on the final S-coalgebra happens to be `interleave`.

But unlike the situation with the addition, here one of the arguments of `interleave` (τ in the above formulation) appears by itself on the right hand side of the equation for the tail. Given only $\mathtt{head}(\tau)$ and $\mathtt{tail}(\tau)$ we do not know how to express $\mathtt{tail}(\mathtt{interleave}(\sigma, \tau))$. Of course, in the final coalgebra one could use $\langle \mathtt{head}, \mathtt{tail} \rangle^{-1}(\mathtt{head}(\tau), \mathtt{tail}(\tau)) = \tau$, since the final operation is an isomorphism, but we have to define λ for arbitrary carrier sets where we do not have an operation that gives us the original state when we provide its head and tail.

5.2 Extended Distributive Laws

We want to capture T-algebra operations β_{F} on the carrier of a final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ that use in their definition the original arguments from Ω_{F} appearing in an element of $\mathrm{T}\Omega_{\mathrm{F}}$ and their F-steps, like `interleave` from above. To do so, one could switch to a distributive law of the shape

$$\lambda : \mathrm{T}(\mathrm{Id} \times \mathrm{F}) \Rightarrow \mathrm{F\,T}.^{5} \tag{4}$$

This would change the definition of $\mathit{lift}_\alpha^\lambda$ for an F-coalgebra operation $\alpha : X \to \mathrm{F}X$ such that instead of just applying α to all the arguments from X inside an element of $\mathrm{T}X$ one takes $\langle \mathtt{Id}, \alpha \rangle$ to turn the arguments into pairs where the first component keeps the original:

$$\mathit{lift}_\alpha^\lambda := \lambda_X \circ \mathrm{T}\langle \mathtt{Id}, \alpha \rangle.$$

Changing Definition 3.5 accordingly, we again obtain a functor $\mathrm{T}_\lambda : \mathsf{Coalg}_{\mathrm{F}} \to \mathsf{Coalg}_{\mathrm{F}}$ (in particular, Lemma 3.6 would still hold).

In this extended framework it is now possible to capture e.g. `interleave` from above: For $\mathrm{T}X := X \times X$, the interleaving of streams arises as the coiterative arrow from $\mathrm{T}_\lambda \langle \mathbb{R}^\omega, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ to $\langle \mathbb{R}^\omega, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ for the natural transformation

$$\begin{aligned} \lambda &: \mathrm{T}(\mathrm{Id} \times \mathrm{S}) \Rightarrow \mathrm{S\,T}, \\ \text{i.e.} \quad \lambda_X &: (X \times (\mathbb{R} \times X)) \times (X \times (\mathbb{R} \times X)) \to \mathbb{R} \times (X \times X) \end{aligned}$$

given by

$$\lambda_X(\langle x, \langle x_0, x' \rangle \rangle, \langle y, \langle y_0, y' \rangle \rangle) := \langle x_0, \langle y, x' \rangle \rangle. \tag{5}$$

[5] Note that when talking about such a natural transformation, we implicitly assume that the category C has binary products.

5.3 Adapting the λ-Coiteration Definition Schema

Our aim now is to show that when working with a distributive law λ of the extended type introduced above we still have the results about λ-coiterative morphisms (with the straightforward adaptation of this notion) as shown in Section 3.5. The arguments basically stay the same, therefore we will just mention the small changes needed in the definitions and statements given there.

First, now that λ is of the shape $\mathrm{T}(\mathrm{Id} \times \mathrm{F}) \Rightarrow \mathrm{F}\,\mathrm{T}$, the notion of a λ-bialgebra would be a T,F-bialgebra $\langle X, \alpha, \beta\rangle$ such that the following diagram commutes:

$$\begin{tikzcd}
 & \mathrm{T}X \arrow[dl, "\mathrm{T}\langle \mathtt{Id}_X{,}\alpha\rangle"'] \arrow[dd, "\beta"] \\
\mathrm{T}(X \times \mathrm{F}X) \arrow[dd, "\lambda_X"'] & \\
{} \arrow[r, phantom, "\lambda\text{-bialg.}"] & X \arrow[dd, "\alpha"] \\
\mathrm{F}\,\mathrm{T}X \arrow[dr, "\mathrm{F}\,\beta"'] & \\
 & \mathrm{F}X
\end{tikzcd}$$

Given a final F-coalgebra, we can use this notion of a λ-bialgebra as before to obtain a definition of a λ-coiterative arrow induced by an operation $\phi : X \to \mathrm{F}\,\mathrm{T}X$.

Adapting Theorem 3.9 to the new setting is slightly more complicated, because the proof does not only use $\mathit{lift}^{\lambda}_{\alpha}$ for a F-coalgebra operation α, but $\mathit{lift}^{\lambda}_{\phi}$ for the $\mathrm{F}\,\mathrm{T}$-coalgebra ϕ. The problem is that for the second type of lifting we cannot simply use a pairing with the identity as before to keep the arguments themselves, since by using ϕ we "change the carrier" from X to $\mathrm{T}X$: the application of $\mathrm{T}\langle \mathtt{Id}_X, \phi\rangle$ would get us from $\mathrm{T}X$ to $\mathrm{T}(X \times \mathrm{F}\,\mathrm{T}X)$, but we cannot apply λ to the latter. As a remedy, we will ask for an embeding of X into $\mathrm{T}X$, i.e. a natural transformation $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$. Note that a pair $\langle \mathrm{T}, \eta\rangle$ of a functor with such a transformation is known in the literature as a *pointed endofunctor*. With this additional structure we can now construct an $\mathrm{F}\,\mathrm{T}$-coalgebra operation on $\mathrm{T}X$:

$$\mathit{lift}^{\lambda,\eta}_{\phi} := \lambda_{\mathrm{T}X} \circ \mathrm{T}\langle \eta_X, \phi\rangle.$$

Not surprisingly, this construction only makes sense if we can show that the embedding η_X preserves behaviours in a certain sense. For this we will make an assumption on the interplay of η and λ, namely that they make the diagram below in the category of functors and natural transformations commute. Since η is usually called the *unit* of the pointed functor, we will refer to this condition later as the *unit law for* λ:

$$\begin{tikzcd}
\mathrm{Id} \times \mathrm{F} \arrow[r, Rightarrow, "\pi_2"] \arrow[d, Rightarrow, "\eta_{\mathrm{Id}\times\mathrm{F}}"'] \arrow[dr, phantom, "\text{unit } \lambda"] & \mathrm{F} \arrow[d, Rightarrow, "\mathrm{F}\,\eta"] \\
\mathrm{T}(\mathrm{Id} \times \mathrm{F}) \arrow[r, Rightarrow, "\lambda"'] & \mathrm{F}\,\mathrm{T}
\end{tikzcd}$$

We can now state the adapted version of Theorem 3.9:

Theorem 5.1 *Let the category* C *have countable coproducts. Assume we are given a* C*-functor* F *with the final coalgebra* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$*, a pointed functor* $\langle \mathrm{T}, \eta\rangle$ *in* C *and a natural transformation* $\lambda : \mathrm{T}(\mathrm{Id} \times \mathrm{F}) \Rightarrow \mathrm{F}\,\mathrm{T}$ *satisfying the unit law from above. Then for every* $\mathrm{F}\,\mathrm{T}$*-coalgebra* $\langle X, \phi\rangle$ *there exists a unique* λ*-coiterative morphism induced by* ϕ*.*

In the following we will sum up the changes needed to adapt the proof of Theorem 3.9 to this new setting.

The F-coalgebra constructed from $\langle X, \phi\rangle$ is again $\langle L_X, \alpha_{\phi}\rangle$ as given in subsection 3.6 with the inductive case in the definition of the ϕ_i changed to

$$\phi_{i+1} := \lambda_{\mathrm{T}^{i+1}X} \circ \mathrm{T}\langle \eta_{\mathrm{T}^{i}X}, \phi_i\rangle.$$

Note that this adaptation does not influence the validity of Lemma 3.11.

In order to prove a correspondent of Lemma 3.12 we need a new statement justifying the use of η_X as an embedding:

Lemma 5.2 *In the setting of Theorem 5.1 let $h = [h_i]_{i=0}^{\infty}$ denote the coiterative morphism from $\langle L_X, \alpha_\phi \rangle$ as above to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$. Then for all $i \in I\!N$ we have*

$$h_{i+1} \circ \eta_{\mathrm{T}^{\mathrm{i}} X} = h_i.$$

Proof: We claim that the arrow $[\mathtt{in}_{j+1} \circ \eta_{\mathrm{T}^{\mathrm{j}} X}]_{j=0}^{\infty}$ is a homomorphism from the F-coalgebra $\langle L_X, \alpha_\phi \rangle$ to itself. By finality we then get $h = h \circ [\mathtt{in}_{j+1} \circ \eta_{\mathrm{T}^{\mathrm{j}} X}]_{j=0}^{\infty}$, which contains the statement. To prove the claim, we need to show that the following diagram commutes:

$$\begin{tikzcd}
L_X \arrow[r, "{[\mathtt{in}_{j+1} \circ \eta_{\mathrm{T}^{\mathrm{j}} X}]_{j=0}^{\infty}}"] \arrow[d, "\alpha_\phi"'] & L_X \arrow[d, "\alpha_\phi"] \\
\mathrm{F} L_X \arrow[r, "{\mathrm{F}\, [\mathtt{in}_{j+1} \circ \eta_{\mathrm{T}^{\mathrm{j}} X}]_{j=0}^{\infty}}"'] & \mathrm{F} L_X
\end{tikzcd}$$

By case analysis, this boils down to the commutativity of the outer part of the picture below for all $j \in I\!N$ (abbreviating $\mathrm{T}^{\mathrm{j}} X$ to Y). The inner rectangles commute by the naturality of η and the unit law for λ:

$$\begin{tikzcd}
Y \arrow[r, "\eta_Y"] \arrow[d, "{\langle \eta_Y, \phi_j \rangle}"] \arrow[dr, phantom, "\text{nat. } \eta"] \arrow[dd, bend right=60, "\phi_j"'] & \mathrm{T} Y \arrow[d, "{\mathrm{T}\,\langle \eta_Y, \phi_j \rangle}"'] \arrow[dd, bend left=60, "\phi_{j+1}"] \\
\mathrm{T} Y \times \mathrm{F}\mathrm{T} Y \arrow[r, "\eta_{\mathrm{T} Y \times \mathrm{F}\mathrm{T} Y}"'] \arrow[d, "\pi_2"] \arrow[dr, phantom, "\text{unit } \lambda"] & \mathrm{T}(\mathrm{T} Y \times \mathrm{F}\mathrm{T} Y) \arrow[d, "\lambda_{\mathrm{T} Y}"'] \\
\mathrm{F}\mathrm{T} Y \arrow[r, "\mathrm{F}\, \eta_{\mathrm{T} Y}"'] & \mathrm{F}\mathrm{T}^2 Y
\end{tikzcd}$$

□

Lemma 5.3 (Lemma 3.12 (updated)) *In the setting of Theorem 5.1, let $h = [h_i]_{i=0}^{\infty}$ be the coiterative morphism from the* F*-coalgebra $\langle L_X, \alpha_\phi \rangle$ as above to the final* F*-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$. Then we have*

$$h_{i+1} = h_i|^{\beta_{\mathrm{F}}} \quad (:= \beta_{\mathrm{F}} \circ \mathrm{T}\, h_i)$$

for all $i \in I\!N$, where β_{F} is again the coiterative morphism from $\mathrm{T}_\lambda \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ to $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$.

Proof: The proof runs along the same line as the one for Lemma 3.12. Only one change has to be made when it comes to showing the homomorphism property of $[\mathrm{T}\, h_j]_{j=0}^{\infty}$. The diagram that does the main job changes to the following one (for $j \in I\!N$ and $Y := \mathrm{T}^{\mathrm{j}} X$):

$$\begin{tikzcd}
\mathrm{T} Y \arrow[r, "\mathrm{T}\, h_j"] \arrow[d, "{\mathrm{T}\,\langle \eta_Y, \phi_j \rangle}"] \arrow[dr, phantom, "\mathrm{T}\,((*))"] \arrow[dd, bend right=60, "\phi_{j+1}"'] & \mathrm{T} \Omega_{\mathrm{F}} \arrow[d, "{\mathrm{T}\,\langle \mathtt{Id}, \omega_{\mathrm{F}} \rangle}"'] \arrow[dd, bend left=60, "lift^{\lambda}_{\omega_{\mathrm{F}}}"] \\
\mathrm{T}(\mathrm{T} Y \times \mathrm{F}\mathrm{T} Y) \arrow[r, "\mathrm{T}\,(h_{j+1} \times \mathrm{F}\, h_{j+1})"] \arrow[d, "\lambda_{\mathrm{T} Y}"] \arrow[dr, phantom, "\text{nat. } \lambda"] & \mathrm{T}(\Omega_{\mathrm{F}} \times \mathrm{F} \Omega_{\mathrm{F}}) \arrow[d, "\lambda_{\Omega_{\mathrm{F}}}"'] \\
\mathrm{F}\mathrm{T}^2 Y \arrow[r, "\mathrm{F}\,\mathrm{T}\, h_{j+1}"'] & \mathrm{F}\mathrm{T} \Omega_{\mathrm{F}}
\end{tikzcd}$$

Now the upper rectangle is the T-image of a diagram (*) that consists of the two parts shown below. The left one is Lemma 5.2, the right one follows from the assumption on h being a homomorphism by (the adapted version of) Lemma 3.11:

$$\begin{tikzcd}
Y \arrow[r, "h_j"] \arrow[d, "\eta_Y"'] & \Omega_{\mathrm{F}} \arrow[d, Rightarrow, "\mathtt{Id}"] \\
\mathrm{T}Y \arrow[r, "h_{j+1}"'] & \Omega_{\mathrm{F}}
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[r, "h_j"] \arrow[d, "\phi_j"'] & \Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}\mathrm{T}Y \arrow[r, "\mathrm{F}\,h_{j+1}"'] & \mathrm{F}\Omega_{\mathrm{F}}
\end{tikzcd}$$

□

As a consequence of moving from a simple functor T to the pointed functor $\langle \mathrm{T}, \eta \rangle$, we set up an additional assumption on the distributive laws, namely that they should satisfy the unit law. It turns out that it is appropriate to impose a new law on the algebras under consideration as well. We will be working with *algebras for the pointed functor* $\langle \mathrm{T}, \eta \rangle$, these are T-algebras $\langle Z, \beta \rangle$ satisfying the following unit law:

$$\begin{tikzcd}
Z \arrow[r, "\eta_Z"] \arrow[dr, Rightarrow, bend right, "\mathtt{Id}"'] \arrow[dr, phantom, "\text{unit } \beta"] & \mathrm{T}Z \arrow[d, "\beta"] \\
 & Z
\end{tikzcd}$$

Accordingly, when talking about a λ-bialgebra in this context, we will assume that the T-algebra operation involved is an algebra for the pointed functor $\langle \mathrm{T}, \eta \rangle$.

With this new reading of a λ-bialgebra, we can take over Lemma 3.13 as it is. Again the proof is very similar to the original one, except that the diagram that handles the inductive case of the induction proof contained gets slightly more complicated (for $i \in I\!N$ and $Y := \mathrm{T}^{\mathrm{i}} X$):

$$\begin{tikzcd}
 & & \mathrm{T}Z \arrow[d, "\beta"] \\
\mathrm{T}Y \arrow[r, "\mathrm{T}\,f_i"'] \arrow[rr, dashed, bend left, "f_{i+1}=f_i|^{\beta}"] \arrow[d, "\mathrm{T}\,\langle \eta_Y, \phi_i \rangle"] \arrow[dd, dashed, bend right=60, "\phi_{i+1}"'] \arrow[dr, phantom, "\mathrm{T}((*))"] & \mathrm{T}Z \arrow[r, "\beta"'] \arrow[d, "\mathrm{T}\,\langle \mathtt{Id}, \alpha \rangle"] \arrow[dr, phantom, "\lambda\text{-bialg}"] & Z \arrow[dd, "\alpha"] \\
\mathrm{T}(\mathrm{T}Y \times \mathrm{F}\mathrm{T}Y) \arrow[r, "\mathrm{T}(f_i|^{\beta} \times \mathrm{F}\,f_i|^{\beta})"'] \arrow[d, "\lambda_{\mathrm{T}Y}"] \arrow[dr, phantom, "\text{nat.}\,\lambda"] & \mathrm{T}(Z \times \mathrm{F}Z) \arrow[d, "\lambda_Z"] & \\
\mathrm{F}\mathrm{T}^2 Y \arrow[r, "\mathrm{F}\,\mathrm{T}\,f_i|^{\beta}", "=\mathrm{F}\,\mathrm{T}\,f_{i+1}"'] \arrow[rr, dashed, bend right, "\mathrm{F}\,f_{i+1}|^{\beta}"'] & \mathrm{F}\mathrm{T}Z \arrow[r, "\mathrm{F}\,\beta"] & \mathrm{F}Z
\end{tikzcd}$$

Now the upper left rectangle is the T-image of a diagram that combines the two below. For the left one we need the additional assumption on β satisfying the unit law, with the right one we recover the induction hypothesis:

$$\begin{tikzcd}
Y \arrow[r, "f_i"] \arrow[d, "\eta_Y"'] \arrow[dr, phantom, "\text{nat. } \eta"] & Z \arrow[d, "\eta_Z"] \arrow[dr, Rightarrow, bend left, "\mathtt{Id}"] \arrow[dr, phantom, "\text{unit } \beta"] & \\
\mathrm{T}Y \arrow[r, "\mathrm{T}\,f_i"] \arrow[rr, bend right, "f_i|^{\beta}"'] & \mathrm{T}Z \arrow[r, "\beta"] & Z
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[r, "f_i"] \arrow[d, "\phi_i"'] \arrow[dr, phantom, "\text{ind. Hyp.}"] & Z \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}\mathrm{T}Y \arrow[r, "\mathrm{F}\,f_i|^{\beta}"'] & \mathrm{F}Z
\end{tikzcd}$$

To be able to apply the new reading of Lemma 3.13 inside the proof of Theorem 5.1, we have to check that the bialgebra $\langle \Omega_{\mathrm{F}}, \beta_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$ that we have to instantiate it with is indeed a λ-bialgebra in the new sense:

Lemma 5.4 *Using the setting of Theorem 5.1 we have that the coiterative morphism β_{F} from $\mathrm{T}_\lambda\langle\Omega_{\mathrm{F}},\omega_{\mathrm{F}}\rangle$ to $\langle\Omega_{\mathrm{F}},\omega_{\mathrm{F}}\rangle$ is a $\langle\mathrm{T},\eta\rangle$-algebra operation, i.e. it satisfies the appropriate unit law.*

Proof: We can easily show that in case λ satisfies the unit law, we have that η_X is an F-coalgebra homomorphism from any given F-coalgebra $\langle X,\alpha\rangle$ to $\mathrm{T}_\lambda\langle X,\alpha\rangle$. This yields that the diagram below exists in $\mathsf{Coalg}_{\mathrm{F}}$. It commutes by finality. The image of it under the forgetful functor is the unit law for β_{F}.

$$\begin{tikzcd}
\langle\Omega_{\mathrm{F}},\omega_{\mathrm{F}}\rangle \arrow[r, "\eta_{\Omega_{\mathrm{F}}}"] \arrow[dr, Rightarrow, bend right, "\mathtt{Id}"'] \arrow[dr, phantom, "\text{finality}"] & \mathrm{T}_\lambda\langle\Omega_{\mathrm{F}},\omega_{\mathrm{F}}\rangle \arrow[d, "\beta_{\mathrm{F}}"] \\
 & \langle\Omega_{\mathrm{F}},\omega_{\mathrm{F}}\rangle
\end{tikzcd}$$

□

With these adaptations of the statements used in the proof of Theorem 3.9 we can prove Theorem 5.1 along the same line.

The initial goal was to show that we can allow distributive laws of the more general type form above in Theorem 3.9. But Theorem 5.1 introduces the additional need for a unit η interacting nicely with λ – which we may not have as in the case of the interleaving function that we started out with. But in case the category has binary coproducts as well, we can move from the setting for the functor T to one for a modified functor having a unit. Then we can apply Theorem 5.1 in the second setting and transform the result back. This leads to the following corollary:

Corollary 5.5 *Let the category C have binary and countable coproducts. Assume we are given the C-functors F and T with the final F-coalgebra $\langle\Omega_{\mathrm{F}},\omega_{\mathrm{F}}\rangle$ and a natural transformation $\lambda : \mathrm{T}(\mathrm{Id}\times\mathrm{F}) \Rightarrow \mathrm{F}\mathrm{T}$.*

Then for every $\mathrm{F}\mathrm{T}$-coalgebra $\langle X,\phi\rangle$ there exists a unique λ-coiterative morphism induced by ϕ.

Proof: Consider the following transformation:

$$\begin{array}{rcl}
\tilde{\mathrm{T}} & := & \mathrm{Id}+\mathrm{T} \\
\eta & := & \mathtt{in}_l : \mathrm{Id} \Rightarrow \tilde{\mathrm{T}} \\
\nu & := & \mathtt{in}_r : \mathrm{T} \Rightarrow \tilde{\mathrm{T}} \\
\tilde{\lambda} & := & [\mathrm{F}\eta\circ\pi_2, \mathrm{F}\nu\circ\lambda] : \tilde{\mathrm{T}}(\mathrm{Id}\times\mathrm{F}) \Rightarrow \mathrm{F}\tilde{\mathrm{T}} \\
\tilde{\beta} & := & [\mathtt{Id}_Z,\beta] : \tilde{\mathrm{T}}Z \to Z \quad (\text{for } \beta : \mathrm{T}Z \to Z) \\
\tilde{\phi} & := & \mathrm{F}\nu_X\circ\phi : X \to \mathrm{F}\tilde{\mathrm{T}}X \quad (\text{for } \phi : X \to \mathrm{F}\mathrm{T}X)
\end{array}$$

This gives us:

- a pointed functor $\langle\tilde{\mathrm{T}},\eta\rangle$,
- a natural transformation $\tilde{\lambda}$ satisfying the unit law,
- for a T-algebra $\langle Z,\beta\rangle$ we get a $\tilde{\mathrm{T}}$-algebra operation $\tilde{\beta}$ satisfying the unit law,
- moreover, for a λ-bialgebra $\langle Z,\beta,\alpha\rangle$ we get a $\tilde{\lambda}$-bialgebra $\langle Z,\tilde{\beta},\alpha\rangle$,
- an arrow $h : X \to Z$ is a homomorphism up-to-β from the FT-coalgebra $\langle X,\phi\rangle$ to the F-coalgebra $\langle Z,\alpha\rangle$ if and only if it is a homomorphism up-to-$\tilde{\beta}$ from the $\mathrm{F}\tilde{\mathrm{T}}$-coalgebra $\langle X,\tilde{\phi}\rangle$ to $\langle Z,\alpha\rangle$.

From these statements we can conclude that for an FT-coalgebra $\langle X,\phi\rangle$ a λ-coiterative arrow induced by ϕ fits as a $\tilde{\lambda}$-coiterative arrow induced by the $\mathrm{F}\tilde{\mathrm{T}}$-operation $\tilde{\phi}$ and vice versa. Thus, Theorem 5.1 proves the statement instantiated with $\langle\tilde{\mathrm{T}},\eta\rangle$ and $\tilde{\lambda}$. □

Coming back to our guiding example from Section 5.1, the above corollary tells us that the equations given for `encode` uniquely define the function. Technically, for $\mathrm{T}X := X \times X$ and λ as in equation (5) on page 29, it is the λ-coiterative arrow induced by

$$\langle \mathtt{label}, \mathtt{left}, \mathtt{right} \rangle : Btree \to \underbrace{\mathbb{R} \times Btree \times Btree}_{= \mathrm{S}\,\mathrm{T}(Btree)}.$$

5.4 Adapting the λ-Coinduction Proof Technique

In a similar procedure one can extend the notion of a λ-bisimulation to distributive laws λ as in (4) on page 29 and prove an adapted version of Theorem 4.15. In a first step one would state it analog to Theorem 5.1 for a pointed functor $\langle \mathrm{T}, \eta \rangle$ and λ as assumed there. Here the λ-bisimulation has to be taken with respect to T-algebra operations satisfying the unit law as in 5.3. This time the transformation from the proof of Corollary 5.5 yields the following:

Corollary 5.6 *Assume the category* C *has binary and countable coproducts and we are given two functors* $\mathrm{T}, \mathrm{F} : \mathsf{C} \to \mathsf{C}$*, such that* T *distributes over* F *via a distributive law* λ *as in (4), and two* F*-coalgebras* $\langle X, \alpha_X \rangle$ *and* $\langle Y, \alpha_Y \rangle$*. If* $\mathcal{B}$ *is a* λ*-bisimulation (defined using the adapted notion of a* λ*-bialgebra) between* $\langle X, \alpha_X \rangle$ *and* $\langle Y, \alpha_Y \rangle$ *then there exists a (standard) bisimulation* $\tilde{\mathcal{B}}$ *between them with* $\mathcal{B} \preceq \tilde{\mathcal{B}}$.

5.5 Extended Distributive Laws and Theorem 3.10

As of yet we have only talked about distributive laws of the more expressive type in the context of the first version of the λ-coiteration Theorem 3.9. Of course one can treat the second version (Theorem 3.10) as well. As it turns out, it even allows for a more structured approach. Again, we will not give the details of the construction here, but we will mention the outline for the interested reader. The idea is to first prove a version of Theorem 3.10 where the functor F is replaced by a *copointed functor* $\langle \tilde{\mathrm{F}}, \epsilon \rangle$. (The final coalgebra is replaced by a final coalgebra for the copointed functor, for the distributive law there is an additional *counit law*, and $\phi : X \to \tilde{\mathrm{F}}\,\mathrm{T}X$ has to satisfy a unit/counit law $\epsilon_{\mathrm{T}X} \circ \phi = \eta_X$.) Then one transforms the initial problem to one fitting into the latter framework by setting $\langle \tilde{\mathrm{F}}, \epsilon \rangle = \langle \mathrm{Id} \times \mathrm{F}, \pi_1 \rangle$. The resulting theorem would be the following:

Theorem 5.7 *Assume we are given a functor* $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ *with a final coalgebra* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}} \rangle$*, a monad* $\langle \mathrm{T}, \eta, \mu \rangle$ *in* C *and a distributive law* λ *of the functor* T *over* F *of the type (4) on page 4 such that the following diagrams commute:*

$$\begin{tikzcd}
\mathrm{Id} \times \mathrm{F} \arrow[r, Rightarrow, "\pi_2"] \arrow[d, Rightarrow, "\eta_{\mathrm{Id}\times\mathrm{F}}"'] & \mathrm{F} \arrow[d, Rightarrow, "\mathrm{F}\,\eta"] \\
\mathrm{T}(\mathrm{Id} \times \mathrm{F}) \arrow[r, Rightarrow, "\lambda"'] & \mathrm{F}\,\mathrm{T}
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathrm{T}(\mathrm{Id} \times \mathrm{F}) \arrow[rr, Rightarrow, "\lambda"] & & \mathrm{F}\,\mathrm{T} & \\
\mathrm{T}^2(\mathrm{Id} \times \mathrm{F}) \arrow[ur, Rightarrow, "\mu(\mathrm{Id}\times\mathrm{F})"] \arrow[drr, Rightarrow, "\mathrm{T}\langle \mathrm{T}\pi_1, \lambda \rangle"'] & & & & \mathrm{F}\,\mathrm{T}^2 \arrow[ul, Rightarrow, "\mathrm{F}\mu"'] \\
& & \mathrm{T}(\mathrm{T} \times \mathrm{F}\,\mathrm{T}) \arrow[urr, Rightarrow, "\lambda\mathrm{T}"'] & &
\end{tikzcd}$$

Then, for every $\mathrm{F}\,\mathrm{T}$*-coalgebra* $\langle X, \phi \rangle$ *there exists a unique* λ*-coiterative arrow induced by* ϕ.

6 Example: The Dual of Course-of-Value Iteration

In this section we are going to present yet another example for the theory explained above. The basic coiteration schema has the property that exactly one stage of a behaviour gets defined by unfolding the specification once. Here we will develop a format that weakens this close relationship. It seems reasonable to demand that the specification should determine *at least* one stage at a time to make sure that the definition is what is sometimes called *productive*, i.e. that eventually all information about the behaviour can be derived. But it would not hurt if more than one stage would be revealed by unfolding the behavioural specification once.

As a simple example consider infinite streams of elements from a set A, i.e. A^ω, the final coalgebra of the Set-functor $\mathrm{S_A}\,X := A \times X$. A function into those streams could be specified such that for each state in the domain a nonempty prefix of the resulting stream and the argument for the construction of the rest is given. It is not surprising that this still uniquely assigns a stream to each element, as one could prove by giving a construction that would split one such transition $s \overset{w}{\to} s'$ with a whole prefix $w = a_1 \ldots a_n \in A^+$ into several one element transitions via fresh intermediate states $s_1, \ldots, s_{n-1}$:

$$s \overset{a_1}{\to} s_1 \overset{a_2}{\to} \cdots \overset{a_{n-1}}{\to} s_{n-1} \overset{a_n}{\to} s'.$$

The schema that defines streams in A^ω not via coalgebras of the functor $\mathrm{S_A}\,X := A \times X$ but via $(A^+ \times -)$-coalgebras arises as a λ-coiteration schema for a certain functor T and an extended distributive law $\lambda : \mathrm{T}(\mathrm{Id} \times \mathrm{S_A}) \Rightarrow \mathrm{S_A}\,\mathrm{T}$. The functor would be $\mathrm{T}X := A^* \times X$, the distributive law is set as

$$\lambda_X\langle w, \langle x, \langle x_0, x'\rangle\rangle\rangle := \begin{cases} \langle a, \langle w', x\rangle\rangle & \text{if } w = aw', \\ \langle x_0, \langle \epsilon, x'\rangle\rangle & \text{if } w = \epsilon. \end{cases}$$

for $w, w' \in A^*$; $a, x_0 \in A$; $x, x' \in X$ where $\epsilon \in A^*$ denotes the empty word.

To see what the corresponding λ-coiteration schema would be like, we need to check what the T-algebra operation on the final S_A-coalgebra $\langle A^\omega, \langle \mathtt{head}, \mathtt{tail}\rangle\rangle$ would be. It turns out to be the prefixing $\mathtt{prefix} : A^* \times A^\omega \to A^\omega$ of an infinite stream $\sigma \in A^\omega$ with the finite word $w \in A^*$, that we can either describe by corecursion on the resulting stream or by iteration on w:

$$\begin{aligned} \mathtt{prefix}(\epsilon, \sigma) &:= \sigma \\ \mathtt{prefix}(aw, \sigma) &:= \langle \mathtt{head}, \mathtt{tail}\rangle^{-1}(a, \mathtt{prefix}(w, \sigma)). \end{aligned}$$

With the observation that $\mathrm{S_A}\,\mathrm{T}$-coalgebras can equivalently be expressed as $(A^+ \times -)$-coalgebras and some rearrangement of the resulting diagram we get that the λ-coiterative arrow $h : X \to A^\omega$ induced by an operation $\phi : X \to A^+ \times X$ is the unique arrow fitting in the diagram below:

$$\begin{tikzcd} X \arrow[r, dashed, "h"] \arrow[d, "\phi"'] & A^\omega \\ A^+ \times X \arrow[r, dashed, "\mathtt{Id} \times h"'] & A^+ \times A^\omega \arrow[u, "\mathtt{prefix}"'] \end{tikzcd}$$

One function that could be specified nicely using this schema is the function $\mathtt{subst}_t : A^\omega \to B^\omega$ that, for a given mapping $t : A \to B^+$, outputs for every stream $\langle a_0 a_1 \ldots\rangle \in A^\omega$ the stream $\langle t(a_0)t(a_1)\ldots\rangle \in B^\omega$ that results after replacing every element a_i by the word $t(a_i)$. It arises as the λ-coiterative arrow (for λ as above) induced by

$$(t \times \mathtt{Id}) \circ \langle \mathtt{head}, \mathtt{tail}\rangle : A^\omega \to B^+ \times A^\omega.$$

Turning to the corresponding proof technique, we have that a relation R is a λ-bisimulation on the final S_A-coalgebra $\langle A^\omega, \langle \mathtt{head}, \mathtt{tail}\rangle\rangle$ if for every pair $\langle \sigma, \tau\rangle \in R$ there is a common nonempty prefix $w \in A^+$ such that $\sigma = \mathtt{prefix}(w, \sigma')$ and $\tau = \mathtt{prefix}(w, \tau')$ for $\langle \sigma', \tau'\rangle \in R$.

As an example one could use this notion to prove that for two functions $t_1 : A \to B^+$ and $t_2 : B \to C^+$ we have that

$$\mathtt{subst}_{t_2} \circ \mathtt{subst}_{t_1} = \mathtt{subst}_{\tilde{t_2} \circ t_1}$$

for the element-wise extension $\tilde{t_2} : B^+ \to C^+$ of t_2 to nonempty words. This follows from the fact that the relation

$$R := \{\langle \mathtt{subst}_{t_2}(\mathtt{subst}_{t_1}(\sigma)), \mathtt{subst}_{\tilde{t_2} \circ t_1}(\sigma)\rangle \mid \sigma \in A^\omega\}$$

is a λ-bisimulation. (To show this, one would use the lemma

$$\mathtt{subst}_t(\mathtt{prefix}(w,\sigma)) = \mathtt{prefix}(\tilde{t}(w), \mathtt{subst}_t(\sigma)),$$

which is easily proved by induction on $w \in A^*$.)

Under certain assumptions the above situation can be generalised from stream systems to F-coalgebras for functors $\mathrm{F} : \mathsf{C} \to \mathsf{C}$ with a final coalgebra $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$. For the interested reader we briefly outline the construction.

The assumptions are that C has binary coproducts and that for every object Z in C the functor $\mathrm{F}_\mathrm{Z} X := Z + \mathrm{F}X$ has an initial algebra – say $\langle \Delta_Z, \delta_Z \rangle$. Then the mapping $Z \mapsto \Delta_Z$ can be extended to a functor Δ, using initiality for its action on arrows. To generalise the distributive law from above to this setting, we need a natural transformation $\lambda : \Delta(\mathrm{Id} \times \mathrm{F}) \Rightarrow \mathrm{F}\Delta$. It can be set as $\lambda_X := [\mathrm{F}\eta_X \circ \pi_2, \mathrm{F}\Delta\pi_1] \circ \delta^{-1}_{X \times \mathrm{F}X}$ where $\eta_X := \delta_X \circ \mathtt{in}_l : X \to \Delta X$. Let β_F denote again the coiterative arrow from $\Delta_\lambda \langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ to $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$. Theorem 5.1 says that in case C has countable coproducts[6] a specification $\phi : X \to \mathrm{F}\Delta X$ uniquely characterises an arrow $f : X \to \Omega_\mathrm{F}$ by requiring it to satisfy $\omega_\mathrm{F} \circ f = \mathrm{F} f|^{\beta_\mathrm{F}} \circ \phi$.

The above λ-coiteration schema turns out to be the same as Vene and Uustalu's [UV99] characterisation of "futurmorphisms". They state that there is a unique arrow satisfying the equation from above with $f|^{\beta_\mathrm{F}}$ replaced by the iterative arrow from $\langle \Delta_X, \delta_X \rangle$ to $\langle \Omega_\mathrm{F}, [f, \omega_\mathrm{F}^{-1}] \rangle$. But the two are equal, which easily follows from the observation that β_F can be seen to coincide with the iterative arrow from $\langle \Delta_{\Omega_\mathrm{F}}, \delta_{\Omega_\mathrm{F}} \rangle$ to $\langle \Omega_\mathrm{F}, [\mathtt{Id}_{\Omega_\mathrm{F}}, \omega_\mathrm{F}^{-1}] \rangle$.

Vene and Uustalu derive this result by dualising their categorical description of course-of-value iteration. The duality can intuitively be understood as follows: Course-of-value iteration allows one to define a function's value on a given term using its value on sub-terms of arbitrary depth instead of accessing immediate sub-terms only. In this dual schema we are allowed to specify arguments for the construction of remaining parts of a behaviour at stages arbitrarily further out in the future instead of being forced to give them for the following stage already.

At the same time we get a corresponding notion of a λ-bisimulation. Intuitively, a relation B is a λ-bisimulation for λ as above, if for every pair of related states their behaviour can be split into a common initial part, which has to be nonempty and finite, and continuations generated by successors which are related by B again. We call such a relation a *multi-step bisimulation* because it may take several equivalent steps from two related states until one encounters successors which are related again. Although this seems to be a natural generalisation of the conventional notion of a bisimulation, we have could not find it in the literature yet.

7 Example: λ-Coiteration and Terms

In this section we will again consider specifications involving operators in the definition of the dynamics of a state. We have already encountered such an example in Section 3.1 when we defined the multiplication of power series using sums and in Section 5.1 where the encoding of infinite binary trees into streams was specified using an interleaving operation. The simple treatment employing $\mathrm{T}X := X \times X$ was possible in both cases because of the very regular occurrence of the operator inside the specification. The tail of $\sigma \otimes \tau$ for instance could always be characterised as the sum of two products of streams. Here we will give a schema that allows an arbitrary use of possibly several operators. As always, we start with a concrete example.

7.1 The Stream of Hamming Numbers

Taking up an example from Dijkstra's [Dij81] (also treated by Sijtsma [Sij89]), we consider the sorted stream $\mathtt{ham} \in \mathbb{N}^\omega$ of all Hamming Numbers in a simplified version, namely as the natural numbers whose only prime factors are 2 and 3. Streams of natural numbers will again be regarded as given by the final $\mathrm{S}_{\mathbb{N}}$-coalgebra for $\mathrm{S}_\mathbf{N} X := \mathbb{N} \times X$. We will concentrate on the following

[6] We can again get rid of this assumption using a variant of Theorem 3.10, since Δ can be shown to be a monad.

specification using the auxiliary operators $\mathtt{merge} : I\!N^\omega \times I\!N^\omega \to I\!N^\omega$ and $\mathtt{map}_g : I\!N^\omega \to I\!N^\omega$ for $g : I\!N \to I\!N$:

$$\begin{aligned}
\langle \mathtt{head}, \mathtt{tail}\rangle(\mathtt{ham}) &= \langle 1, \mathtt{merge}(\mathtt{map}_{*2}(\mathtt{ham}), \mathtt{map}_{*3}(\mathtt{ham}))\rangle \\
\langle \mathtt{head}, \mathtt{tail}\rangle(\mathtt{merge}(\sigma, \tau)) &= \begin{cases} \langle \sigma_0, \mathtt{merge}(\sigma', \tau)\rangle & \text{if } \sigma_0 < \tau_0 \\ \langle \sigma_0, \mathtt{merge}(\sigma', \tau')\rangle & \text{if } \sigma_0 = \tau_0 \\ \langle \tau_0, \mathtt{merge}(\sigma, \tau')\rangle & \text{if } \sigma_0 > \tau_0 \end{cases} \\
\langle \mathtt{head}, \mathtt{tail}\rangle(\mathtt{map}_g(\sigma)) &= \langle g(\sigma_0), \mathtt{map}_g(\sigma')\rangle
\end{aligned}$$

for $\sigma = \langle \sigma_0 : \sigma' \rangle, \tau = \langle \tau_0 : \tau' \rangle \in I\!N^\omega$ and the functions $*2$ and $*3$ that double and triple their arguments.

The idea is to view the stream $\mathtt{ham}$ as a function $\mathtt{ham} : 1 \to I\!N^\omega$ and to define it as the λ-coiterative arrow induced by some $\phi : 1 \to \mathrm{S_N}\,\mathrm{T}1$. Here $\mathrm{T}X$ is the set of terms freely generated by $\mathtt{merge}$ and $\mathtt{map}_g$ over X. We will develop this schema in a more general setting in the remainder of this section.

7.2 The λ-Coiteration Schema on Terms

Assume we are given a signature $\langle \Sigma, ar\rangle$, i.e. a set Σ of operator symbols coming with an arity assignment $ar : \Sigma \to I\!N$. Let $\mathrm{T}X$ denote the set of terms freely generated by this signature over X. We will sometimes call the set X a set of *variables* in this context.

A term just consisting of the variable $x \in X$ will be written as $\underline{x} \in \mathrm{T}X$, and the injection function $x \mapsto \underline{x}$ will be denoted by $\eta_X : X \to \mathrm{T}X$. A term that has again terms as variables can be flattened by the operation $\mu_X : \mathrm{T}^2X \to \mathrm{T}X$, inductively defined as

$$\begin{aligned}
\mu_X(\underline{t}) &:= t \\
\mu_X(op(tt_1, \ldots, tt_{ar(op)})) &:= op(\mu_X(tt_1), \ldots, \mu_X(tt_{ar(op)}))
\end{aligned}$$

for $t \in \mathrm{T}X$, $op \in \Sigma$, and $tt_i \in \mathrm{T}^2X$.

We can extend the mapping $X \mapsto \mathrm{T}X$ to a functor by declaring its operation on a function $f : X \to Y$ to be the application of f to all its variables, i.e.

$$\begin{aligned}
(\mathrm{T}f)(\underline{x}) &:= \underline{f(x)}, \\
(\mathrm{T}f)(op(t_1, \ldots, t_{ar(op)})) &:= op((\mathrm{T}f)(t_1), \ldots, (\mathrm{T}f)(t_{ar(op)}))
\end{aligned}$$

for each $op \in \Sigma$. Note that this definition makes both – the injection of variables and the flattening – natural transformations: [7] $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$ and $\mu : \mathrm{T}^2 \Rightarrow \mathrm{T}$.

To use the functor T within λ-coiteration, we need to specify its interaction with F, i.e. a distributive law $\lambda : \mathrm{TF} \Rightarrow \mathrm{FT}$ or, as in Section 5, $\lambda : \mathrm{T}(\mathrm{Id} \times \mathrm{F}) \Rightarrow \mathrm{FT}$. We will concentrate on the latter type, since many examples in this context require the extended expressibility it offers.

Assume we are given such a distributive law λ. Let $\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ be a final F-coalgebra and let $\llbracket . \rrbracket$ denote the coiterative morphism from $\mathrm{T}_\lambda\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$ to $\langle \Omega_\mathrm{F}, \omega_\mathrm{F}\rangle$. By applying Theorem 3.9 we get that for every operation $\phi : X \to \mathrm{FT}X$ there is a unique arrow $f : X \to \Omega_\mathrm{F}$ fitting into the diagram below

$$\begin{tikzcd}
 & \mathrm{T}\,\Omega_\mathrm{F} \arrow[d, "\llbracket \cdot \rrbracket"] \\
X \arrow[r, dashed, "\exists! f"] \arrow[d, "\forall\phi"'] & \Omega_\mathrm{F} \arrow[d, "\omega_\mathrm{F}"] \\
\mathrm{FT}X \arrow[r, dashed, "\mathrm{F}\, f|\llbracket \cdot \rrbracket"'] & \mathrm{F}\,\Omega_\mathrm{F}
\end{tikzcd}$$

[7] Taken together this defines the familiar term monad $\langle \mathrm{T}, \eta, \mu\rangle$.

The above statement is not fully satisfactory, since it is stated referring to an evaluation of terms $[\![.]\!]$ of which we do not know much yet. The property we would like it to have is *compositionality.* Because if this is the case, then every syntactical operation $op \in \Sigma$, $ar(op) = n$ represents an operation $op^* : \Omega_{\mathrm{F}}^n \to \Omega_{\mathrm{F}}$ such that

$$[\![op(t_1, \ldots, t_n)]\!] = op^*([\![t_1]\!], \ldots, [\![t_n]\!])$$

and the characterisation of f from above is turned into one based on these "real" operators.

Following Turi and Plotkin [TP97], this holds in case the distributive law is compositional itself (i.e. it satisfies the assumption made in Theorem 3.10). The latter condition is equivalent to saying that λ is constructed from a semantics of the single operator symbols in the shape of a natural transformation

$$\rho^{op} : (\mathrm{Id} \times \mathrm{F})^n \Rightarrow \mathrm{F}\,\mathrm{T} \tag{6}$$

for each $op \in \Sigma$ with $ar(op) = n$.

Formally, the distributive law λ is derived from these operator specifications as follows:

$$\begin{array}{rcl} \lambda_X(\underline{\langle x, fx\rangle}) & := & (\mathrm{F}\eta_X)(fx) \\ \lambda_X(op(s_1, \ldots, s_n)) & := & \varrho_X^{op}(\langle \mathrm{T}\pi_1, \lambda_X\rangle(s_1), \ldots, \langle \mathrm{T}\pi_1, \lambda_X\rangle(s_n)) \end{array}$$

for $x \in X$, $fx \in \mathrm{F}X$, $op \in \Sigma$ with $ar(op) = n$, and $s_i \in \mathrm{T}(X \times \mathrm{F}X)$, using the abbreviation

$$\begin{array}{rcl} \varrho^{op} & : & (\mathrm{T} \times \mathrm{F}\,\mathrm{T})^n \Rightarrow \mathrm{F}\,\mathrm{T} \\ \varrho_X^{op} & := & \mathrm{F}\mu_X \circ \rho_{\mathrm{T}X}^{op}. \end{array}$$

A specification of the format in (6) determines one F-step for any term $op(\underline{x_1}, \ldots, \underline{x_n})$ of depth one with $x_i \in X$ as $\rho_X^{op}(\langle x_1, d_1\rangle, \ldots, \langle x_n, d_n\rangle) \in \mathrm{F}\,\mathrm{T}X$ given that $d_i \in \mathrm{F}X$ is the F-step of the variable x_i from the i-th argument. The role of the assumption on ρ^{op} to be a natural transformation is actually to restrict the access to the x_i and their direct successors possibly occurring inside the d_i. They can just be used as references, no further checks can be made on them.

To determine the F-steps of a term $op(t_1, \ldots, t_n) \in \mathrm{T}X$ of greater depth, one recursively computes those of the sub-terms t_i and then applies $\rho_{\mathrm{T}X}^{op}$, i.e. the semantics of op treating the t_i as variables. This results in an element from $\mathrm{F}\,\mathrm{T}^2X$, but by flattening the terms we end up in the desired set.

Without proof we state that for a compositional distributive law we obtain a compositional evaluation of terms on the final F-coalgebra.

Theorem 7.1 *Let a functor* $\mathrm{F} : \mathsf{Set} \to \mathsf{Set}$ *with a final coalgebra* $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ *and a signature* $\langle \Sigma, ar\rangle$ *with a natural transformation* ρ^{op} *as in (6) for every* $op \in \Sigma$ *be given. Then the coiterative morphism*

$$[\![.]\!] : \mathrm{T}_\lambda \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$$

for λ *as above is compositional, i.e. we can derive from it a set of operators*

$$op^* : \Omega_{\mathrm{F}}^{ar(op)} \to \Omega_{\mathrm{F}} \quad \textit{for} \quad op \in \Sigma$$

such that $[\![.]\!] = [\![.]\!]^*$ *where*

$$\begin{array}{rcl} [\![\underline{\sigma}]\!]^* & := & \sigma \\ [\![op(t_1, \ldots, t_{ar(op)})]\!]^* & := & op^*([\![t_1]\!]^*, \ldots, [\![t_{ar(op)}]\!]^*) \end{array}$$

where $\sigma \in \Omega_{\mathrm{F}}$ *and* $t_i \in \mathrm{T}\Omega_{\mathrm{F}}$. *Furthermore, the operators are the unique ones fitting for every* $op \in \Sigma$ *with* $ar(op) = n$ *into the following diagram:*

$$\begin{tikzcd}
 & \Omega_{\mathrm{F}}^n \arrow[dl, "\langle \mathtt{Id}, \omega_{\mathrm{F}}\rangle^n"'] \arrow[dd, "op^*"] \\
(\Omega_{\mathrm{F}} \times \mathrm{F}\,\Omega_{\mathrm{F}})^n \arrow[dd, "\rho^{op}_{\Omega_{\mathrm{F}}}"'] & \\
 & \Omega_{\mathrm{F}} \arrow[dd, "\omega_{\mathrm{F}}"] \\
\mathrm{F}\,\mathrm{T}\,\Omega_{\mathrm{F}} \arrow[dr, "\mathrm{F}\,[\![\cdot]\!]^*"'] & \\
 & \mathrm{F}\,\Omega_{\mathrm{F}}
\end{tikzcd}$$

Turi and Plotkin [TP97] show that specifications of operators by means of natural transformations ρ^{op} as in (6) are closely related to those by means of structural transition rules in GSOS format [BIM95]. We will show below how `merge` and $\mathtt{map}_g$ from above can be captured. Many important operators are definable this way. The operators on formal power series Rutten considers for his stream calculus [Rut00a], of which we have seen the sum and convolution product already, all fit into this framework. Other examples are regular operators on automata [Rut98a] and many operators used in Process Algebra, like parallel composition (see e.g. [BW90, Fok00] for introductions).

Again we take a look at the proof principle that we get for this setting of T and λ. For simplicity we will concentrate on λ-bisimulations on the final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$. Let again op^* for $op \in \Sigma$ denote the decomposition of $[\![.]\!] : \mathrm{T}_\lambda \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ from above. For a relation $B \subseteq \Omega_{\mathrm{F}} \times \Omega_{\mathrm{F}}$ we set B^* to be the congruence closure of B under op^* for $op \in \Sigma$, i.e. the smallest relation B^* containing B such that for all $op \in \Sigma$, $ar(op) = n$ and $\langle \sigma_i, \tau_i\rangle \in B^*$, $1 \leq i \leq n$ also $\langle op^*(\sigma_1, \ldots, \sigma_n), op^*(\tau_1, \ldots, \tau_n)\rangle \in B^*$. The relation B^* represents the span $\langle \mathrm{T}B, \pi_1|^{[\![\cdot]\!]}, \pi_2|^{[\![\cdot]\!]}\rangle$ according to Lemma 4.2, because by applying the closure condition several times one gets that the relation B^* contains a pair if it can be obtained by evaluating the same Σ-context (with multiple holes) where states are plugged into corresponding holes that are related by B.

We get that B is a λ-bisimulation on $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ if and only if there is an operation $\chi : B \to \mathrm{F}B^*$ making the diagram below commute (note again that we use different projections π_i in the upper and lower part of the diagram)

$$\begin{tikzcd}
\Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"'] & B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "\exists\chi"] & \Omega_{\mathrm{F}} \arrow[d, "\omega_{\mathrm{F}}"] \\
\mathrm{F}\,\Omega_{\mathrm{F}} & \mathrm{F}B^* \arrow[l, "\mathrm{F}\,\pi_1"] \arrow[r, "\mathrm{F}\,\pi_2"'] & \mathrm{F}\,\Omega_{\mathrm{F}}
\end{tikzcd}$$

The above characterisation of B^* explains that a relation satisfying this condition is sometimes called a *bisimulation up-to-context* [San98].

From Theorem 4.15 we get that a bisimulation up-to-context on the final F-coalgebra only relates identical states in case the contexts are built from a set of operators definable by a specification of the type (6) on page 38.

7.3 The Hamming Numbers Example revisited

Turning back to the example from Section 7.1, we will consider the signature $\langle \Sigma, ar\rangle$ with $\Sigma = \{\underline{\mathtt{merge}}\} \cup \{\underline{\mathtt{map}}_g \mid g : \mathbb{N} \to \mathbb{N}\}$ and $ar(\underline{\mathtt{merge}}) = 2$, $ar(\underline{\mathtt{map}}_g) = 1$. The definitions of the functions `merge` and $\mathtt{map}_g$ given there can be transformed into a semantics of the syntactic operators as follows

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

The naturality of these definitions is easily verified. Note that x_0 and y_0, on which the case distinction is made and to which the function g is applied, are natural numbers taken from the direct observations. Thus, this does not violate the above argument that the elements from X may only be used as a reference.

From Theorem 7.1 we get that the term evaluation $[\![.]\!]$ on the final coalgebra $\langle I\!N^{\omega}, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$ replaces the term constructors $\underline{\mathtt{merge}}$ and $\underline{\mathtt{map}}_g$ by the operations $\underline{\mathtt{merge}}^*$ and $\underline{\mathtt{map}}_g{}^*$. The characterisation of these functions at the end of the theorem can easily be seen to be equivalent to the definitions of $\mathtt{merge}$ and $\mathtt{map}_g$ in Section 7.1.

Now we can define the arrow $\mathtt{ham} : 1 \to I\!N^{\omega}$ as the λ-coiterative arrow induced by $\phi : 1 \to I\!N \times \mathrm{T}\,1$ with (recall that $1 = \{*\}$)

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

This yields a unique arrow satisfying

$$\begin{array}{rcl}
\langle \mathtt{head}, \mathtt{tail} \rangle(\mathtt{ham}(*)) & = & (\mathtt{Id} \times \mathtt{ham}|^{[\![\cdot]\!]})(\phi(*)) \\
& = & \langle 1, [\![\underline{\mathtt{merge}}(\underline{\mathtt{map}}_{*2}(\underline{\mathtt{ham}(*)}), \underline{\mathtt{map}}_{*3}(\underline{\mathtt{ham}(*)}))]\!] \rangle \\
& = & \langle 1, \mathtt{merge}([\![\underline{\mathtt{map}}_{*2}(\underline{\mathtt{ham}(*)})]\!], [\![\underline{\mathtt{map}}_{*3}(\underline{\mathtt{ham}(*)})]\!]) \rangle \\
& = & \langle 1, \mathtt{merge}(\mathtt{map}_{*2}([\![\underline{\mathtt{ham}(*)}]\!]), \mathtt{map}_{*3}([\![\underline{\mathtt{ham}(*)}]\!])) \rangle \\
& = & \langle 1, \mathtt{merge}(\mathtt{map}_{*2}(\mathtt{ham}(*)), \mathtt{map}_{*3}(\mathtt{ham}(*))) \rangle
\end{array}$$

as wanted.

We want to conclude the treatment of this example with two side remarks.

- The above argument establishes a unique solution of the specification of $\mathtt{ham}$. It does not formally show that the resulting stream indeed contains all Hamming Numbers in sorted order – although this may be clear by inspection. To formally do so, one could consider another more intuitive but less effective way to specify this stream, namely as the ordering of the set of Hamming Numbers, i.e. $\mathtt{sort}(H)$ where $H := \{2^m \cdot 3^n \mid m, n \in I\!N\}$ and $\mathtt{sort}$ is the function mapping an infinite set of natural numbers to the ordered stream of all its elements, easily definable by coiteration. One can show that the singleton relation $\{\langle H, \mathtt{ham}(*) \rangle\}$ is a λ-bisimulation (or bisimulation up-to-context) between the stream system defining $\mathtt{sort}$ and $\langle I\!N^{\omega}, \langle \mathtt{head}, \mathtt{tail} \rangle \rangle$, provided one includes only operators $\mathtt{map}_g$ for strictly monotonic g.

- A standard means to formalise properties of coalgebra states are *invariants*. The question arises how to prove invariant properties of the image of λ-coiterative morphisms. In the concrete example, one might want to prove that the stream $\mathtt{ham}(*)$ is strictly increasing (independent of the approach taken in the previous point). The set of all such streams can be specified as the greatest invariant $\overline{P}$ contained in the set $P \subseteq I\!N^{\omega}$, where P consists of all streams of which the first element is smaller than the second. That $\mathtt{ham}(*)$ is contained in $\overline{P}$ could be proved by showing that indeed it is in P and that P is closed under $\mathtt{merge}$ and $\mathtt{map}_g$, again for strictly monotonic g. One direction for future work would be to formulate general principles embodying this type of reasoning.

8 Related Work

Two known generalisations of the basic coiteration schema that we are aware of are the schemata that arise as the duals of primitive recursion and course-of-value iteration, as derived on a categorical level e.g. by Vene and Uustalu [VU97, UV99]. We have already treated these schemata in Sections 3.2 and 6. In both cases the universal characterisation Vene and Uustalu give can be derived from the one the corresponding instance of our theory yields.

Furthermore, Vene and Uustalu prove some laws to calculate with the arrows obtained. All of them easily follow from our treatment as well. As an example, they provide a *fusion law* for both schemata. In our setting it reads as follows: Given a functor T that distributes over F via λ and two

FT-coalgebras $\langle X, \phi_X\rangle$ and $\langle Y, \phi_Y\rangle$, the λ-coiterative arrow $f_X : X \to \Omega_{\mathrm{F}}$ induced by ϕ_X can be written as $f_Y \circ h$ if $h : X \to Y$ is an FT-homomorphism from $\langle X, \phi_X\rangle$ to $\langle Y, \phi_Y\rangle$ and $f_Y : Y \to \Omega_{\mathrm{F}}$ is the λ-coiterative arrow induced by ϕ_Y. It easily follows from $f_Y|^{\beta_{\mathrm{F}}} \circ \mathrm{FT}h = (f_Y \circ h)|^{\beta_{\mathrm{F}}}$ and the unique existence of λ-coiterative arrows.

The research presented here was influenced by the work of Lenisa [Len99a], who made a first step towards a generalised categorical description of extended coinduction schemata known in set theory. A central ingredient is her schema of *coiteration up-to-$\mathcal{T}$* for a pointed functor $\mathcal{T} = \langle \mathrm{T}, \eta\rangle$ – i.e. a functor $\mathrm{T} : \mathsf{C} \to \mathsf{C}$ with a natural transformation $\eta : \mathrm{Id} \Rightarrow \mathrm{T}$. Given an F-coalgebra $\langle \mathrm{T}X, \alpha\rangle$, it defines the arrow $f := h \circ \eta_X$ from the set X into the carrier of a final F-coalgebra $\langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$ as the composition of the unit at X and the coiterative morphism $h : \langle \mathrm{T}X, \alpha\rangle \to \langle \Omega_{\mathrm{F}}, \omega_{\mathrm{F}}\rangle$. For most of the examples that we presented here one can equip the functor T with a unit η such that the functions under consideration are $\mathcal{T}$-coiterative for a certain α. But the schema as such does not give a characterisation. Her theory only provides further information in case the pointed functor is taken from a monad $\mathcal{T} = \langle \mathrm{T}, \eta, \mu\rangle$ and distributes over the functor F via a distributive law λ. Then a proof principle applies which is based on her notion of a *bisimulation up-to-$\mathcal{T}$*.

In our approach the distributive law is already a part of the definition principle. It enables us to replace the operation $\alpha : \mathrm{T}X \to \mathrm{FT}X$ by the – to our impression – more natural $\phi : X \to \mathrm{FT}X$ leading to a characterisation of the arrow $f : X \to \Omega_{\mathrm{F}}$ as a unique one fitting into the λ-coiteration diagram. As a side effect, the need for additional structure built upon T can be dropped.

Lenisa's bisimulations up-to-$\mathcal{T}$ relate the states of two F-coalgebras of the shape $\langle \mathrm{T}X, \alpha_X\rangle$ and $\langle \mathrm{T}Y, \alpha_Y\rangle$ as they appear within coiteration up-to-$\mathcal{T}$. It turns out to be a special case of our notion of λ-bisimulation where the bialgebras are $\langle \mathrm{T}X, \mu_X, \alpha_X\rangle$ and $\langle \mathrm{T}Y, \mu_Y, \alpha_Y\rangle$. (Note that her definition does not assume these bialgebras to be λ-bialgebras. But an almost equivalent, though more technical assumption is made later in her main theorem [Len99a, Theorem 8 ii) assumption 3)] justifying the use of these relations.[8])

Our result is wider in scope in that it neither requires the functor T to be taken from a monad nor does it only apply to coalgebras where the carrier is of the shape $\mathrm{T}X$. In particular, it can directly be used to reason about equivalences on the final coalgebra, as it was done in the example from Section 4.2. Actually we even consider this to be its most important application.

Another paper that our work on bisimilarity proofs is related to is Sangiorgi's *bisimulation proof method* [San98]. It is about relational bisimulations for labeled transition systems in Set, but a reformulation in terms of spans between arbitrary F-coalgebras in an abstract category C should be possible (certainly requiring extra assumptions to be made for particular aspects of the theory to carry over). We will make our comparison with respect to this envisaged generalisation.

An important notion in his work is that of a progression: Given two F-coalgebras $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$ and two spans $\mathcal{R} = \langle R, r_1, r_2\rangle$ and $\mathcal{R}' = \langle R', r_1', r_2'\rangle$ on X and Y, we say that $\mathcal{R}$ *progresses* to $\mathcal{R}'$ and write $\mathcal{R} \rightarrowtail \mathcal{R}'$, if there exists a $\gamma : R \to \mathrm{F}R'$ making the diagram below commute:

$$\begin{tikzcd}
X \arrow[d, "\alpha_X"'] & R \arrow[l, "r_1"'] \arrow[r, "r_2"] \arrow[d, dashed, "\gamma"] & Y \arrow[d, "\alpha_Y"] \\
\mathrm{F}X & \mathrm{F}R' \arrow[l, "\mathrm{F}\,r_1'"] \arrow[r, "\mathrm{F}\,r_2'"'] & \mathrm{F}Y
\end{tikzcd}$$

With this terminology, a span is a bisimulation between the given coalgebras if and only if it progresses to itself.

Sangiorgi considers what he calls *sound* functions. These are functions G on spans between the carriers of $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$, such that for every span $\mathcal{R}$ it suffices to show $\mathcal{R} \rightarrowtail G(\mathcal{R})$ in order to conclude $\mathcal{R} \preceq \mathcal{B}$ for some bisimulation $\mathcal{B}$ between $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$. He defines a class of *respectful* functions on spans and shows that it contains interesting examples and has desirable closure properties (for the latter to carry over to our more general case one needs to make assumptions on C and F). His main theorem states that every respectful function is sound.

[8] Take a look at [LPW00] for a clarified version of its proof.

To apply Sangiorgi's terminology in our setting of a functor T that distributes over F via λ, we could consider the following function G on spans between X and Y: Given two T-algebra operations β_X and β_Y on X and Y respectively, we define

$$G(\langle R, r_1, r_2\rangle) := \langle \mathrm{T}R, r_1|^{\beta_X}, r_2|^{\beta_Y}\rangle.$$

In case $\langle X, \beta_X, \alpha_X\rangle$ and $\langle Y, \beta_Y, \alpha_Y\rangle$ are λ-bialgebras, a span $\mathcal{B}$ is a λ-bisimulation between $\langle X, \alpha_X\rangle$ and $\langle Y, \alpha_Y\rangle$ with regard to β_X and β_Y if and only if $\mathcal{B} \rightarrowtail G(\mathcal{B})$. It turns out that under the same assumption we can prove that G is respectful. Thus, an alternative way to obtain Theorem 4.15 would be to first generalise Sangiorgi's theorem and then get our result via the instantiation from above. Seen that way we proved a class of functions respectful, including some of those already considered by Sangiorgi, like the contextual closure operator.

Our use of a distributive law λ and λ-bialgebras follows that of Turi and Plotkin [TP97]. Starting from a signature functor Σ and a behavioural functor F, they describe program syntax using the term monad $\langle \mathrm{T}, \eta, \mu\rangle$ generated by Σ and global program behaviour by means of the behavioural comonad $\langle \mathrm{D}, \varepsilon, \delta\rangle$ generated by F. The specification of the semantics takes the shape of a distributive law λ of the monad over the comonad and its models are the λ-bialgebras. As in our treatment, the final F-coalgebra gives rise to a final λ-bialgebra and, dually, the initial Σ-algebra can be turned into an initial λ-bialgebra. These are taken as the canonical denotational and operational model of the specification.

In the example of corecursive definitions using operators from Section 7 our theory gets related to their setting, in that the functor T is taken from the term monad as well and that the class of well behaved operators here coincides with the main class considered by Turi and Plotkin. We add to their approach a separate treatment of specifications of the type $\phi : X \to \mathrm{F}\mathrm{T}X$, which are similar to what is sometimes called *guarded recursive definitions* in this context. Alternatively those could be dealt with by adding the set X as a set of constants to the signature Σ and extending λ according to ϕ. To define e.g. the stream of Hamming Numbers from Section 7.1, one would include `ham` as a constant in the signature and extend the construction of λ by

$$\rho_X^{\mathtt{ham}} := * \mapsto \langle 1, \underline{\mathtt{merge}}(\underline{\mathtt{map}}_{*2}(\mathtt{ham}), \underline{\mathtt{map}}_{*3}(\mathtt{ham}))\rangle : 1 \to \mathrm{F}\mathrm{T}X.$$

One of the advantages of treating the operators and solutions of the recursive specifications separately is that we obtain a notion of bisimulation up-to-context to reason about these solutions, which is not available in the setting of Turi and Plotkin.

The setting of Turi and Plotkin is more general in the sense that they consider distributive laws of the term monad over the behavioural comonad $\langle \mathrm{D}, \varepsilon, \delta\rangle$ generated by F instead of F alone. This allows them to show that a second class of operators is well-behaved, namely those definable by natural transformations of the type below, which is dual to those of type (6) on page 38:

$$\rho^{op} : D^n \Rightarrow \mathrm{F}\mathrm{T}^{\leq 1} \tag{7}$$

for $op \in \Sigma$, $ar(op) = n$, and $\mathrm{T}^{\leq 1}$ mapping a set of variables to the set of Σ-terms over those variables of depth at most 1. That this dual class may be considered in Turi and Plotkin's approach as well is not surprising since algebras and coalgebras play symmetric roles there. This is not true for our setting. Here the construction does not work for the second class of operators, which is in agreement with the fact that those may not safely be used for bisimulation up-to-context. (Note that the alternative approach indicated above would not work either, since the specification of constants X allowed by (7) does not correspond to $\phi : X \to \mathrm{F}\mathrm{T}X$.) The classes of operators fitting in Turi and Plotkin's approach are related to those for which bisimulation is a congruence. Here it is rather the question whether the operators may be used for bisimulation up-to-context. Sangiorgi has already pointed out that the former condition is not sufficient for the latter (see the example at the end of Section 2 in [San98]), which nicely explains the difference encountered here.

9 Conclusion

In this paper we presented a new general categorical framework for extended coinduction definition and proof principles.

For two functors $\mathrm{F}, \mathrm{T} : \mathsf{C} \to \mathsf{C}$ such that F has a final coalgebra $\langle \Omega_\mathrm{F}, \omega_\mathrm{F} \rangle$ and T distributes over F via a natural transformation λ we defined the notion of a λ-coiterative morphism $f : X \to \Omega_\mathrm{F}$ induced by an operation $\phi : X \to \mathrm{FT}X$. The occurrence of $\mathrm{T}X$ in the description of the codomain of ϕ replaces the use of X in the basic coiteration schema and serves as a more expressive set to point to for the next stage of the behaviour to be defined. The role of the distributive law λ is to lift the specification from X to this set.

We gave sufficient conditions for the unique existence of λ-coiterative arrows for every FT-operation ϕ. One approach assumes the existence of countable coproducts in the category C, for the other the functor T has to be taken from a monad and λ should be a distributive law of that monad over F.

Schemata arising as instances of this framework yield characterisations of many interesting functions into the carrier of a final coalgebra that cannot be captured directly by its finality.

As examples for already known principles that can be stated as λ-coiteration schemata (besides coiteration itself) we have treated primitive corecursion and the categorical dual of course-of-value iteration. We briefly mentioned that one can obtain a justification for unique solutions of behavioural differential equations as treated by Rutten [Rut00a] using operators like addition and multiplication of formal power series or sequential and parallel composition of transition systems.

Furthermore we investigated the usability of the above specification technique for proof purposes. This lead to the introduction of the notion of a λ-bisimulation. The same conditions as above are sufficient to show that all states related by a λ-bisimulation between certain F-coalgebras are bisimilar. We have shown that these relations can often be far simpler than the standard bisimulations needed otherwise.

In our examples we demonstrated that the notion of a λ-bisimulation specialises to bisimulation up-to-equality, a notion that one might call a multi-step bisimulation, and bisimulation up-to-context (see e.g. [San98]).

We view our presentation as an advancement of Lenisa's framework based on coiteration- and coinduction up-to-$\mathcal{T}$ [Len99a] which we consider as a first step towards a categorical description of generalised coinduction principles. Compared to her work we have emphasised the role of the distributive law appearing there as well, which allowed us to give a characterisation of the functions defined instead of a construction. Furthermore we provided a simpler and more widely applicable proof principle and added detailed examples.

We have shown that from the point of view of Sangiorgi's Bisimulation Proof Method [San98], every instance of our schema yields at a categorical level what he calls a respectful function in the set-theoretic context of labeled transition systems.

We left for future work the quest for further interesting instances of our framework. Since we found sufficient conditions for our schema to work that do not assume the functor T to come as a pointed functor or monad, it would be particularly nice to come up with examples exploiting this generality over alternative approaches in the literature. In all the examples we have so far this structure can be added immediately or at least after a straightforward reformulation of the problem.

As another interesting point we would like to study the relation between λ-coiterative arrows and invariant properties.

Furthermore, one could figure out the details of the categorical reformulation of Sangiorgi's technique envisaged in the previous section and state the relation to our work indicated there precisely. The possible gain of this effort would be that one could combine our technique with others from this setting – provided this class of respectful functions can be shown to have the closure properties required for this combination given the particular choice of C and F. This may lead to even more powerful coinduction principles, like a combination of bisimulation up-to-context and bisimulation up-to-bisimilarity [Mil89] that one often needs when one does not work on a final coalgebra. The latter cannot be covered by our schema.

Acknowledgements

I would like to thank my colleagues, in particular Jan Rutten, Alexandru Baltag, and Alexander Kurz, for intensive discussions, guidance and proof reading. I am further grateful to Dirk Pattinson for his comments. Special thanks go to Matteo Coccia, who raised my interest in Category Theory during his stay at the CWI.

References

[AM89] Peter Aczel and Nax Mendler. A final coalgebra theorem. In D.H. Pitt, D.E. Rydeheard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, *Proceedings 3rd Conf. on Category Theory and Computer Science, CTCS'89, Manchester, UK, 5–8 Sept 1989*, volume 389 of *Lecture Notes in Computer Science*, pages 357–365. Springer-Verlag, Berlin, 1989.

[BIM95] Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, January 1995.

[BW90] J.C.M. Baeten and W.P. Weijand. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, England, 1990.

[Dij81] E.W. Dijkstra. Hamming's exercise in sasl. Personal Note EWD792, see `http://www.cs.utexas.edu/users/EWD/`, 1981.

[Fok00] Wan Fokkink. *Introduction to Process Algebra*. Springer-Verlag, Berlin, 2000.

[JR96] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the EATCS*, 62:222–259, 1996.

[Len99a] M. Lenisa. From set-theoretic coinduction to coalgebraic coinduction: Some results, some problems. Unpublished extension of [Len99b] (available through the author's homepage at `http://www.dimi.uniud.it/~lenisa/`), December 1999.

[Len99b] M. Lenisa. From set-theoretic coinduction to coalgebraic coinduction: some results, some problems. In B. Jacobs and J. Rutten, editors, *Proceedings 2nd Int. Workshop on Coalgebraic Methods in Computer Science, CMCS'99, Amsterdam, The Netherlands, 20–21 March 1999*, volume 19 of *Electronic Notes in Theoretical Computer Science*, pages 1–21, 1999.

[LPW00] M. Lenisa, J. Power, and H. Watanabe. Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads. In H. Reichel, editor, *Proc. Coalgebraic Methods in Computer Science (CMCS 2000)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 233–263, 2000.

[McI99] M.D. McIlroy. Functional pearl: Power series, power serious. *J. of Functional Programming*, 9:323–335, 1999.

[Mil89] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.

[Rut98a] J.J.M.M. Rutten. Automata and coinduction (an exercise in coalgebra). Technical Report SEN-R9803, CWI - Centrum voor Wiskunde en Informatica, December 31, 1998. Also appeared as [Rut98b].

[Rut98b] J.J.M.M. Rutten. Automata and coinduction (an exercise in coalgebra). In *CONCUR'98: 9th International Conference on Concurrency Theory*, volume 1466 of *LNCS*, pages 194–218. Springer-Verlag, 1998.

[Rut00a] J.J.M.M. Rutten. Behavioural differential equations: a coinductive calculus of streams, automata, and power series. Technical Report SEN-R0023, CWI - Centrum voor Wiskunde en Informatica, 2000.

[Rut00b] J.J.M.M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249(1):3–80, October 2000.

[San98] D. Sangiorgi. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science*, 8:447–479, 1998.

[Sij89] Ben A. Sijtsma. On the Productivity of Recursive List Definitions. *ACM Transactions on Programming Languages and Systems*, 11(4):633–649, October 1989.

[TP97] D. Turi and G.D. Plotkin. Towards a mathematical operational semantics. In *Proc. 12*th *LICS Conf.*, pages 280–291. IEEE, Computer Society Press, 1997.

[UV99] Tarmo Uustalu and Varmo Vene. Primitive (co)recursion and course-of-value (co)iteration, categorically. *INFORMATICA (IMI, Lithuania)*, 10(1):5–26, 1999.

[VU97] Varmo Vene and Tarmo Uustalu. Functional programming with apomorphisms (Corecursion). In *9th Nordic Workshop on Programming Theory, NWPT'97*, Oct 1997.