Math. Struct. in Comp. Science (2011), *vol.* 21, *pp.* 321–361. © Cambridge University Press 2011
doi:10.1017/S096012951000054X

Pointwise extensions of GSOS-defined operations

HELLE HVID HANSEN[†] and BARTEK KLIN[‡§]

[†]*Technische Universiteit Eindhoven and Centrum Wiskunde & Informatica, P.O.Box 513, 5300 MB Eindhoven, The Netherlands*
Email: h.h.hansen@tue.nl
[‡]*University of Cambridge and University of Warsaw, 15 JJ Thomson Avenue, Cambridge CB3 0FD, U.K.*
Email: klin@mimuw.edu.pl

Received 15 April 2010; revised 15 September 2010

Final coalgebras capture system behaviours such as streams, infinite trees and processes. Algebraic operations on a final coalgebra can be defined by distributive laws (of a syntax functor Σ over a behaviour functor F). Such distributive laws correspond to abstract specification formats. One such format is a generalisation of the GSOS rules known from structural operational semantics of processes. We show that given an abstract GSOS specification ρ that defines operations σ on a final F-coalgebra, we can systematically construct a GSOS specification $\overline{\rho}$ that defines the pointwise extension $\overline{\sigma}$ of σ on a final F^A-coalgebra. The construction relies on the addition of a family of auxiliary 'buffer' operations to the syntax. These buffer operations depend only on A, so the construction is uniform for all σ and F.

1. Introduction

In coalgebra, state-based systems are modelled as F-coalgebras where F is a functor that determines the system type. By varying F, we obtain (non-)deterministic automata, (labelled) transition systems and many others – see Rutten (2000) for an introduction and plenty of examples. Of particular importance are *final* coalgebras, which represent abstract behaviours of F-coalgebras. For this reason we refer to elements of a final F-coalgebra as F-behaviours. Examples of F-behaviours include streams, infinite trees, causal stream functions and processes. Thanks to the properties of final coalgebras, operations on them can be conveniently defined by *coinduction*.

In this paper, we focus on pointwise extensions of operations on F-behaviours to operations on F^A-behaviours, where F is an arbitrary functor and A is a fixed set. Intuitively, F^A-coalgebras behave as F-coalgebras, but are additionally dependent on an external source of input from the alphabet A. For example, if $FX = B \times X$, then F^A-coalgebras are Mealy machines with input in A and output in B, and F^A-behaviours are causal stream functions $f : A^\omega \to B^\omega$. We show that, in general, elements of the set $\overline{Z}$ of F^A-behaviours can be thought of as certain functions from A^ω to a final F-coalgebra Z. An operation σ on Z can therefore be pointwise extended in the standard way to an

§ Bartek Klin was supported by EPSRC grant EP/F042337/1.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

operation $\overline{\sigma}$ on the function space Z^{A^ω}. Depending on σ, the operation $\overline{\sigma}$ restricts or not to an operation on $\overline{Z}$.

A well-structured way of defining operations on final coalgebras is by means of distributive laws of syntax over behaviour (Turi and Plotkin 1997; Bartels 2004; Klin 2009), where syntax is given by an algebraic signature Σ and behaviour is given by a functor F. The rather abstract notion of a distributive law can often be formulated in a more intuitive fashion as a set of equations or rules. Concrete examples include Rutten's behavioural differential equations (see, for example, Rutten (2003)) and rules of structural operational semantics of processes (Aceto *et al.* 2001).

Distributive laws provide a setting in which specification formats for varying syntax and behaviour types can be treated uniformly (using parametricity in Σ and F). In particular, the GSOS format of processes (Aceto *et al.* 2001) generalises to an abstract GSOS format for arbitrary Σ and F. A further benefit of working in this more abstract setting is that a distributive law of Σ over F not only defines Σ-operations on F-behaviours, but also provides an inductive definition of an F-coalgebra on Σ-terms, and relates the final semantics of the latter to the initial semantics of the former.

Viewing specification formats as distributive laws, it makes sense to ask whether given an operation σ, defined on a final F-coalgebra by a distributive law λ, its pointwise extension $\overline{\sigma}$ is an operation on a final F^A-coalgebra defined by another distributive law $\overline{\lambda}$, and can we obtain $\overline{\lambda}$ from λ in a systematic manner? The main technical contribution of the current paper is an affirmative answer to these questions under certain conditions. In Theorem 6.1, we show that definitions in a simple format can be extended straightforwardly. Then, in Theorem 6.3, we deal with the more expressive, abstract GSOS format, where the situation is more subtle. We show that GSOS-defined operations can also be extended pointwise, but the extension relies on a family of auxiliary operations that work intuitively as 'input buffers' for F^A-coalgebras. It is worth noting that the choice of auxiliary operators only depends on the set A, and does not depend on λ, or even on the behaviour functor F. In order to illustrate these general results, we provide several detailed examples.

The structure of the paper is as follows. In Section 2, we recall some basic facts on **Set**-functors, algebras and coalgebras, together with a few examples of F-coalgebras that are used in the paper. Section 3 contains two simple examples of operations on streams, and their pointwise extensions to Mealy machines. The techniques used in these examples motivate, and, hopefully, provide some intuition about the abstract development of the subsequent sections. In Section 4, we show how to interpret elements of F^A-coalgebras as functions from A-streams to final F-coalgebras through their pointwise behaviour; based on this, we provide an abstract definition of pointwise extension of operations, and illustrate it with a few examples. In Section 5, we recall the distributive law approach to defining operations, as developed in Turi and Plotkin (1997) and Lenisa *et al.* (2004). Section 6 contains the two main technical theorems described above, followed by a few example applications in Section 7, and a brief discussion of directions for future work in Section 8. The proofs of the main theorems are given in an Appendix.

An extended abstract of this paper appeared as Hansen and Klin (2010).

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

2. Preliminaries

In this section we fix some notation and provide the basic definitions of algebras and coalgebras needed in this paper. We assume familiarity with the notions of category, functor and natural transformation; the notions of adjunction and monad are mentioned a few times, but a detailed understanding is not necessary to follow the main development. All functors considered in this paper are endofunctors on **Set**, the category of sets and functions.

2.1. *Strength and costrength*

In this section, we recall some basic properties of **Set** functors that will be useful later in the paper.

For any set A, there is an adjunction $A \times - \dashv (-)^A$. The obvious unit and the counit of this adjunction will be denoted by

$$\eta_X : X \to (A \times X)^A$$
$$\epsilon_X : A \times X^A \to X.$$

Any endofunctor F on **Set** has a *strength*, that is, a map

$$\mathrm{st}^F_{A,X} : A \times FX \to F(A \times X)$$

natural in A and X, defined by

$$\mathrm{st}^F_{A,X}(a,t) = (F\eta^a)(t)$$

where $\eta^a : X \to A \times X$ is given by $\eta^a(x) = (a,x)$. Dually, every F has a *costrength*

$$\mathrm{cs}^F_{A,X} : F(X^A) \to (FX)^A$$

natural in A and X and defined by

$$\mathrm{cs}^F_{A,X}(t)(a) = (F\epsilon^a)(t)$$

where $\epsilon^a : X^A \to X$ is given by $\epsilon^a(f) = f(a)$.

2.2. *Syntax through algebras*

An *algebraic signature* Σ consists of a collection of function symbols $\{\sigma_i \mid i \in I\}$, where each σ_i has an arity $n_i \in \mathbb{N}$, $i \in I$. A Σ-*algebra* with carrier set X is a map $\coprod_{i\in I} X^{n_i} \to X$, so we identify a signature Σ with the functor $\Sigma X = \coprod_{i\in I} X^{n_i}$. In general, given a functor G, a G-*algebra* is a pair $\langle X, \sigma\rangle$ where X is the carrier set and $\sigma : GX \to X$ is a function.

The set of Σ-*terms over a set (of variables)* X is denoted by $T_\Sigma X$; we shall omit the subscript in the following as there will be no risk of confusion. In fact, T is a functor, and, together with obvious natural transformations $\eta^T : Id \Longrightarrow T$ (the interpretation of variables as terms) and $\mu^T : TT \Longrightarrow T$ (glueing terms built of terms), it forms the so-called free monad over Σ. By structural induction on terms, any algebra $\sigma : \Sigma X \to X$ induces a function $\sigma^\sharp : TX \to X$ (that is, term interpretation in σ). The T-algebra $\sigma^\sharp$ satisfies the

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

axioms

$$\sigma^\sharp \circ \eta^T_X = \mathrm{id}_X$$
$$\sigma^\sharp \circ T\sigma^\sharp = \sigma^\sharp \circ \mu^T_X,$$

that is, $\sigma^\sharp$ is an *Eilenberg–Moore algebra* for the monad T. The construction of $\sigma^\sharp$ from σ provides a 1–1 correspondence between Σ-algebras and Eilenberg–Moore T-algebras.

2.3. *Behaviour through coalgebras*

Coalgebra provides a uniform framework for studying the behaviour of systems such as automata and labelled transition systems. We only provide the basic definitions here; for a more thorough introduction to the theory of coalgebra, see Rutten (2000). Formally, given a functor F, an *F-coalgebra* is a pair $\langle X, \xi\rangle$ where X is a set (called the carrier, or the set of states) and $\xi : X \to FX$ is a function (called the structure). Since $\xi : X \to FX$ implicitly defines the carrier X, we often refer to an F-coalgebra simply by its structure. Different types of systems emerge by varying F. We will provide some concrete examples in Section 2.4.

Behaviour preserving maps are formalised as the general notion of a morphism between F-coalgebras. An *F-coalgebra morphism* $h : \langle X_1, \xi_1\rangle \to \langle X_2, \xi_2\rangle$ is a function $h : X_1 \to X_2$ such that $\xi_2 \circ h = Fh \circ \xi_1$. F-coalgebras and F-coalgebra morphisms form a category $\mathsf{Coalg}(F)$. An abstract notion of behaviour is obtained through finality. An F-coalgebra $\langle Z, \zeta\rangle$ is *final* if for every F-coalgebra $\langle X, \xi\rangle$ there is a unique F-coalgebra morphism $beh_{\langle X,\xi\rangle}$ (called the *final map*) from $\langle X, \xi\rangle$ to $\langle Z, \zeta\rangle$. A final F-coalgebra $\langle Z, \zeta\rangle$ can be viewed as a system of behaviours, and we refer to the elements of Z as *F-behaviours*. By the so-called Lambek Lemma (Lambek 1968), the structure ζ of a final coalgebra is always an isomorphism. There may not always be a final F-coalgebra because of cardinality considerations (Aczel and Mendler 1989), but all functors considered in this paper do admit final coalgebras.

The existence and uniqueness of the final map give rise to a definition principle usually referred to as *coinduction*. We will use coinduction to define operations on the carriers of final coalgebras. Let F be a functor and $\langle Z, \zeta\rangle$ be the final F-coalgebra. By defining an F-coalgebra structure $\xi : Z \times Z \to F(Z \times Z)$, the final map $beh_\xi : Z \times Z \to Z$ defines a binary operation $\star$ on Z by coinduction. So the F-coalgebra structure ξ essentially specifies how $x \star y$ behaves for all $x, y \in Z$. More generally, for an algebraic signature Σ, an F-coalgebra structure on ΣZ induces a Σ-algebra on Z by coinduction as illustrated by

$$\begin{array}{ccc} \Sigma Z & \overset{\sigma}{\cdots\cdots\rightarrow} & Z \\ {\scriptstyle\xi}\downarrow & & \downarrow{\scriptstyle\zeta} \\ F\Sigma Z & \xrightarrow{F\sigma} & FZ \end{array} \tag{1}$$

We will show in Section 5 how certain natural transformations correspond to various kinds of specification ξ.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

2.4. *Examples*

We shall now provide a number of concrete examples of functors and their final coalgebras. All of these are well known, but we will present them in some detail as they will be used later to illustrate pointwise extensions.

Example 2.1 (Stream automata). Given a set A, a *stream automaton (with output in A)* is a coalgebra for the functor $A \times -$, that is, it is a function $\langle o, d\rangle : X \to A \times X$ that maps an $x \in X$ to an output value $o(x) \in A$ and a next state $d(x) \in X$. We will use the notation $x \xrightarrow{a} y$ to denote $o(x) = a$ and $d(x) = y$. The final $(A \times -)$-coalgebra is obtained as the set A^ω of streams over A together with the head and tail maps: $\langle hd, tl\rangle : A^\omega \to A \times A^\omega$. The $(A \times -)$-behaviour of a state x is the stream of outputs generated on transitions starting in x. We will use the following standard notation. A stream $\alpha \in A^\omega$ may be written $(\alpha(0), \alpha(1), \alpha(2), \ldots)$ or as an ω-regular expression over A. For $\alpha \in A^\omega$ and $n \in \mathbb{N}$, $\alpha\restriction_n = (\alpha(0), \ldots, \alpha(n))$ is the prefix of α of length $n+1$. For $\alpha \in A^\omega$ and $a \in A$, we use $a : \alpha$ to denote the stream $(a, \alpha(0), \alpha(1), \ldots)$.

Example 2.2 (Mealy machines). Given sets A and B, a $(B \times -)^A$-coalgebra $m : X \to (B \times X)^A$ is a *Mealy machine* with input in A and output in B: for each state $x \in X$, $m(x) = \langle o_x, d_x\rangle : A \to B \times X$ defines for each $a \in A$, the output $o_x(a)$ and the next state $d_x(a)$. We write $x \xrightarrow{a|b} y$ if $o_x(a) = b$ and $d_x(a) = y$. The behaviour of a state x is the input–output mapping computed by $m : X \to (B \times X)^A$ when starting in x. Formally, $beh(x) : A^\omega \to B^\omega$ maps $\alpha \in A^\omega$ to the stream $beh(x)(\alpha) \in B^\omega$ where for all $n \in \mathbb{N}$,

$$beh(x)(\alpha)(n) = o_{x_n}(\alpha(n)) \tag{2}$$

with

$$\begin{aligned} x_0 &:= x \\ x_{n+1} &:= d_{x_n}(\alpha(n)). \end{aligned}$$

It is clear from (2) that the nth element of $beh(x)(\alpha)$ depends only on $\alpha(0), \ldots, \alpha(n)$. A stream function $f : A^\omega \to B^\omega$ is *causal* if for all $n \in \mathbb{N}$ and all $\alpha, \beta \in A^\omega$,

$$\alpha\restriction_n = \beta\restriction_n \quad \Longrightarrow \quad f(\alpha)(n) = f(\beta)(n).$$

The set $\Gamma = \{f : A^\omega \to B^\omega \mid f \text{ is causal}\}$ carries a final Mealy structure, as shown in Rutten (2006). We will now briefly summarise the construction. Let $f \in \Gamma$ and $a \in A$. We write $f(a:-)$ for the stream function that maps $\alpha \in A^\omega$ to $f(a:\alpha) \in B^\omega$. The *initial output* $f[a]$ and the *stream function derivative* f_a of f on a are defined by

$$\begin{aligned} f[a] &:= hd \circ f(a:-) \in A^\omega \to B \\ f_a &:= tl \circ f(a:-) \in A^\omega \to B^\omega. \end{aligned} \tag{3}$$

Since f is causal, $f[a]$ is constant (so we can consider $f[a]$ to be an element of B) and f_a is causal, so by defining for all $f \in \Gamma$ and $a \in A$,

$$\gamma(f)(a) = \langle f[a], f_a\rangle,$$

γ is a map of type $\gamma : \Gamma \to (B \times \Gamma)^A$, and it can be shown that $\langle \Gamma, \gamma\rangle$ is a final Mealy machine.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

Example 2.3 (Partial maps). We denote the one-element set by $1 = \{\perp\}$. Using $\perp$ to represent the undefined value, a $(1 + -)$-coalgebra $\xi : X \to 1 + X$ can be viewed as a partial function on X, and we write $x \longrightarrow y$ if $\xi(x) = y$ (for $y \in 1 + X$). The final $(1 + -)$-coalgebra consists of the natural numbers extended with ω together with the predecessor function

$$\text{pred} : \mathbb{N} + \{\omega\} \to 1 + \mathbb{N} + \{\omega\}$$

where

$$\begin{aligned} \text{pred}(0) &= \perp \\ \text{pred}(\omega) &= \omega \\ \text{pred}(n) &= n - 1 \text{ for all } n \in \mathbb{N}_{>0}. \end{aligned}$$

The behaviour of a state x is the number of steps that can be made from x.

Example 2.4 (Partial DLTSs). A $(1 + -)^A$-coalgebra $t : X \to (1 + X)^A$ can be viewed as a partial, deterministic labelled transition system where the a-successor of a state x is given by $t(x)(a)$ when $t(x)(a) \neq \perp$. We write $x \xrightarrow{a} y$ if $t(x)(a) = y$ (for $y \in 1 + X$).

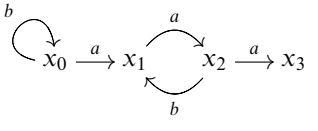
The final $(1+-)^A$-coalgebra $\langle \mathscr{L}, D \rangle$ (Rutten 1991) has as its carrier the set of non-empty, prefix-closed languages over A, that is,

$$\mathscr{L} = \{L \subseteq A^* \mid L \neq \varnothing; \text{for all } u, v \in A^* : uv \in L \Rightarrow u \in L\}.$$

For $L \in \mathscr{L}$, the function $D(L) : A \to 1 + \mathscr{L}$ is defined by

$$D(L)(a) = \begin{cases} L_a := \{w \in A^* \mid aw \in L\} & \text{if } a \in L \\ \perp & \text{if } a \notin L. \end{cases}$$

L_a is called the language derivative of L with respect to a. Note that if $a \in L$, then L_a is non-empty, and if L is prefix-closed, then L_a is prefix-closed, hence D is well defined. The behaviour of a state x is the set $L(x)$ of finite words that label some path in t starting in x. We can illustrate this with a small example:



$$\begin{aligned} L(x_0) &= b^* + b^*a(ab)^* + b^*a(ab)^*a + b^*a(ab)^*aa \\ L(x_1) &= (ab)^* + (ab)^*a + (ab)^*aa \\ L(x_2) &= (ba)^* + (ba)^*a + (ba)^*b \\ L(x_3) &= \{\varepsilon\}. \end{aligned}$$

Example 2.5 (Non-deterministic systems). We use $\mathcal{P}_\omega$ to denote the finitary (covariant) powerset functor that maps a set X to the set of all of its finite subsets. A $\mathcal{P}_\omega$-coalgebra $\xi : X \to \mathcal{P}_\omega X$ models a *finitely branching (non-deterministic) transition system*, and we write $x \longrightarrow y$ if $y \in \xi(x)$, and $x \not\longrightarrow$ if $\xi(x) = \varnothing$.

The final $\mathcal{P}_\omega$-coalgebra $\langle T, \tau \rangle$ is usually described as the set of strongly extensional, finitely branching trees together with the subtree relation (Worrell 2005). An alternative way of thinking about the $\mathcal{P}_\omega$-behaviour of a state x is as the bisimilarity quotient of the subsystem generated from x (Rutten and Turi 1994).

Example 2.6 (LTSs). A $(\mathcal{P}_\omega -)^A$-coalgebra $t : X \to (\mathcal{P}_\omega X)^A$ is an image-finite labelled transition system with state space X. We use the notation $x \xrightarrow{a} y$ if $y \in t(x)(a)$, and $x \not\xrightarrow{a}$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

if $t(x)(a) = \varnothing$, for all $x, y \in X$ and $a \in A$. The final $(\mathcal{P}_\omega -)^A$-coalgebra $\langle P, \phi \rangle$ is carried by the set P of image-finite *processes*, which can also be described as the $(\mathcal{P}_\omega -)^A$-bisimulation classes of all finitely branching, A-labelled trees with possibly infinite branches (Rutten and Turi 1994). The behaviour of a state x is thus the $(\mathcal{P}_\omega -)^A$-bisimulation quotient of the substructure generated from x.

3. Motivating examples

To motivate the abstract development of Sections 4 and 6, and provide some intuitions about the techniques used there, we shall begin by giving two concrete examples of operations on the set $\mathbb{R}^\omega$ of streams over the real numbers, that is, the carrier of a final $(\mathbb{R} \times -)$-coalgebra, and show how to extend their definitions to operations on causal stream functions.

As explained in Rutten (2003), operations on streams can be defined with the help of *stream differential equations*, where the resulting stream is unambiguously specified by its initial value (head) and stream derivative (tail) in terms of the initial values and the derivatives of the argument streams.

Example 3.1. The element-wise addition $\oplus$ of streams over $\mathbb{R}$ is defined by the stream differential equation:

$$\begin{aligned} hd(x_0 \oplus y_0) &= hd(x_0) + hd(y_0) \\ tl(x_0 \oplus y_0) &= tl(x_0) \oplus tl(y_0). \end{aligned}$$

The above stream differential equation can be formulated as a family of derivation rules:

$$\frac{x_0 \xrightarrow{a} x_1 \quad y_0 \xrightarrow{b} y_1}{x_0 \oplus y_0 \xrightarrow{a+b} x_1 \oplus y_1}$$

where a, b range over $\mathbb{R}$.

The operation $\oplus : \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ adds two streams of numbers element-wise:

$$\alpha \oplus \beta = (\alpha(0) + \beta(0), \alpha(1) + \beta(1), \alpha(2) + \beta(2), \ldots)$$

for all $\alpha, \beta \in \mathbb{R}^\omega$. The details of how this follows from the equations (or, equivalently, the rules) can be found in Rutten (2003). An alternative but equivalent explanation will be provided in Section 5, where the equations/rules are interpreted abstractly as distributive laws.

Here is another, deceptively similar, example.

Example 3.2. The operation $\boxplus : \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ is defined by

$$\begin{aligned} hd(x_0 \boxplus y_0) &= hd(x_0) + hd(y_0) \\ tl(x_0 \boxplus y_0) &= x_0 \boxplus tl(y_0) \end{aligned}$$

that is,

$$\frac{x_0 \xrightarrow{a} x_1 \quad y_0 \xrightarrow{b} y_1}{x_0 \boxplus y_0 \xrightarrow{a+b} x_0 \boxplus y_1} \tag{4}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

The operation $\boxplus$ adds the head element of its first argument to each of the elements of the second argument stream:

$$\alpha \boxplus \beta = (\alpha(0) + \beta(0), \alpha(0) + \beta(1), \alpha(0) + \beta(2), \ldots)$$

for all $\alpha, \beta \in \mathbb{R}^\omega$.

The operations $\oplus$ and $\boxplus$ can be extended pointwise to the set of causal stream functions $\Gamma_{\mathbb{R}} = \{f : \mathbb{R}^\omega \to \mathbb{R}^\omega \mid f \text{ is causal}\}$ by defining

$$\begin{aligned} (f \bar{\oplus} g)(\alpha) &= f(\alpha) \oplus g(\alpha) \\ (f \bar{\boxplus} g)(\alpha) &= f(\alpha) \boxplus g(\alpha) \end{aligned}$$

for all $f, g \in \Gamma_{\mathbb{R}}$ and $\alpha \in \mathbb{R}^\omega$. Note, in particular, that $f \bar{\oplus} g$ and $f \bar{\boxplus} g$ are again causal, so $\bar{\oplus}$ and $\bar{\boxplus}$ are well defined as operations on $\Gamma_{\mathbb{R}}$. In general, given an n-ary operation σ on a set Y, the *pointwise extension* of σ to the function space Y^X, is the operation $\bar{\sigma} : (Y^X)^n \to Y^X$ defined for all $f_0, \ldots, f_{n-1} : X \to Y$ by

$$\bar{\sigma}(f_0, \ldots, f_{n-1})(x) = \sigma(f_0(x), \ldots, f_{n-1}(x)).$$

The question is, can we give a rule-based definition of the pointwise extensions $\bar{\oplus}$ and $\bar{\boxplus}$, possibly based on the defining rules for $\oplus$ and $\boxplus$, respectively?

Recall from Example 2.2 that $\Gamma_{\mathbb{R}}$ is the final $(\mathbb{R} \times -)^{\mathbb{R}}$-coalgebra under the operations initial output $f[a]$ and stream function derivative f_a. Using the notation $f \xrightarrow{a|f[a]} f_a$, the pointwise extension $\bar{\oplus} : \Gamma_{\mathbb{R}} \times \Gamma_{\mathbb{R}} \to \Gamma_{\mathbb{R}}$ can be defined as follows:

$$\frac{x_0 \xrightarrow{a|b} x_1 \qquad y_0 \xrightarrow{a|c} y_1}{x_0 \bar{\oplus} y_0 \xrightarrow{a|b+c} x_1 \bar{\oplus} y_1} \tag{5}$$

To see how this rule works, suppose $\alpha = (a_0, a_1, a_2, \ldots) \in \mathbb{R}^\omega$ and $f, g \in \Gamma_{\mathbb{R}}$ with

$$\begin{aligned} f(\alpha) &= (b_0, b_1, b_2, \ldots) \\ g(\alpha) &= (c_0, c_1, c_2, \ldots). \end{aligned}$$

It follows, in particular, that for the repeated derivatives we have:

$$\begin{aligned} f_{a_0 \ldots a_n}[a_{n+1}] &= b_{n+1} \\ g_{a_0 \ldots a_n}[a_{n+1}] &= c_{n+1} \end{aligned}$$

for all $n \in \mathbb{N}$. The rule in (5) induces the following transitions of $f \bar{\oplus} g$ on input α:

$$f \bar{\oplus} g \xrightarrow{a_0|b_0+c_0} f_{a_0} \bar{\oplus} g_{a_0} \xrightarrow{a_1|b_1+c_1} f_{a_0 a_1} \bar{\oplus} g_{a_0 a_1} \xrightarrow{a_2|b_2+c_2} \ldots$$

and we see that

$$(f \bar{\oplus} g)(\alpha) = f(\alpha) \oplus g(\alpha).$$

Since α was arbitrary, $\bar{\oplus}$ is indeed the pointwise extension of $\oplus$.

Let us now consider the pointwise extension of $\boxplus$. By analogy with (5), we could try to define $\bar{\boxplus} : \Gamma_{\mathbb{R}} \times \Gamma_{\mathbb{R}} \to \Gamma_{\mathbb{R}}$ with the following rule:

$$\frac{x_0 \xrightarrow{a|b} x_1 \qquad y_0 \xrightarrow{a|c} y_1}{x_0 \bar{\boxplus} y_0 \xrightarrow{a|b+c} x_0 \bar{\boxplus} y_1} \tag{6}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

For $f, g \in \Gamma_{\mathbb{R}}$, the rule (6) implies that for all $a \in \mathbb{R}$,

$$\begin{array}{lcl}(f\bar{\boxplus}g)[a] & = & f[a] + g[a] \\ (f\bar{\boxplus}g)_a & = & f\bar{\boxplus}g_a.\end{array} \tag{7}$$

However, the induced operation on $\Gamma_{\mathbb{R}}$ is *not* the pointwise extension of $\boxplus$. To see this, consider $\mathrm{id}\bar{\boxplus}\mathrm{id}$ where id is the identity on $\mathbb{R}^\omega$. Note, in particular, that

$$\begin{aligned}\mathrm{id}[a] &= a \\ \mathrm{id}_a &= \mathrm{id}\end{aligned}$$

for all $a \in \mathbb{R}$.

The rule (6) then yields the following transitions on input $\alpha = (a_0, a_1, a_2, \ldots) \in \mathbb{R}^\omega$:

$$\mathrm{id}\bar{\boxplus}\mathrm{id} \xrightarrow{a_0|a_0+a_0} \mathrm{id}\bar{\boxplus}\mathrm{id} \xrightarrow{a_1|a_1+a_1} \mathrm{id}\bar{\boxplus}\mathrm{id} \xrightarrow{a_2|a_2+a_2} \ldots$$

so

$$(\mathrm{id}\bar{\boxplus}\mathrm{id})(\alpha) \neq \alpha \boxplus \alpha = (a_0 + a_0, a_0 + a_1, a_0 + a_2, \ldots).$$

To understand what goes wrong, we must look into the structure of the final Mealy machine. Recall from Example 2.2 that for all $f \in \Gamma_{\mathbb{R}}$ and $a \in \mathbb{R}$, we have $f_a = tl(f(a:-))$. In order for $\bar{\boxplus}$ to be the pointwise extension of $\boxplus$, the following must hold for all $f, g \in \Gamma_{\mathbb{R}}$, $a \in \mathbb{R}$ and $\alpha \in \mathbb{R}^\omega$:

$$(f\bar{\boxplus}g)(a:\alpha) = f(a:\alpha) \boxplus g(a:\alpha).$$

By taking tails on both sides, and applying the rule (4) for $\boxplus$, we get

$$(f\bar{\boxplus}g)_a(\alpha) = f(a:\alpha) \boxplus g_a(\alpha). \tag{8}$$

Comparing (8) with (7), we see that in the definition of $(f\bar{\boxplus}g)_a$, f should be replaced with the function $f(a:-)$. In order to give a rule-based definition of $\bar{\boxplus}$ (which applies not only to $\Gamma_{\mathbb{R}}$, but to all Mealy machines), it makes sense to use a family of auxiliary, unary operations $\{a \triangleright - \mid a \in \mathbb{R}\}$. These operations act like one-element input buffers: for a Mealy machine $m\colon X \to (\mathbb{R} \times X)^{\mathbb{R}}$ and $x \in X$, we have that $a \triangleright x$ behaves like x as if it had seen an a already. Formally, $beh(a \triangleright x)(\alpha) = beh(x)(a:\alpha)$ for all $\alpha \in \mathbb{R}^\omega$ and $a \in \mathbb{R}$. We can now define $\bar{\boxplus}$, together with the buffer operations, by the rules

$$\frac{x_0 \xrightarrow{a|b} x_1 \quad y_0 \xrightarrow{a|c} y_1}{x_0\bar{\boxplus}y_0 \xrightarrow{a|b+c} (a \triangleright x_0)\bar{\boxplus}y_1} \qquad \frac{x_0 \xrightarrow{a|b} x_1}{a \triangleright x_0 \xrightarrow{c|b} c \triangleright x_1} \tag{9}$$

To illustrate how this definition works, let $\alpha = (a_0, a_1, a_2, \ldots) \in \mathbb{R}^\omega$ and $f, g \in \Gamma_{\mathbb{R}}$ with

$$\begin{aligned}f(\alpha) &= (b_0, b_1, b_2, \ldots) \\ g(\alpha) &= (c_0, c_1, c_2, \ldots).\end{aligned}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

First, the rule for $a \triangleright -$ gives us,

$$\begin{aligned} f \xrightarrow{a_0|b_0} f_{a_0} \implies & (a_0 \triangleright f) \xrightarrow{a_1|b_0} (a_1 \triangleright f_{a_0}) \\ \implies & (a_1 \triangleright a_0 \triangleright f) \xrightarrow{a_2|b_0} (a_2 \triangleright a_1 \triangleright f_{a_0}) \\ \implies & (a_2 \triangleright a_1 \triangleright a_0 \triangleright f) \xrightarrow{a_3|b_0} (a_3 \triangleright a_2 \triangleright a_1 \triangleright f_{a_0}) \\ \implies & \dots \end{aligned}$$

We now find that,

$$\begin{aligned} f \bar{\boxplus} g \xrightarrow{a_0|b_0+c_0} & (a_0 \triangleright f) \bar{\boxplus} g_{a_0} \\ \xrightarrow{a_1|b_0+c_1} & (a_1 \triangleright a_0 \triangleright f) \bar{\boxplus} g_{a_0 a_1} \\ \xrightarrow{a_2|b_0+c_2} & (a_2 \triangleright a_1 \triangleright a_0 \triangleright f) \bar{\boxplus} g_{a_0 a_1 a_2} \\ \xrightarrow{a_3|b_0+c_3} & \dots \end{aligned}$$

So $\bar{\boxplus}$ does indeed behave as the pointwise extension of $\boxplus$.

In the rest of this paper, we show that the constructions used in the above examples apply more generally. In particular, the idea of defining pointwise extensions by adding input buffer operations works not only for the specific stream operation $\boxplus$, but, in fact, for all GSOS-defined operations on F-behaviours for any functor F. The meaning of the phrase 'GSOS-defined' is explained in detail in Section 5. Before that, however, we need to understand the notion of pointwise extension for operations on final F-coalgebras.

4. Pointwise behaviour and pointwise extensions of operations

In this section, we introduce the notion of pointwise behaviour of F-coalgebras with input. From now on, let A be a fixed set. An *F-coalgebra with input (in A)* is a coalgebra for the functor composition $F^A = (-)^A \circ F$. That is, $F^A(X) = (FX)^A$ and for a function $f : X \to Y$, the function

$$F^A(f) : (FX)^A \to (FY)^A$$

is defined by

$$F^A(f)(t)(a) = (Ff)(t(a)).$$

Our motivating examples in Section 3 focused on the case where $F = (B \times -)$, that is, F^A-coalgebras are Mealy machines, and F^A-behaviours are causal stream functions of type $f : A^\omega \to B^\omega$. Evaluating such a causal stream function at an input stream $\alpha \in A^\omega$ yields an element in B^ω, the set of F-behaviours. We will show that for any functor F, under the assumption that the final coalgebras for F and F^A exist, we can evaluate F^A-behaviours at input streams from A^ω to obtain F-behaviours. This observation relies on a construction that is similar to the wreath product of automata (Carton 2000; Straubing 1989).

Definition 4.1. Let $s : Y \to A \times Y$ be an $(A \times -)$-coalgebra. For an F^A-coalgebra $m : X \to (FX)^A$, we define the F-coalgebra

$$s \ltimes m : Y \times X \to F(Y \times X)$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

to be the function composition:

$$Y \times X \xrightarrow{s\times m} (A \times Y) \times (FX)^A \xrightarrow{\cong} Y \times (A \times (FX)^A)$$
$$\xrightarrow{\mathrm{id}_Y \times \epsilon_{FX}} Y \times FX \xrightarrow{\mathrm{st}^F_{Y,X}} F(Y \times X).$$

If $\langle Z, \zeta \rangle$ is a final F-coalgebra, then for all

$$s : Y \to A \times Y$$
$$m : X \to (FX)^A,$$

we get by finality, a unique F-coalgebra morphism $beh_{s\ltimes m} : Y \times X \to Z$:

$$\begin{array}{ccc} Y \times X & \overset{beh_{s\ltimes m}}{\cdots\cdots\cdots

Example 4.3. In our main example of streams and Mealy machines, the relevant functors are

$$F = (B \times -)$$
$$F^A = (B \times -)^A.$$

F-behaviours are streams over B and F^A-behaviours are causal stream functions of type $f : A^\omega \to B^\omega$ (*cf.* Example 2.2). The input evaluation and pointwise behaviour of causal stream functions are illustrated in the diagram

$$\begin{array}{ccccc} A^\omega \times \Gamma & & \langle \alpha, f \rangle & \xmapsto{\text{ev}} & f(\alpha) & & B^\omega \\ {\scriptstyle \langle hd,tl \rangle \ltimes \gamma}\downarrow & & \downarrow & & \downarrow & & \downarrow{\scriptstyle \langle hd,tl \rangle} \\ B \times (A^\omega \times \Gamma) & & \langle f[\alpha_0], \langle \alpha', f_{\alpha_0} \rangle \rangle & \xrightarrow{\text{id} \times \text{ev}} & \langle f[\alpha_0], f_{\alpha_0}(\alpha') \rangle & & B \times B^\omega \end{array}$$

(where we have simplified the notation by writing α_0 and α' for $hd(\alpha)$ and $tl(\alpha)$, respectively). Clearly, for all $f \in \Gamma$ and $\alpha \in A^\omega$, we have

$$\text{ev}(\langle \alpha, f \rangle) = f(\alpha).$$

We have already shown some examples of pointwise extensions of stream operations to causal stream functions in Section 3. Note, however, that not all stream operations can be extended pointwise to Γ. A simple example of this is given by the tail operation $tl : B^\omega \to B^\omega$. Its pointwise extension $\bar{tl} : \Gamma \to \Gamma$ would have to be defined for $f \in \Gamma$ and $\alpha \in A^\omega$ by

$$\bar{tl}(f)(\alpha) = tl(f(\alpha)).$$

But then $\bar{tl}(f)$ is not causal, in general. For example, taking $A = B$ and $f = \text{id}$, we have $\bar{tl}(\text{id}) = tl$, which is not causal, so $\bar{tl}$ is not well defined as an operation on Γ.

On the other hand, causal stream operations *can* be pointwise extended to Γ. More precisely, we say that a k-ary operation $\sigma : (B^\omega)^k \to B^\omega$ is causal if for all $\beta_0, \ldots, \beta_{k-1} \in B^\omega$ and all $n \in \mathbb{N}$, we have that $\sigma(\beta_0, \ldots, \beta_{k-1})(n)$ depends only on $\beta_i \restriction_n$ for $i \in \{0, \ldots, k-1\}$. It is straightforward to show that if $\sigma : (B^\omega)^k \to B^\omega$ is causal, then its pointwise extension $\bar{\sigma}$ preserves causality, that is, for $f_0, \ldots, f_{k-1} \in \Gamma$, the function

$$\bar{\sigma}(f_0, \ldots, f_{k-1}) : A^\omega \to B^\omega$$

is causal, and hence $\bar{\sigma}$ is a well defined operation on Γ.

Example 4.4. Consider the functor $FX = 1 + X$ and recall from Examples 2.3 and 2.4 that the set of F-behaviours is the extended natural numbers, and the final F^A-coalgebra $\langle \mathscr{L}, D \rangle$ is carried by the set of non-empty, prefix-closed languages. A transition in the F-coalgebra $\langle hd, tl \rangle \ltimes D$ is defined by

$$\begin{array}{ccc} A^\omega \times \mathscr{L} & \xrightarrow{\langle hd,tl \rangle \ltimes D} & 1 + (A^\omega \times \mathscr{L}) \\ \langle \alpha_0 : \alpha', L \rangle & \longmapsto & \begin{cases} \langle \alpha', L_{\alpha_0} \rangle & \text{if } \alpha_0 \in L \\ \bot & \text{if } \alpha_0 \notin L \end{cases} \end{array}$$

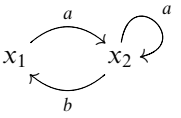
https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

Given a state x in an F^A-coalgebra $t: X \to (1+X)^A$ and an $\alpha \in A^\omega$, the pointwise F-behaviour of x on α is the number of steps that can be made in t starting from x on input α. More precisely,

$$\mathrm{ev}(\langle \alpha, x\rangle) = \sup\{n \leqslant \omega \mid \alpha\restriction_n \in L(x)\} \quad \in \quad \mathbb{N} + \{\omega\}.$$

In particular, if α labels an infinite path starting in x, then $\mathrm{ev}(\langle \alpha, x\rangle) = \omega$.

As an example, consider the following (fragment of the final) $(1+-)^A$-coalgebra with input alphabet $A = \{a, b\}$:



We list a few examples of pointwise behaviours:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}l}
\mathrm{ev}(\langle a^\omega, x_1\rangle) &=& \omega & \mathrm{ev}(\langle a^\omega, x_2\rangle) &=& \omega & \\
\mathrm{ev}(\langle b^\omega, x_1\rangle) &=& 0 & \mathrm{ev}(\langle b^\omega, x_2\rangle) &=& 1 & \\
\mathrm{ev}(\langle a^n b^\omega, x_1\rangle) &=& n+1 & \mathrm{ev}(\langle a^n b^\omega, x_2\rangle) &=& n+1 & \forall n > 0.
\end{array}$$

Natural operations on F-behaviours are max ($\vee$), plus ($+$) and times ($\cdot$) in the extended arithmetic where $\omega + x = \omega$ for all $x \in \{\omega\} + \mathbb{N}$ and $\omega \cdot x = \omega$ for all $x \in \{\omega\} + \mathbb{N}_{>0}$ and $\omega \cdot 0 = 0$. Using the above example, the pointwise extensions of these operations should satisfy:

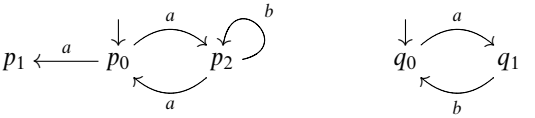
$$\begin{array}{rcl@{\qquad}rcl}
\mathrm{ev}(\langle a^\omega, x_1 \bar{\vee} x_2\rangle) &=& \omega & \mathrm{ev}(\langle b^\omega, x_1 \bar{\vee} x_2\rangle) &=& 1 \\
\mathrm{ev}(\langle a^\omega, x_1 \bar{\cdot} x_2\rangle) &=& \omega & \mathrm{ev}(\langle b^\omega, x_1 \bar{+} x_2\rangle) &=& 1 \\
\mathrm{ev}(\langle a^\omega, x_1 \bar{+} x_2\rangle) &=& \omega & \mathrm{ev}(\langle b^\omega, x_1 \bar{\cdot} x_2\rangle) &=& 0
\end{array}$$

$$\begin{array}{rcll}
\mathrm{ev}(\langle a^n b^\omega, x_1 \bar{\vee} x_2\rangle) &=& n+1 & \\
\mathrm{ev}(\langle a^n b^\omega, x_1 \bar{+} x_2\rangle) &=& 2n+2 & \\
\mathrm{ev}(\langle a^n b^\omega, x_1 \bar{\cdot} x_2\rangle) &=& n^2+2n+1 & \forall n > 0.
\end{array}$$

Example 4.5. Recall from Examples 2.5 and 2.6 that the final $\mathcal{P}_\omega$-coalgebra $\langle T, \tau\rangle$ consists of all strongly extensional, finitely branching trees with the subtree relation, and the final $\mathcal{P}_\omega^A$-coalgebra $\langle P, \phi\rangle$ is carried by the set of image-finite processes. For $\alpha \in A^\omega$ and $p \in P$, we have

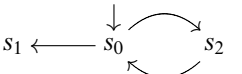
$$\begin{array}{ccc}
A^\omega \times P & \xrightarrow{\langle hd, tl\rangle \ltimes \phi} & \mathcal{P}_\omega(A^\omega \times P) \\
\langle \alpha, p\rangle & \longmapsto & \{\langle \alpha', q\rangle \mid p \xrightarrow{\alpha_0} q\}.
\end{array}$$

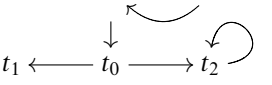
Evaluating $p \in P$ at an input stream $\alpha \in A^\omega$ can be viewed as removing from p the paths that are not labelled by a prefix of α, and dropping labels. The pointwise behaviour is obtained by quotienting the resulting tree with $\mathcal{P}_\omega$-bisimilarity. For example, let $A = \{a, b\}$, and p_0 and q_0 be the processes

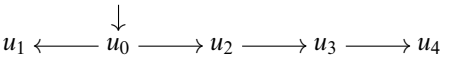


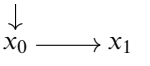
https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

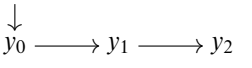
Then we find that the pointwise behaviours of p_0 and q_0 on a^ω, ab^ω and $(ab)^\omega$, respectively, are given by

$\mathrm{ev}(\langle a^\omega, p_0\rangle)$ 

$\mathrm{ev}(\langle ab^\omega, p_0\rangle)$ 

$\mathrm{ev}(\langle (ab)^\omega, p_0\rangle)$ 

$\mathrm{ev}(\langle a^\omega, q_0\rangle)$ 

$\mathrm{ev}(\langle ab^\omega, q_0\rangle)$ 

$\mathrm{ev}(\langle (ab)^\omega, q_0\rangle)$ 

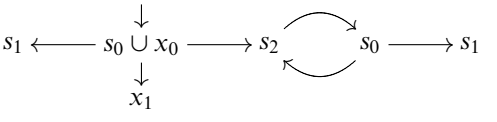
We now consider the following operations on F-behaviours s and t. The *join* $s \cup t$ is a root node that has all the subtrees of s and t as subtrees. The *composition* $s;t$ plugs in t at all leaf nodes of s. The *interleaving* $s \otimes t$ is a tree version of shuffle product. These operations are defined by the following rules:

join :
$$\frac{x_0 \longrightarrow x_1}{x_0 \cup y \longrightarrow x_1} \qquad \frac{y_0 \longrightarrow y_1}{x \cup y_0 \longrightarrow y_1}$$

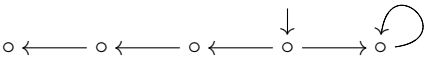
composition :
$$\frac{x_0 \longrightarrow x_1}{x_0; y \longrightarrow x_1; y} \qquad \frac{x \not\longrightarrow \qquad y_0 \longrightarrow y_1}{x; y_0 \longrightarrow y_1}$$

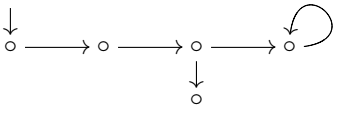
interleaving :
$$\frac{x_0 \longrightarrow x_1 \qquad y_0 \longrightarrow y_1}{x_0 \otimes y_0 \longrightarrow (x_1 \otimes y_0) \cup (x_0 \otimes y_1)}$$

We construct the join $s_0 \cup x_0$ by first computing transitions using the rules for $\cup$:



Quotienting with bisimilarity, we find that $s_0 \cup x_0 = s_0$. Compositions are also easily computed:

$t_0; y_0$: 

$y_0; t_0$: 

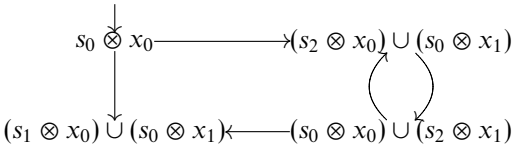
$x_0; z_0 = z_0; x_0$: 

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

It follows that the pointwise extension $\bar{;}$ should satisfy

$$\begin{aligned}\mathrm{ev}(\langle ab^\omega, p_0 \mathbin{\bar{;}} q_0\rangle) &= t_0 ; y_0\\ \mathrm{ev}(\langle ab^\omega, q_0 \mathbin{\bar{;}} p_0\rangle) &= y_0 ; t_0\\ \mathrm{ev}(\langle (ab)^\omega, p_0 \mathbin{\bar{;}} q_0\rangle) &= z_0\\ \mathrm{ev}(\langle (ab)^\omega, q_0 \mathbin{\bar{;}} p_0\rangle) &= z_0.\end{aligned}$$

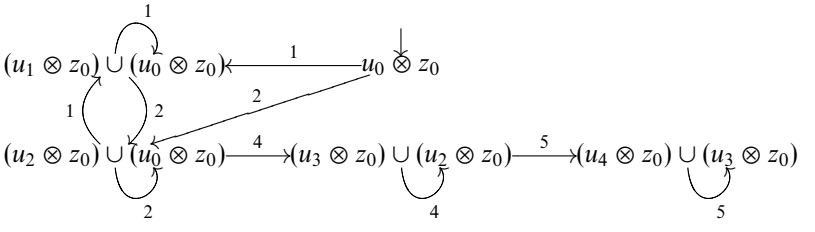
Finally, we compute a few interleaving products. First, $s_0 \otimes x_0$:



Quotienting with bisimilarity, we find that $s_0 \otimes x_0 = s_0$. As a final example, we compute $u_0 \otimes z_0$. To aid the calculation, we first compute some intermediate transitions:

1. $u_0 \otimes z_0 \longrightarrow (u_1 \otimes z_0) \cup (u_0 \otimes z_0)$
2. $u_0 \otimes z_0 \longrightarrow (u_2 \otimes z_0) \cup (u_0 \otimes z_0)$
3. $u_1 \otimes z_0 \not\longrightarrow$ (since $u_1 \not\longrightarrow$)
4. $u_2 \otimes z_0 \longrightarrow (u_3 \otimes z_0) \cup (u_2 \otimes z_0)$
5. $u_3 \otimes z_0 \longrightarrow (u_4 \otimes z_0) \cup (u_3 \otimes z_0)$
6. $u_4 \otimes z_0 \not\longrightarrow$ (since $u_4 \not\longrightarrow$).

The transition graph of $u_0 \otimes z_0$ is shown below – we have labelled the transition arrows with a reference to the intermediate transition from which it arises:



No states can terminate, so $u_0 \otimes z_0 = z_0$. We conclude that the pointwise extension of $\otimes$ should satisfy

$$\begin{aligned}\mathrm{ev}(\langle a^\omega, p_0 \mathbin{\bar{\otimes}} q_0\rangle) &= s_0\\ \mathrm{ev}(\langle (ab)^\omega, p_0 \mathbin{\bar{\otimes}} q_0\rangle) &= z_0.\end{aligned}$$

In the above examples, the pointwise extensions of operations were calculated on some specific argument values. To check whether these pointwise extensions are well defined in general, and whether they can be defined in a rule-based fashion as in Section 3, we need to understand these rule-based specifications in a general coalgebraic setting – this is the aim of the next section.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

5. Distributive laws and GSOS rules

The equation- and rule-based definitions in the examples of Sections 3 and 4 are special cases of a general framework for defining operations on final coalgebras, parameterised both by the behaviour functor F and the signature Σ of operations. The framework, developed in Turi and Plotkin (1997), is based on the abstract notion of a distributive law.

5.1. *Basic distributive laws*

Let F be a behaviour functor, $\langle Z, \zeta \rangle$ be the final F-coalgebra and Σ be an algebraic signature. A natural transformation $\lambda : \Sigma F \Longrightarrow F\Sigma$, that is, a distributive law of Σ over F, induces a Σ-algebra on Z by finality as the unique F-coalgebra map from $\lambda_Z \circ \Sigma\zeta$ to $\langle Z, \zeta \rangle$. This is illustrated by

$$\begin{array}{ccccc} \Sigma Z & \xrightarrow{\Sigma\zeta} & \Sigma F Z & \xrightarrow{\lambda_Z} & F\Sigma Z \\ {\scriptstyle\sigma}\downarrow & & & & \downarrow{\scriptstyle F\sigma} \\ Z & & \xrightarrow[\cong]{\zeta} & & FZ \end{array} \tag{12}$$

For instance, the definition of $\oplus$ in Example 3.1 induces a natural transformation

$$\lambda : (\mathbb{R} \times -) \times (\mathbb{R} \times -) \implies \mathbb{R} \times (- \times -)$$

(that is, $\Sigma X = X \times X$ and $FX = \mathbb{R} \times X$) whose X-component is given by

$$\begin{array}{rccc} \lambda_X : & (\mathbb{R} \times X) \times (\mathbb{R} \times X) & \Longrightarrow & \mathbb{R} \times (X \times X) \\ & \langle\langle a, x_1\rangle, \langle b, y_1\rangle\rangle & \mapsto & \langle a + b, \langle x_1, y_1\rangle\rangle. \end{array} \tag{13}$$

It is straightforward to check that the operation $\sigma : \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ arising from this λ as in (12), is the expected operation $\oplus$ induced by the equations of Example 3.1 according to Rutten (2003).

5.2. *Monadic distributive laws*

In specifications associated with basic distributive laws, all expressions on the right-hand sides of rule conclusions must be Σ-terms of depth exactly 1. However, the definitions of some useful operations do not conform to this restriction.

For an easy example, consider for $FX = \mathbb{R} \times X$ a unary 'head replacement' operation $a/- : \mathbb{R}^\omega \to \mathbb{R}^\omega$, defined equivalently by equations or rules by

$$\begin{aligned} hd(a/x_0) &= a \\ tl(a/x_0) &= tl(x_0) \end{aligned}$$

that is,

$$\frac{x_0 \xrightarrow{b} x_1}{a/x_0 \xrightarrow{a} x_1}.$$

Note how the conclusion of the above rule is a variable, that is, a term of depth 0 rather than 1. For this reason, the above definition does not correspond to a basic distributive

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

law $\lambda : \Sigma F \Longrightarrow F\Sigma$. There are also useful examples where the relevant terms have depth more than 1 (see, for example, the interleaving operation in Example 4.5).

To deal with such examples, one can consider natural transformations of the type $\rho : \Sigma F \Longrightarrow FT$, which allow arbitrary Σ-terms where only terms of depth 1 were previously allowed. Such natural transformations induce Σ-operations on final F-coalgebras in much the same way as basic distributive laws, as the unique maps $\sigma : \Sigma Z \to Z$ that make the diagram

$$\begin{array}{ccccc} \Sigma Z & \xrightarrow{\Sigma\zeta} & \Sigma F Z & \xrightarrow{\rho_Z} & FTZ \\ {\scriptstyle\sigma}\,\vdots & & & & \downarrow{\scriptstyle F\sigma^\sharp} \\ Z & & \xrightarrow[\cong]{\zeta} & & FZ, \end{array} \tag{14}$$

commute, where $\sigma^\sharp$ is the T-algebra corresponding to σ as described in Section 2.2.

An alternative way to understand this definition, and convince oneself that σ as above does indeed exist and is unique, is to observe that transformations $\rho : \Sigma F \Longrightarrow FT$ are in 1–1 correspondence with distributive laws of the monad T over the endofunctor F, that is, natural transformations $\lambda : TF \Longrightarrow FT$ that respect the monad structure of T. Such distributive laws induce the algebraic structure $\sigma^\sharp$ uniquely as in (12), but with T substituted for Σ.

5.3. *GSOS specifications*

For an example that does not conform to either of the law types explained above, consider the definition of $\boxplus$ in Example 3.2. It does not correspond to a natural transformation $\lambda : \Sigma F \Longrightarrow F\Sigma$ (for $\Sigma X = X \times X$ and $FX = \mathbb{R} \times X$), and intuitively, the reason for this is the use of the variable x_0 on the right-hand side of an equation, or in the target of the conclusion of a rule. However, the definition corresponds to a law of the type

$$\rho : \Sigma(Id \times F) \Longrightarrow F\Sigma$$

where ρ has X-component

$$\begin{array}{rccc} \rho_X : & (X \times \mathbb{R} \times X) \times (X \times \mathbb{R} \times X) & \Longrightarrow & \mathbb{R} \times (X \times X) \\ & \langle\langle x_0, a, x_1\rangle, \langle y_0, b, y_1\rangle\rangle & \mapsto & \langle a + b, \langle x_0, y_1\rangle\rangle. \end{array} \tag{15}$$

One can also combine the expressivity of this type of laws with that of monadic distributive laws described above, and consider the following definition.

Definition 5.1. A *GSOS specification* (of Σ over F) is a natural transformation

$$\rho : \Sigma(Id \times F) \Longrightarrow FT.$$

GSOS specifications are a particularly useful type of distributive laws, which can be used to describe many interesting definitions. Their name comes from the fact that for the endofunctor $FX = (\mathcal{P}_\omega X)^A$ (see Example 2.6), they correspond (Turi and Plotkin 1997; Bartels 2004) to structural operational semantic specifications of LTSs in the well known GSOS format (Aceto *et al.* 2001).

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

To induce a Σ-operation σ on the final F-coalgebra from a GSOS specification ρ, we proceed by analogy with the case of monadic distributive laws and define it as the unique map that makes the diagram

$$\begin{array}{ccccc} \Sigma Z & \xrightarrow{\Sigma\langle \mathrm{id},\zeta\rangle} & \Sigma(Z \times FZ) & \xrightarrow{\rho_Z} & FTZ \\ {\scriptstyle\sigma}\vdots & & & & \downarrow{\scriptstyle F\sigma^\sharp} \\ Z & & \xrightarrow[\cong]{\zeta} & & FZ \end{array} \tag{16}$$

commute. Again, the unique existence of σ follows from the correspondence of GSOS specifications $\rho : \Sigma(Id \times F) \Longrightarrow FT$ with distributive laws of the monad T over the copointed endofunctor $(Id \times F)$, that is, natural transformations

$$\lambda : T(Id \times F) \Longrightarrow (Id \times F)T$$

subject to a few axioms – see Lenisa *et al.* (2004) for details. Again, such distributive laws induce the algebraic structure $\sigma^\sharp$ uniquely as in (12), but with T substituted for Σ and $(Id \times F)$ for F.

6. Pointwise extensions of distributive laws

We shall now focus on the problem of extending distributive laws for arbitrary behaviour functors F to similar laws for F^A, so that the resulting operations on final F^A-coalgebras are pointwise extensions of the operations on final F-coalgebras defined by the original distributive laws. It turns out that the solutions applied to the two particular examples in Section 3 also work in this general setting.

6.1. *Basic distributive laws*

As before, let F be a behaviour functor, $\langle Z, \zeta\rangle$ be a final F-coalgebra and Σ be an algebraic signature. Suppose an operation $\sigma : \Sigma Z \to Z$ arises from a basic distributive law $\lambda : \Sigma F \Longrightarrow F\Sigma$ by coinduction as in (12). Assume, moreover, that $\langle \overline{Z}, \overline{\zeta}\rangle$ is a final F^A-coalgebra.

We define a basic distributive law

$$\overline{\lambda} : \Sigma(F-)^A \Longrightarrow (F\Sigma-)^A$$

from λ, with $\overline{\lambda}_X$ given by

$$\Sigma(FX)^A \xrightarrow{\mathrm{cs}^{\Sigma}_{A,FX}} (\Sigma FX)^A \xrightarrow{\lambda^A_X} (F\Sigma X)^A, \tag{17}$$

and define $\overline{\sigma} : \Sigma\overline{Z} \to \overline{Z}$ from $\overline{\lambda}$ as in (12).

Theorem 6.1. Let F be a functor with final coalgebra $\langle Z, \zeta\rangle$ and let $\langle \overline{Z}, \overline{\zeta}\rangle$ be a final F^A-coalgebra. If

$$\lambda : \Sigma F \Longrightarrow F\Sigma$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

is a distributive law (of Σ over F) that induces a Σ-algebra $\sigma : \Sigma Z \to Z$, then λ can be lifted to a distributive law

$$\overline{\lambda} : \Sigma(F-)^A \Longrightarrow (F\Sigma-)^A$$

(of Σ over F^A) that induces a Σ-algebra $\overline{\sigma} : \Sigma\overline{Z} \to \overline{Z}$ that is the pointwise extension of σ.

Proof. Define $\overline{\lambda}$ by (17) – see the Appendix for the full proof. □

Example 6.2. Let us calculate $\overline{\lambda}$ for the law λ in (13) that defines the operation $\oplus : \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$ from Example 3.1. The syntax and behaviour functors are $\Sigma X = X \times X$ and $F^A X = (\mathbb{R} \times X)^{\mathbb{R}}$, so an X-component of $\overline{\lambda}$ is of the type

$$\overline{\lambda}_X : (\mathbb{R} \times X)^{\mathbb{R}} \times (\mathbb{R} \times X)^{\mathbb{R}} \Longrightarrow (\mathbb{R} \times X \times X)^{\mathbb{R}}.$$

Below, for $\phi \in (\mathbb{R} \times X)^{\mathbb{R}}$ and $a \in \mathbb{R}$, we denote the projections of $\phi(a)$ by $\phi_0(a)$ and $\phi_1(a)$, that is, $\phi(a) = \langle \phi_0(a), \phi_1(a) \rangle$. Instantiating (17) and (13), $\overline{\lambda}_X$ is defined by

$$\begin{array}{ccc}
\langle \phi, \psi \rangle & \in & (\mathbb{R} \times X)^{\mathbb{R}} \times (\mathbb{R} \times X)^{\mathbb{R}} \\
\downarrow & & \downarrow {\scriptstyle \mathrm{cs}^{\Sigma}_{\mathbb{R}, \mathbb{R} \times X}} \\
\lambda a. \langle \phi_0(a), \phi_1(a), \psi_0(a), \psi_1(a) \rangle & \in & (\mathbb{R} \times X \times \mathbb{R} \times X)^{\mathbb{R}} \\
\downarrow & & \downarrow {\scriptstyle \lambda_X^{\mathbb{R}}} \\
\lambda a. \langle \phi_0(a) + \psi_0(a), \langle \phi_1(a), \psi_1(a) \rangle \rangle & \in & (\mathbb{R} \times X \times X)^{\mathbb{R}}
\end{array}$$

Formulating the above $\overline{\lambda}$ as an inference rule, we recognise the rule for $\bar{\oplus}$ in (5), with

$$\begin{aligned} b &= \phi_0(a) \\ c &= \psi_0(a) \\ x_1 &= \phi_1(a) \\ y_1 &= \psi_1(a). \end{aligned}$$

It should be noted that the same construction also works for the more expressive type of specifications $\rho : \Sigma F \Longrightarrow FT$. To define the pointwise extension of an operation defined by such a specification, we first obtain from ρ a monadic distributive law $\lambda : TF \Longrightarrow FT$, and then apply the construction (17) and Theorem 6.1, with Σ replaced by T throughout.

6.2. *GSOS specifications*

We shall now move on to defining pointwise extensions of operations defined by GSOS specifications $\rho : \Sigma(Id \times F) \Longrightarrow FT$.

First, note that unlike in the case of monadic distributive laws above, we cannot use the correspondence of GSOS specifications and distributive laws of T over $Id \times F$, and then apply (17) and Theorem 6.1 with Σ replaced by T and F replaced by $Id \times F$ throughout.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

The reason for this is that in the natural transformation $\overline{\lambda}$ obtained from (17) in this case, an X-component has domain $T(X \times FX)^A$, and not $T(X \times (FX)^A)$ as required to define the pointwise extension operation by (16).

It seems that the problem can be circumvented by precomposing the counterpart of (17) with an appropriate (co)strength. In this way, from a GSOS specification ρ, we obtain a GSOS specification

$$\overline{\rho} : \Sigma(Id \times (F-)^A) \Longrightarrow (FT)^A$$

with $\overline{\rho}_X$ defined by

$$\Sigma(X \times (FX)^A) \xrightarrow{\mathrm{cs}^{\Sigma(X\times -)}_{A,FX}} (\Sigma(X \times FX))^A \xrightarrow{\rho^A_X} (FTX)^A, \tag{18}$$

and then proceed to define an extended operation $\overline{\sigma} : \Sigma\overline{Z} \to \overline{Z}$ from (16).

However, it turns out that $\overline{\sigma}$ obtained in this way is *not* the pointwise extension of the operation $\sigma : \Sigma Z \to Z$, that is, the relevant diagram (11) does not commute. Indeed, one may repeat the development of Example 6.2 for (18) with ρ defined by (15), and find that the resulting $\overline{\rho}$ corresponds to the rule (6), which does not define the pointwise extension of $\boxplus$, as we saw in Example 3.2.

To treat the case of arbitrary GSOS specifications correctly, we need to proceed in a more subtle way, by analogy with the development of Example 3.2. First, we extend the syntax Σ by defining

$$\begin{aligned} \Sigma_{\triangleright} &= A \times - \\ \overline{\Sigma} &= \Sigma + \Sigma_{\triangleright}. \end{aligned}$$

This amounts to adding $|A|$ auxiliary unary operations to the syntax. In examples, we will denote these operations by $a \triangleright -$, for $a \in A$. Intuitively, their semantics will be 'one-element buffer' operations. Let $\overline{T}$ be the free monad over $\overline{\Sigma}$.

The pointwise extension of σ will be defined with the help of an algebra $\overline{\sigma} : \overline{\Sigma}\,\overline{Z} \to \overline{Z}$ such that the diagram

$$\begin{array}{ccccc} A^\omega \times \Sigma\overline{Z} & \xrightarrow{\mathrm{st}^{\Sigma}_{A^\omega,\overline{Z}}} & \Sigma(A^\omega \times \overline{Z}) & \xrightarrow{\Sigma \mathrm{ev}} & \Sigma Z \\ \downarrow{\scriptstyle \mathrm{id}\times \iota_{\overline{Z}}} & & & & \\ A^\omega \times \overline{\Sigma}\,\overline{Z} & & & & \downarrow{\scriptstyle \sigma} \\ \downarrow{\scriptstyle \mathrm{id}\times \overline{\sigma}} & & & & \\ A^\omega \times \overline{Z} & & \xrightarrow[\mathrm{ev}]{} & & Z \end{array} \tag{19}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

commutes (compare with (11)), where $\iota : \Sigma \to \overline{\Sigma}$ is the coproduct injection. To define $\overline{\sigma}$, we provide a GSOS specification

$$\overline{\rho} : \overline{\Sigma}(Id \times (F-)^A) \Longrightarrow (F\overline{T}-)^A,$$

by cases of $\overline{\Sigma}$:

— for $\Sigma_{\triangleright}$:

$$\begin{array}{c} A \times (X \times (FX)^A) \xrightarrow{\pi_{13}} A \times (FX)^A \xrightarrow{\epsilon_{FX}} FX \xrightarrow{\eta_{FX}} \\ (A \times FX)^A \xrightarrow{(\mathrm{st}^F_{A,X})^A} (F(A \times X))^A \hookrightarrow (F\overline{T}X)^A \end{array} \tag{20}$$

— for Σ:

$$\begin{array}{c} \Sigma(X \times (FX)^A) \xrightarrow{\Sigma(\eta_X \times \mathrm{id})} \Sigma((A \times X)^A \times (FX)^A) \xrightarrow{\mathrm{cs}^{\Sigma(-\times-)}_{A,A\times X,FX}} \\ (\Sigma(A \times X \times FX))^A \hookrightarrow (\Sigma((A \times X + X) \times F(A \times X + X)))^A \xrightarrow{\rho^A_{A\times X+X}} \\ (FT(A \times X + X))^A \hookrightarrow (F\overline{T}X)^A \end{array} \tag{21}$$

This $\overline{\rho}$ defines an algebra $\overline{\sigma} : \overline{\Sigma}\,\overline{Z} \to \overline{Z}$ as usual, by (16). With some straightforward, albeit tedious, diagram chasing, we can show that (19) commutes.

Theorem 6.3. Let F be a functor with final coalgebra $\langle Z, \zeta \rangle$, and let $\langle \overline{Z}, \overline{\zeta} \rangle$ be a final F^A-coalgebra. If

$$\rho : \Sigma(Id \times F-) \Longrightarrow (FT-)$$

is a GSOS specification (of Σ over F) that induces a Σ-algebra $\sigma : \Sigma Z \to Z$, then ρ can be lifted to a GSOS specification

$$\overline{\rho} : \overline{\Sigma}(Id \times (F-)^A) \Longrightarrow (F\overline{T}-)^A$$

(of $\overline{\Sigma}$ over F^A) that induces a $\overline{\Sigma}$-algebra $\overline{\sigma} : \overline{\Sigma}\,\overline{Z} \to \overline{Z}$ such that $\overline{\sigma} \circ \iota_{\overline{Z}} : \Sigma\overline{Z} \to \overline{Z}$ is the pointwise extension of σ.

Proof. Define $\overline{\rho}$ by (20) and (21) – see the Appendix for the full proof. □

We can now illustrate the construction behind Theorem 6.3 using the operation $\boxplus$ from Example 3.2.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

Example 6.4. Let us calculate $\overline{\rho}\colon \overline{\Sigma}(Id \times (F-)^A) \Longrightarrow (F\overline{T}-)^A$ for the law ρ in (15) that defines the operation $\boxplus\colon \mathbb{R}^\omega \times \mathbb{R}^\omega \to \mathbb{R}^\omega$. The syntax and behaviour functors are $\Sigma X = X \times X$ and $F^A X = (\mathbb{R} \times X)^{\mathbb{R}}$, hence

$$\begin{aligned} \Sigma_{\triangleright} X &= \mathbb{R} \times X \\ \overline{\Sigma} X &= (X \times X) + (\mathbb{R} \times X) \end{aligned}$$

and $\overline{\rho}$ is of type

$$\overline{\rho}\colon \overline{\Sigma}(Id \times (\mathbb{R} \times -)^{\mathbb{R}}) \Longrightarrow (\mathbb{R} \times \overline{T}-)^{\mathbb{R}}.$$

The two cases for $\overline{\rho}$ have X-components

$$\begin{aligned} \overline{\rho}^{\triangleright}_X &: \ \mathbb{R} \times X \times (\mathbb{R} \times X)^{\mathbb{R}} \ \to (\mathbb{R} \times \overline{T}X)^{\mathbb{R}} \\ \overline{\rho}^{\Sigma}_X &: \ \Sigma(X \times (\mathbb{R} \times X)^{\mathbb{R}}) \ \to (\mathbb{R} \times \overline{T}X)^{\mathbb{R}} \end{aligned}$$

given by (20) and (21), respectively. Intuitively, $\overline{\rho}^{\triangleright}$ specifies the F^A-behaviour of the buffer operations and $\overline{\rho}^{\Sigma}$ specifies the F^A-behaviour of the Σ-operation.

Again, for $\phi \in (\mathbb{R} \times X)^{\mathbb{R}}$ we let ϕ_0 and ϕ_1 be defined by $\phi(a) = \langle \phi_0(a), \phi_1(a) \rangle$ for all $a \in \mathbb{R}$. To simplify the notation, we shall suppress some coproduct injections and treat them as set inclusions.

For $\overline{\rho}^{\triangleright}$, (20) instantiates to

$$\begin{array}{ccc} \langle a, x, \phi \rangle & \in & \mathbb{R} \times (X \times (\mathbb{R} \times X)^{\mathbb{R}}) \\ \downarrow & & \downarrow{\scriptstyle \pi_{13}} \\ \langle a, \phi \rangle & \in & \mathbb{R} \times (\mathbb{R} \times X)^{\mathbb{R}} \\ \downarrow & & \downarrow{\scriptstyle \epsilon_{\mathbb{R} \times X}} \\ \langle \phi_0(a), \phi_1(a) \rangle & \in & \mathbb{R} \times X \\ \downarrow & & \downarrow{\scriptstyle \eta_{\mathbb{R} \times X}} \\ \lambda c. \langle c, \langle \phi_0(a), \phi_1(a) \rangle \rangle & \in & (\mathbb{R} \times (\mathbb{R} \times X))^{\mathbb{R}} \\ \downarrow & & \downarrow{\scriptstyle (\mathrm{st}^{\mathbb{R} \times -}_{\mathbb{R}, X})^{\mathbb{R}}} \\ \lambda c. \langle \phi_0(a), \langle c, \phi_1(a) \rangle \rangle & \in & (\mathbb{R} \times (\mathbb{R} \times X))^{\mathbb{R}} \\ \| & & \downarrow \\ \lambda c. \langle \phi_0(a), \langle c, \phi_1(a) \rangle \rangle & \in & (\mathbb{R} \times \overline{T}X)^{\mathbb{R}}. \end{array}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

The last step involves the inclusion of $\mathbb{R} \times X = \Sigma_{\triangleright} X$ into $\overline{T}X$. So $\langle c, \phi_1(a)\rangle$ should be read as $c \triangleright \phi_1(a)$. Formulating $\overline{\rho}^{\triangleright}$ as a GSOS rule, we can recognise the rule for $a \triangleright x_0$ in (9) (with $b = \phi_0(a)$, $x_0 = x$ and $x_1 = \phi_1(a)$).

For $\overline{\rho}^{\Sigma}$, (21) instantiates to:

$$
\begin{array}{ccc}
\langle\langle x, \phi\rangle, \langle y, \psi\rangle\rangle & \in & (X \times (\mathbb{R} \times X)^{\mathbb{R}})^2 \\
\Big\downarrow & & \Big\downarrow{\scriptstyle \Sigma(\eta_X \times \mathrm{id})} \\
\langle\langle \lambda a.\langle a, x\rangle, \phi\rangle, \langle \lambda a.\langle a, y\rangle, \psi\rangle\rangle & \in & ((\mathbb{R} \times X)^{\mathbb{R}} \times (\mathbb{R} \times X)^{\mathbb{R}})^2 \\
\Big\downarrow & & \Big\downarrow{\scriptstyle \mathrm{cs}^{\Sigma(-\times-)}_{\mathbb{R},\mathbb{R}\times X,\mathbb{R}\times X}} \\
\lambda a.\langle\langle\langle a, x\rangle, \langle \phi_0(a), \phi_1(a)\rangle\rangle, & \in & (((\mathbb{R} \times X) \times (\mathbb{R} \times X))^2)^{\mathbb{R}} \\
\quad \langle\langle a, y\rangle, \langle \psi_0(a), \psi_1(a)\rangle\rangle\rangle & & \Big\downarrow \\
\| & & \| \\
\lambda a.\langle\langle\langle a, x\rangle, \phi_0(a), \phi_1(a)\rangle, & \in & (((\mathbb{R} \times X + X) \times \mathbb{R} \times (\mathbb{R} \times X + X))^2)^{\mathbb{R}} \\
\quad \langle\langle a, y\rangle, \psi_0(a), \psi_1(a)\rangle\rangle & & \Big\downarrow{\scriptstyle \rho^{\mathbb{R}}_{\mathbb{R}\times X+X}} \\
\Big\downarrow & & \\
\lambda a.\langle \phi_0(a) + \psi_0(a), \langle\langle a, x\rangle, \psi_1(a)\rangle\rangle & \in & (\mathbb{R} \times T(\mathbb{R} \times X + X))^{\mathbb{R}} \\
\| & & \Big\downarrow \\
\lambda a.\langle \phi_0(a) + \psi_0(a), \langle\langle a, x\rangle, \psi_1(a)\rangle\rangle & \in & (\mathbb{R} \times \overline{T}(X))^{\mathbb{R}}.
\end{array}
$$

The last step involves the natural inclusion of terms induced by mapping a pair $\langle a, x\rangle \in \mathbb{R} \times X$ to the buffer expression $a \triangleright x \in \Sigma_{\triangleright} X \subseteq \overline{T}X$. With this inclusion in mind, we see that the rule for $\bar{\boxplus}$ in (9) is the rule-based formulation of $\overline{\rho}^{\Sigma}$.

Informally, $\overline{\rho}^{\Sigma}$ can be described as being obtained from ρ by replacing all occurrences of $x_0 \in X$ on the right-hand sides with $a \triangleright x_0$ and adding input labels to arrows. In the next section we will see more examples of how $\overline{\rho}$ is obtained from ρ.

7. Further examples

We shall now sketch a few more example applications of the constructions and results of Section 6.

Example 7.1. Taking $F = 2 \times -$ and $A = 2$, the $(2 \times -)$-behaviours are bitstreams and $(2 \times -)^2$-coalgebras are binary Mealy machines whose behaviours are causal bitstream functions (*cf.* Examples 2.1 and 2.2). Hansen and Rutten (2010) gives a coalgebraic treatment of the so-called 2-adic bitstream operations and their pointwise extensions to causal bitstream functions. The main goal of Hansen and Rutten (2010) was to construct binary Mealy machine realisations from 2-adic function expressions. A crucial step in this result consists of showing that 2-adic function expressions can be given a Mealy machine structure. However, it was not clear whether this Mealy machine of terms comes about

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

from the existence of a distributive law. A reason for this is that the buffer operations were not explicitly used in Hansen and Rutten (2010), so the general picture did not emerge. Since the 2-adic bitstream operations are defined in the GSOS format, Theorem 6.3 tells us that a distributive law exists and, moreover, how to derive it. We now present this distributive law, except that we leave out the (guarded) inverse operation to simplify the presentation – it is straightforward to apply Theorem 6.3 to include the inverse as well.

The 2-adic bitstream operations arise from viewing $(a_0, a_1, a_2, \ldots) \in 2^\omega$ as the coefficients of the power series representation of the *2-adic integer* $\sum_{i=0}^{\omega} a_i 2^i$ (Koblitz 1984). The 2-adic integers include the set Q_{odd} of rational numbers with odd denominator via the map $B : Q_{odd} \to 2^\omega$ that converts numbers to their base 2 representation. For a positive integer n, $B(n)$ is simply obtained by writing n in binary and padding with a tail of zeros, for example, $B(6) = (0, 1, 1, 0, 0, 0, \ldots)$. Negative integers are represented by taking an infinitary version of two's complement, for example, $B(-6) = (0, 1, 0, 1, 1, 1, \ldots)$. More generally, for any $q \in Q_{odd}$, $B(q)$ is an eventually periodic bitstream.

The 2-adic signature $\Sigma_{2\text{-adic}}$ (without inverse) consists of constants [0], [1] and X denoting $B(0), B(1)$ and $B(2)$, respectively, together with $+$ (addition), $-$ (unary minus) and $\times$ (multiplication). An attractive property of 2-adic representations is that rational arithmetic can be carried out in a sequential manner very similar to how one computes with natural numbers in decimal notation (Hehner and Horspool 1979). The rules below define the 2-adic operations by means of a GSOS specification

$$\rho : \Sigma(Id \times 2 \times -) \Longrightarrow 2 \times (T-)$$

where T is the free monad generated by $\Sigma_{2\text{-adic}}$. In these rules, $\wedge$ and $\oplus$ denote the bit operations of Boolean AND and addition modulo 2, that is, for all $a, b \in 2$: $a \wedge b = 1$ if and only if $a = b = 1$ and $a \oplus b = 1$ if and only if $a \neq b$.

$$\frac{}{[0] \xrightarrow{0} [0]} \qquad \frac{}{[1] \xrightarrow{1} [0]} \qquad \frac{}{\mathtt{X} \xrightarrow{0} [1]}$$

$$\frac{x_0 \xrightarrow{b} x_1 \quad y_0 \xrightarrow{c} y_1}{x_0 + y_0 \xrightarrow{b \oplus c} (x_1 + y_1) + [b \wedge c]} \qquad \frac{x_0 \xrightarrow{b} x_1}{-x_0 \xrightarrow{b} -(x_1 + [b])}$$

$$\frac{x_0 \xrightarrow{b} x_1 \quad y_0 \xrightarrow{c} y_1}{x_0 \times y_0 \xrightarrow{b \wedge c} (x_1 \times y_0) + ([b] \times y_1)}$$

The rule for $+$ shows that carry bits may propagate infinitely to the right, for example,

$$(1, 0, 0, 0, \ldots) + (1, 1, 1, \ldots) = (0, 0, 0, \ldots),$$

and the rule for $\times$ shows that multiplication is done in the usual shift-add manner – see Hansen and Rutten (2010) for further details.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

In order to specify bitstream functions as well as bitstreams, the syntax is extended with a variable $\mathtt{s}$. The *2-adic function expressions* is thus the set $T(\{\mathtt{s}\})$ of $\Sigma_{2\text{-adic}}$-terms freely generated over the single variable $\mathtt{s}$. Algebraically, a term t in $T(\{\mathtt{s}\})$ is evaluated in Γ by interpreting $\mathtt{s}$ as the identity map $\mathrm{id}\colon 2^\omega \to 2^\omega$, and the 2-adic operations as their pointwise extensions to Γ. The fact that we can define the pointwise extensions with a distributive law implies that the coalgebraic semantics of 2-adic function expressions coincides with the algebraic semantics. The pointwise extensions of $[0], [1], \mathtt{X}, +$ and $-$ to the binary Mealy functor $(2 \times -)^2$ are obtained from the simple format in (17) and its monadic version. The pointwise extension of multiplication, however, requires the use of the buffer operations. We will omit the overline notation and simply write $+$ instead of $\bar{+}$, and so on. The typing should be clear from the context.

$$\frac{}{[0] \xrightarrow{a|0} [0]} \qquad \frac{}{[1] \xrightarrow{a|1} [0]} \qquad \frac{}{\mathtt{X} \xrightarrow{a|0} [1]}$$

$$\frac{x_0 \xrightarrow{a|b} x_1 \quad y_0 \xrightarrow{a|c} y_1}{x_0 + y_0 \xrightarrow{a|b\oplus c} (x_1 + y_1) + [b \wedge c]} \qquad \frac{x_0 \xrightarrow{a|b} x_1}{-x_0 \xrightarrow{a|b} -(x_1 + [b])}$$

$$\frac{x_0 \xrightarrow{a|b} x_1 \quad y_0 \xrightarrow{a|c} y_1}{x_0 \times y_0 \xrightarrow{a|b\wedge c} (x_1 \times (a \triangleright y_0)) + ([b] \times y_1)} \qquad \frac{x_0 \xrightarrow{a|b} x_1}{a \triangleright x_0 \xrightarrow{c|b} c \triangleright x_1}$$

Interestingly, it was possible to define the Mealy machine of terms in Hansen and Rutten (2010), without adding the buffer operations, since these can already be expressed in the existing syntax: for every term $\mathtt{t} \in T(\{\mathtt{s}\})$ and every $a \in 2$, there is a term, denoted by $\mathtt{t}(a:\mathtt{s}) \in T(\{\mathtt{s}\})$, such that for all input streams $\alpha \in 2^\omega$, $beh(\mathtt{t}(a:\mathtt{s}))(\alpha) = beh(\mathtt{t})(a:\alpha) = beh(a \triangleright t)(\alpha)$ – see Hansen and Rutten (2010) for more details of the definition of $\mathtt{t}(a:\mathtt{s})$.

Example 7.2. Consider the functor $FX = 1 + X$ from Example 2.3, that is, F-behaviours are the extended natural numbers $\mathbb{N} + \{\omega\}$, and F^A-behaviours are non-empty, prefix-closed languages (*cf.* Examples 2.3 and 2.4). The operations max ($\vee$), plus ($+$) and times ($\cdot$) on the extended natural numbers (*cf.* Example 4.4) are induced by the following rules:

$$\frac{x_0 \longrightarrow x_1 \quad y_0 \longrightarrow y_1}{x_0 \vee y_0 \longrightarrow x_1 \vee y_1} \qquad \frac{x_0 \longrightarrow x_1 \quad y_0 \longrightarrow \bot}{x_0 \vee y_0 \longrightarrow x_1} \qquad \frac{x_0 \longrightarrow \bot \quad y_0 \longrightarrow y_1}{x_0 \vee y_0 \longrightarrow y_1}$$

$$\frac{x_0 \longrightarrow x_1}{x_0 + y \longrightarrow x_1 + y} \qquad \frac{x_0 \longrightarrow \bot \quad y_0 \longrightarrow y_1}{x_0 + y_0 \longrightarrow y_1} \qquad \frac{x_0 \longrightarrow \bot \quad y_0 \longrightarrow \bot}{x_0 + y_0 \longrightarrow \bot}$$

$$\frac{x_0 \longrightarrow x_1 \quad y_0 \longrightarrow y_1}{x_0 \cdot y_0 \longrightarrow (x_1 \cdot y_0) + y_1} \qquad \frac{x_0 \longrightarrow \bot}{x_0 \cdot y \longrightarrow \bot} \qquad \frac{y_0 \longrightarrow \bot}{x \cdot y_0 \longrightarrow \bot}$$

The pointwise extensions of these operations to the final F^A-coalgebra are defined by the following set of rules (we again omit the overline notation for the pointwise extended

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

operations):

$$\frac{x_0 \xrightarrow{a} x_1}{a \triangleright x_0 \xrightarrow{b} b \triangleright x_1} \qquad \frac{x_0 \xrightarrow{a} \bot}{a \triangleright x_0 \xrightarrow{b} \bot}$$

$$\frac{x_0 \xrightarrow{a} x_1 \quad y_0 \xrightarrow{a} y_1}{x_0 \vee y_0 \xrightarrow{a} x_1 \vee y_1} \qquad \frac{x_0 \xrightarrow{a} x_1 \quad y_0 \xrightarrow{a} \bot}{x_0 \vee y_0 \xrightarrow{a} x_1} \qquad \frac{x_0 \xrightarrow{a} \bot \quad y_0 \xrightarrow{a} y_1}{x_0 \vee y_0 \xrightarrow{a} y_1}$$

$$\frac{x_0 \xrightarrow{a} x_1}{x_0 + y \xrightarrow{a} x_1 + (a \triangleright y)} \qquad \frac{x_0 \xrightarrow{a} \bot \quad y_0 \xrightarrow{a} y_1}{x_0 + y_0 \xrightarrow{a} y_1} \qquad \frac{x_0 \xrightarrow{a} \bot \quad y_0 \xrightarrow{a} \bot}{x_0 + y_0 \xrightarrow{a} \bot}$$

$$\frac{x_0 \xrightarrow{a} x_1 \quad y_0 \xrightarrow{a} y_1}{x_0 \cdot y_0 \xrightarrow{a} (x_1 \cdot (a \triangleright y_0)) + y_1} \qquad \frac{x_0 \xrightarrow{a} \bot}{x_0 \cdot y \xrightarrow{a} \bot} \qquad \frac{y_0 \xrightarrow{a} \bot}{x \cdot y_0 \xrightarrow{a} \bot}$$

For example, let $A = \{a, b\}$ and for $n \in \mathbb{N}$, let $\underline{a}^n = \{\varepsilon, a, aa, \ldots, a^n\}$, that is, $\underline{a}^n$ denotes the prefix-closed language generated by a^n. We then have for all $n \in \mathbb{N}$,

$$\begin{aligned} \underline{a}^{n+1} &\xrightarrow{a} \underline{a}^n \\ \underline{a}^0 &\xrightarrow{a} \bot \\ \underline{a}^n &\xrightarrow{c} \bot \text{ if } c \neq a. \end{aligned}$$

Applying the rule for buffer expressions, we find that for all $c, d \in A$,

$$\begin{aligned} a \triangleright \underline{a}^{n+1} &\xrightarrow{c} c \triangleright \underline{a}^n \\ c \triangleright a \triangleright \underline{a}^{n+1} &\xrightarrow{d} d \triangleright c \triangleright \underline{a}^n \\ &\text{etc.} \end{aligned}$$

In general, for all $k \in \mathbb{N}$ and $c_1, \ldots, c_k \in A$,

$$\begin{aligned} c_{k-1} \ldots \triangleright c_1 \triangleright a \triangleright \underline{a}^{n+1} &\xrightarrow{c_k} c_k \triangleright c_{k-1} \ldots \triangleright c_1 \triangleright \underline{a}^n \\ c_{k-1} \ldots \triangleright c_1 \triangleright a \triangleright \underline{a}^0 &\xrightarrow{c_k} \bot. \end{aligned}$$

Let us check the pointwise extension of $+$ on $\underline{a}^3$ and $\underline{a}^2$. For any stream $\alpha \in A^\omega$ that starts with three a's, that is, $\alpha\restriction_3 = a^3$, we have

$$\mathrm{ev}(\langle \alpha, \underline{a}^3 \rangle) + \mathrm{ev}(\langle \alpha, \underline{a}^2 \rangle) = 3 + 2 = 5.$$

We now use the above rules to compute the transitions that can be made by $\underline{a}^3 + \underline{a}^2$ on input $(a, a, a, c_3, c_4, c_5, \ldots)$ where $c_i \in A$, $i = 3, 4, 5, \ldots$, are arbitrary:

$$\begin{aligned} \underline{a}^3 + \underline{a}^2 &\xrightarrow{a} \underline{a}^2 + (a \triangleright \underline{a}^2) \\ &\xrightarrow{a} \underline{a}^1 + (a \triangleright a \triangleright \underline{a}^2) \\ &\xrightarrow{a} \underline{a}^0 + (a \triangleright a \triangleright a \triangleright \underline{a}^2) \\ &\xrightarrow{c_3} (c_3 \triangleright a \triangleright a \triangleright \underline{a}^1) \qquad (\text{ since } \underline{a}^0 \xrightarrow{c_3} \bot) \\ &\xrightarrow{c_4} (c_4 \triangleright c_3 \triangleright a \triangleright \underline{a}^0) \\ &\xrightarrow{c_5} \bot. \end{aligned}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

We see that $\underline{a}^3 + \underline{a}^2$ can make 5 transitions, so we do indeed have

$$\mathrm{ev}(\langle \alpha, \underline{a}^3 + \underline{a}^2 \rangle) = \mathrm{ev}(\langle \alpha, \underline{a}^3 \rangle) + \mathrm{ev}(\langle \alpha, \underline{a}^2 \rangle)$$

for all α such that $\alpha\restriction_3 = a^3$. If now $\alpha = (a, a, b, c_3, c_4, \ldots)$ for arbitrary $c_3, c_4, \ldots \in A$, we find

$$\begin{aligned}
\underline{a}^3 + \underline{a}^2 &\xrightarrow{a} \underline{a}^2 + (a \triangleright \underline{a}^2) \\
&\xrightarrow{a} \underline{a}^1 + (a \triangleright a \triangleright \underline{a}^2) \\
&\xrightarrow{b} (b \triangleright a \triangleright \underline{a}^1) \qquad (\text{ since } \underline{a}^1 \xrightarrow{b} \bot) \\
&\xrightarrow{c_3} (c_3 \triangleright b \triangleright \underline{a}^0) \\
&\xrightarrow{c_4} \bot.
\end{aligned}$$

Hence

$$\mathrm{ev}(\langle \alpha, \underline{a}^3 + \underline{a}^2 \rangle) = 4 = 2 + 2 = \mathrm{ev}(\langle \alpha, \underline{a}^3 \rangle) + \mathrm{ev}(\langle \alpha, \underline{a}^2 \rangle),$$

as desired. Now consider the product $\underline{a}^3 \cdot \underline{a}^2$ and the input stream a^ω:

$$\begin{aligned}
\underline{a}^3 \cdot \underline{a}^2 &\xrightarrow{a} \underline{a}^2 \cdot (a \triangleright \underline{a}^2) + \underline{a}^1 \\
&\xrightarrow{a} (\underline{a}^1 \cdot (a \triangleright a \triangleright \underline{a}^2) + (a \triangleright \underline{a}^1)) + (a \triangleright \underline{a}^1) \\
&\xrightarrow{a} ((\underline{a}^0 \cdot (a \triangleright a \triangleright a \triangleright \underline{a}^2) + (a \triangleright a \triangleright \underline{a}^1)) + (a \triangleright a \triangleright \underline{a}^1)) + (a \triangleright a \triangleright \underline{a}^1) \\
&\xrightarrow{a} ((a \triangleright a \triangleright \underline{a}^0) + (a \triangleright a \triangleright a \triangleright \underline{a}^1)) + (a \triangleright a \triangleright a \triangleright \underline{a}^1) \\
&\xrightarrow{a} (a \triangleright a \triangleright a \triangleright \underline{a}^0) + (a \triangleright a \triangleright a \triangleright a \triangleright \underline{a}^1) \\
&\xrightarrow{a} (a \triangleright a \triangleright a \triangleright a \triangleright \underline{a}^0) \\
&\xrightarrow{a} \bot.
\end{aligned}$$

Hence,

$$\mathrm{ev}(\langle a^\omega, \underline{a}^3 \cdot \underline{a}^2 \rangle) = 6 = \mathrm{ev}(\langle a^\omega, \underline{a}^3 \rangle) \cdot \mathrm{ev}(\langle a^\omega, \underline{a}^2 \rangle).$$

Example 7.3. We now look at the case where $F = \mathcal{P}_\omega$. Recall that F-behaviours are strongly extensional, finitely branching trees, and F^A-behaviours are image-finite processes (*cf.* Examples 2.5 and 2.6). The pointwise extensions to the final F^A-coalgebra of the operations join, composition and interleaving from Example 4.5 are defined by the rules below (we again omit overlines):

buffer:
$$\frac{x_0 \xrightarrow{a} x_1}{a \triangleright x_0 \xrightarrow{b} b \triangleright x_1} \qquad \frac{x_0 \not\xrightarrow{a}}{a \triangleright x_0 \not\xrightarrow{b}}$$

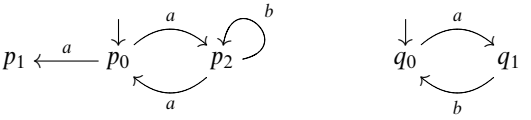
join:
$$\frac{x_0 \xrightarrow{a} x_1}{x_0 \cup y \xrightarrow{a} x_1} \qquad \frac{y_0 \xrightarrow{a} y_1}{x \cup y_0 \xrightarrow{a} y_1}$$

composition:
$$\frac{x_0 \xrightarrow{a} x_1}{x_0 ; y \xrightarrow{a} x_1 ; (a \triangleright y)} \qquad \frac{x \not\xrightarrow{a} \qquad y_0 \xrightarrow{a} y_1}{x ; y_0 \xrightarrow{a} y_1}$$

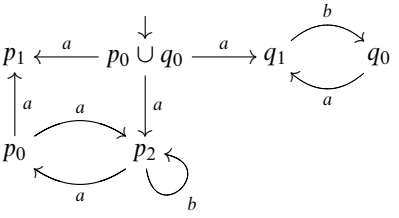
interleaving:
$$\frac{x_0 \xrightarrow{a} x_1 \qquad y_0 \xrightarrow{a} y_1}{x_0 \otimes y_0 \xrightarrow{a} (x_1 \otimes (a \triangleright y_0)) \cup ((a \triangleright x_0) \otimes y_1)}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

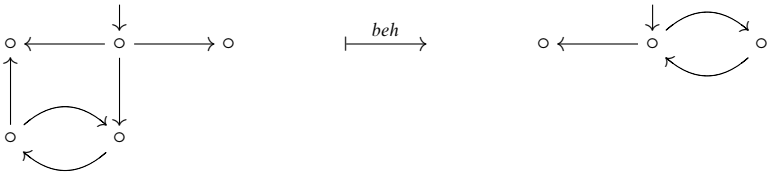
Let us calculate the join of the two processes p_0 and q_0 from Example 4.5. For convenience, their transition diagrams are repeated here (recall that $A = \{a, b\}$):



Applying the rule for $\cup$ to $p_0 \cup q_0$ we find that



Recall that evaluating a process p on a stream α means restricting to all paths in p that are labelled by a prefix in α, and then dropping the labels. Evaluating $p_0 \cup q_0$ at a^ω, we first find the structure on the left in the following diagram, and quotienting with bisimilarity, we get the structure on the right:



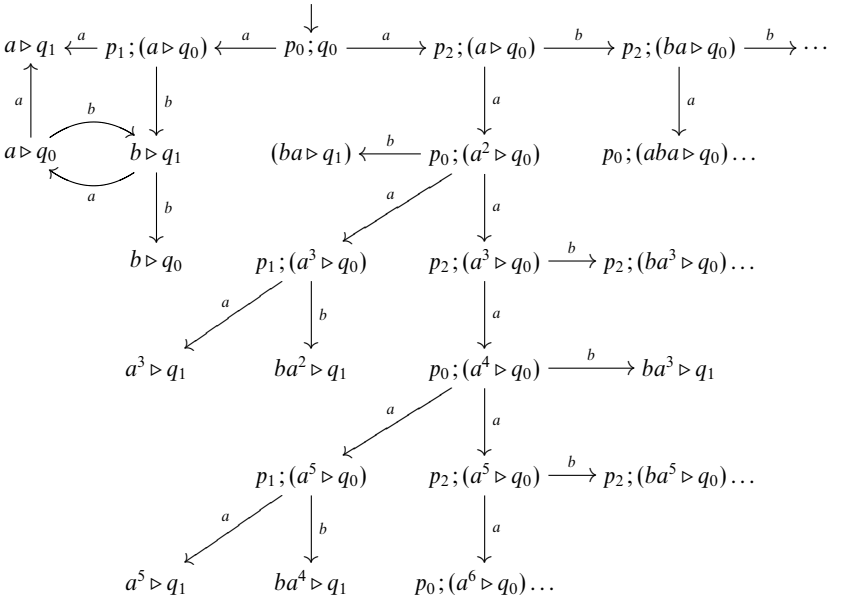
We showed in Example 4.5 that the join of $s_0 = \mathrm{ev}(\langle a^\omega, p_0 \rangle)$ and $x_0 = \mathrm{ev}(\langle a^\omega, q_0 \rangle)$ is equal to s_0. The above shows that

$$\mathrm{ev}(\langle a^\omega, p_0 \cup q_0 \rangle) = \mathrm{ev}(\langle a^\omega, p_0 \rangle) \cup \mathrm{ev}(\langle a^\omega, q_0 \rangle) = s_0.$$

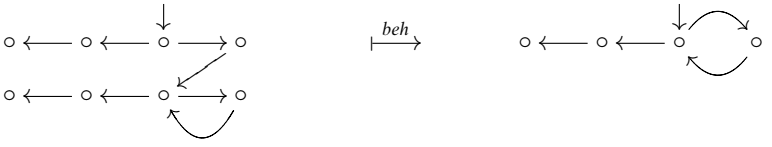
The join was defined in the simple format, so the buffer operations were not needed. However, the rules for composition and interleaving product do use the buffers. We now compute part of the transition structure of $p_0; q_0$. To keep the notation compact, we will write $w \triangleright x$ instead of $a_1 \triangleright \ldots \triangleright a_n \triangleright x$ when $w = a_1 \ldots a_n$. Dots after a node indicate that it

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

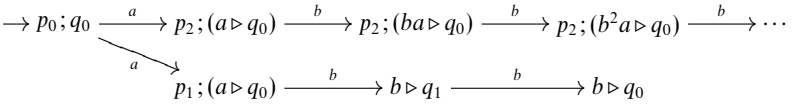
has outgoing transitions that are not shown.



From the above, it is fairly easy to see that evaluating $p_0;q_0$ at a^ω yields a structure bisimilar to the one on the left in the following diagram, and quotienting further with bisimilarity, we get the structure on the right, which is equal to $s_0;x_0$ where $s_0 = \mathrm{ev}(\langle a^\omega, p_0\rangle)$ and $x_0 = \mathrm{ev}(\langle a^\omega, q_0\rangle)$ (*cf.* Example 4.5), hence $\mathrm{ev}(\langle a^\omega, p_0;q_0\rangle) = \mathrm{ev}(\langle a^\omega, p_0\rangle);\mathrm{ev}(\langle a^\omega, q_0\rangle)$:



As another example, we compute the transitions of $p_0;q_0$ on ab^ω:

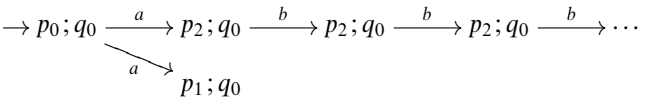


Removing the labels gives a tree that is bisimilar to $t_0;y_0$ where $t_0 = \mathrm{ev}(\langle ab^\omega, p_0\rangle)$ and $y_0 = \mathrm{ev}(\langle ab^\omega, q_0\rangle)$ (*cf.* Example 4.5), so we do indeed have

$$\mathrm{ev}(\langle ab^\omega, p_0;q_0\rangle) = \mathrm{ev}(\langle ab^\omega, p_0\rangle);\mathrm{ev}(\langle ab^\omega, q_0\rangle).$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

Note that the naive approach to defining the operation ; on F^A-behaviours would lead to the following behaviour on ab^ω:



If we remove the labels here, we obtain a tree that is clearly not bisimilar to $t_0 ; y_0$. We leave checking that

$$\mathrm{ev}(\langle \alpha, p_0 \otimes q_0 \rangle) = \mathrm{ev}(\langle \alpha, p_0 \rangle) \otimes \mathrm{ev}(\langle \alpha, q_0 \rangle)$$

for $\alpha \in \{a^\omega, ab^\omega, (ab)^\omega\}$ as an exercise.

8. Conclusions and future work

We have shown how GSOS rules that define operations on F-behaviours can be lifted to GSOS rules that define the corresponding pointwise extensions on F^A-behaviours. The construction is carried out uniformly for all behaviour functors F and relies on extending the signature with a family of buffer operations. Although the proof is technically involved, applying the construction to concrete examples is straightforward. Our original motivating example was the pointwise extension of the 2-adic operations on bitstreams to causal bitstream functions (Mealy machine behaviours) from Hansen and Rutten (2010). In this example, a specification language for bitstreams is turned into one for bitstream functions by adding a variable to the syntax, as in standard mathematical practice. We expect that this 'trick' is useful in other settings. From this perspective, our result can be formulated as: a structural operational semantics for Σ-expressions can be systematically lifted to one for Σ-function expressions in such a way that the operational (coalgebraic) semantics coincides with the algebraic semantics.

As related work, we should mention Jacobs (2006), where it is shown that an Eilenberg–Moore algebra $\sigma : TB \to B$ for a monad T induces a distributive law of T over the (Moore) functor $FX = B \times X^A$, and the Eilenberg–Moore algebra induced on the carrier B^{A^*} of the final F-coalgebra is the pointwise extension of σ. Due to the shape of the functor $FX = B \times X^A$, our present result does not relate directly to that in Jacobs (2006), but it is straightforward to state and prove an analogous one: if $\sigma : TB \to B$ is an Eilenberg–Moore algebra, then σ gives rise to a distributive law of T over the Mealy functor $(B \times -)^A$ such that the induced Eilenberg–Moore algebra $\widehat{\sigma} : T(B^{A^+}) \to B^{A^+}$ is the pointwise extension of σ. Note that causal stream functions $A^\omega \to B^\omega$ are in 1–1 correspondence with functions $A^+ \to B$. In our setting, we can describe the pointwise extension of σ in two steps. First, from σ, it is easy to define a simple distributive law λ of T over the functor $(B \times -)$. The element-wise addition of streams over $\mathbb{R}$ is an example of this. This λ defines an operation σ^ω on B^ω that is the pointwise (that is, element-wise) extension of σ. Then, we apply Theorem 6.1 to lift λ and obtain the pointwise extension $\overline{\sigma^\omega}$ on Γ. It is easy to show that $\overline{\sigma^\omega}$ corresponds to $\widehat{\sigma}$ under the bijection $\Gamma \cong B^{A^+}$.

As mentioned in Section 4, the input evaluation $s \ltimes m$ is a construction very similar to the wreath product of automata (Carton 2000; Pin and Weil 2002; Straubing 1989). The

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

wreath product exists for different types of automata, but the basic idea in all of them is that the output of one automaton becomes the input of the other. Several results in formal language theory have been proved using the wreath product. For example, taking the wreath product of a finite deterministic automaton and a sequential transducer, one easily concludes that sequential functions preserve the regularity of languages under inverse images. Deeper results formulated in terms of the decomposition of semigroups include the Krohn–Rhodes theorem and characterisations of concatenation hierarchies. See Carton (2000), Pin and Weil (2002) and Straubing (1989) for further references to the literature. It would be interesting to study wreath products more generally in a coalgebraic setting, and to explore their applicability in the coalgebraic theory of formal languages (Rutten 2003).

Another observation concerning the $\ltimes$-operation is that it suggests a characterisation of F^A-behaviours as causal functions f from A^ω to F-behaviours. Here we say f is causal if for all $n \in \mathbb{N}$, $\alpha\restriction_n = \beta\restriction_n$ implies that $f(\alpha)$ and $f(\beta)$ are n-step bisimilar. We leave the details of such a characterisation for future work.

Appendix A. Proofs

A.1. *Preliminaries*

We begin with some basic results and definitions that will be useful in the following.

For any functor F, the strength st^F and costrength cs^F (see Section 2.1) correspond to each other through the adjunction $A \times - \dashv (-)^A$. In particular, there is a commuting diagram

$$\begin{array}{ccc} A \times F(X^A) & \xrightarrow{\mathrm{st}^F_{A,X^A}} & F(A \times X^A) \\ {\scriptstyle \mathrm{id}\times \mathrm{cs}^F_{A,X}}\downarrow & & \downarrow{\scriptstyle F\epsilon_X} \\ A \times (FX)^A & \xrightarrow[\epsilon_{FX}]{} & FX. \end{array} \tag{22}$$

Strength respects products: there is

$$\mathrm{st}^F_{A\times B,X} = \mathrm{st}^F_{A,B\times X} \circ (\mathrm{id}_A \times \mathrm{st}^F_{B,X}). \tag{23}$$

Strength transformations compose along functor composition: for endofunctors F and G on **Set**, there is

$$\mathrm{st}^{GF}_{A,X} = G\,\mathrm{st}^F_{A,X} \circ \mathrm{st}^G_{A,FX} \tag{24}$$

for any A and X.

Strength is also natural in the endofunctor F: indeed, for any natural transformation $\alpha : F \Longrightarrow G$, there is

$$\begin{array}{ccc} A \times FX & \xrightarrow{\mathrm{st}^F_{A,X}} & F(A \times X) \\ {\scriptstyle \mathrm{id}\times \alpha_X}\downarrow & & \downarrow{\scriptstyle \alpha_{A\times X}} \\ A \times GX & \xrightarrow[\mathrm{st}^G_{A,X}]{} & G(A \times X). \end{array} \tag{25}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

To simplify the notation, we will sometimes use a natural transformation

$$e : A^\omega \times (-)^A \Longrightarrow A^\omega \times -$$

with the component e_X defined by

$$A^\omega \times X^A \xrightarrow{\langle tl, hd\rangle \times \mathrm{id}} A^\omega \times A \times X^A \xrightarrow{\mathrm{id}\times \epsilon_X} A^\omega \times X. \tag{26}$$

With this, the definition (10) of the map ev can be rewritten as

$$\begin{array}{ccc} A^\omega \times \overline{Z} & \xrightarrow{\mathrm{ev}} & Z \\ \downarrow{\scriptstyle \mathrm{id}\times\overline{\zeta}} & & \\ A^\omega \times (F\overline{Z})^A & & \downarrow{\scriptstyle \zeta} \\ \downarrow{\scriptstyle e_{F\overline{Z}}} & & \\ A^\omega \times F\overline{Z} & & \\ \downarrow{\scriptstyle \mathrm{st}^F_{A^\omega,\overline{Z}}} & & \\ F(A^\omega \times \overline{Z}) & \xrightarrow[F\mathrm{ev}]{} & FZ. \end{array} \tag{27}$$

A.2. *Proof of Theorem 6.1*

Recall that $\overline{\lambda}$ is defined by (17), and $\overline{\sigma}$ is defined from $\overline{\lambda}$ as in (12). Specifically, we have

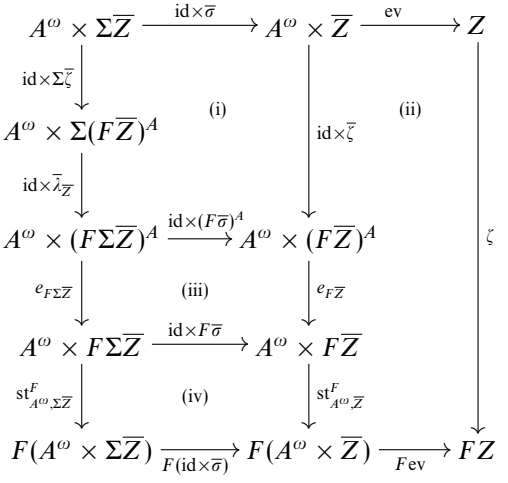
$$\begin{array}{ccccc} \Sigma\overline{Z} & \xrightarrow{\Sigma\overline{\zeta}} & \Sigma(F\overline{Z})^A & \xrightarrow{\overline{\lambda}_{\overline{Z}}} & (F\Sigma\overline{Z})^A \\ \overline{\sigma}\ \vdots & & & & \downarrow{\scriptstyle (F\overline{\sigma})^A} \\ \overline{Z} & & \xrightarrow[\cong]{\overline{\zeta}} & & (F\overline{Z})^A \end{array} \tag{28}$$

We shall prove that (11) commutes. To this end, we define an F-coalgebra structure on $A^\omega \times \Sigma\overline{Z}$:

$$\begin{array}{ccccc} A^\omega \times \Sigma\overline{Z} & \xrightarrow{\mathrm{id}\times\Sigma\overline{\zeta}} & A^\omega \times \Sigma(F\overline{Z})^A & \xrightarrow{\mathrm{id}\times\overline{\lambda}_{\overline{Z}}} & A^\omega \times (F\Sigma\overline{Z})^A \\ & & & & \searrow{\scriptstyle e_{F\Sigma\overline{Z}}} \\ & & A^\omega \times F\Sigma\overline{Z} & \xrightarrow{\mathrm{st}^F_{A^\omega,\Sigma\overline{Z}}} & F(A^\omega \times \Sigma\overline{Z}) \end{array}$$

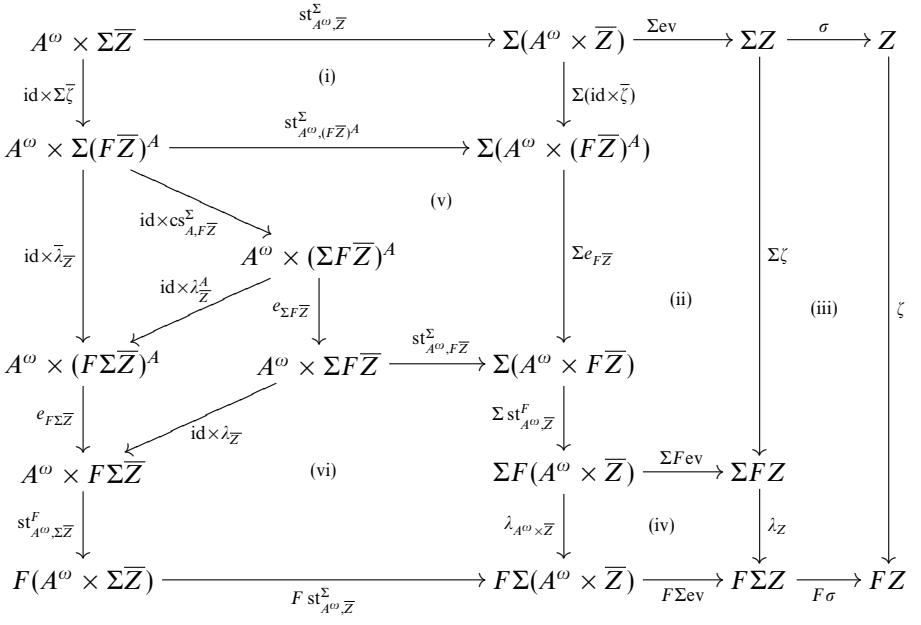
https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

We then check that both sides of (11) are coalgebra morphisms from it to the final F-coalgebra. For the bottom left-hand corner, we check the diagram



where (i) commutes by (28), (ii) by (27), (iii) by naturality of e and (iv) by naturality of st^F.

For the top right-hand corner of (11), we have



https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

where (i) commutes by naturality of st^{Σ}, (ii) by (27), (iii) by (12), (iv) by naturality of λ, (v) follows from (22) and (26) using the naturality of st^{Σ} and (23), and (vi) follows from (24)–(25) – the remaining parts commute by the naturality of e and the definition of $\overline{\lambda}$ in (17). □

A.3. *Proof of Theorem 6.3*

We shall now prove that (19) commutes, recalling that ev is defined by (27) and $\overline{\sigma}$ is defined from $\overline{\rho}$ as in (16). Specifically, we have

$$\begin{array}{ccccc} \overline{\Sigma}\,\overline{Z} & \xrightarrow{\overline{\Sigma}\langle \mathrm{id},\overline{\zeta}\rangle} & \overline{\Sigma}(\overline{Z} \times (F\overline{Z})^A) & \xrightarrow{\overline{\rho}_{\overline{Z}}} & (F\,\overline{T}\,\overline{Z})^A \\ {\scriptstyle\overline{\sigma}}\,\vdots & & & & \downarrow{\scriptstyle (F\overline{\sigma}^{\sharp})^A} \\ \overline{Z} & & \xrightarrow[\cong]{\overline{\zeta}} & & (F\overline{Z})^A, \end{array} \tag{29}$$

where $\overline{\rho}$ is defined by (20) and (21).

In the proof, we shall use an auxiliary natural transformation

$$\theta : A^{\omega} \times \overline{\Sigma} - \Longrightarrow (\Sigma + Id)(A^{\omega} \times -)$$

with θ_X defined by cases of $\overline{\Sigma}$:

— for $\Sigma_{\triangleright}$,

$$A^{\omega} \times A \times X \xrightarrow{\langle tl, hd\rangle^{-1} \times \mathrm{id}} A^{\omega} \times X \hookrightarrow (\Sigma + Id)(A^{\omega} \times X)$$

— for Σ,

$$A^{\omega} \times \Sigma X \xrightarrow{\mathrm{st}^{\Sigma}_{A^{\omega},X}} \Sigma(A^{\omega} \times X) \hookrightarrow (\Sigma + Id)(A^{\omega} \times X)\,.$$

The diagram (19) can be decomposed as

$$\begin{array}{ccccc} A^{\omega} \times \Sigma\overline{Z} & \xrightarrow{\mathrm{st}^{\Sigma}_{A^{\omega},\overline{Z}}} & \Sigma(A^{\omega} \times \overline{Z}) & \xrightarrow{\Sigma \mathrm{ev}} & \Sigma Z \\ \downarrow{\scriptstyle \mathrm{id}\times \iota_{\overline{Z}}} & & \downarrow & & \downarrow \\ A^{\omega} \times \overline{\Sigma}\,\overline{Z} & \xrightarrow{\theta_{\overline{Z}}} & \Sigma(A^{\omega} \times \overline{Z}) + (A^{\omega} \times \overline{Z}) & \xrightarrow{\Sigma \mathrm{ev} + \mathrm{ev}} & \Sigma Z + Z \\ \downarrow{\scriptstyle \mathrm{id}\times\overline{\sigma}} & & & & \downarrow{\scriptstyle [\sigma,\mathrm{id}]} \\ A^{\omega} \times \overline{Z} & & \xrightarrow[\mathrm{ev}]{} & & Z, \end{array} \tag{30}$$

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

where the top left-hand square commutes by the second clause of definition of θ, and the top right-hand square obviously commutes. We shall now prove that the bottom rectangle also commutes.

To this end, consider a map $\gamma : A^\omega \times \overline{\Sigma}\,\overline{Z} \to F(A^\omega \times \overline{T}\,\overline{Z})$ defined by

$$\begin{array}{ccccc} A^\omega \times \overline{\Sigma}\,\overline{Z} & \xrightarrow{\mathrm{id}\times\overline{\Sigma}\langle \mathrm{id},\overline{\zeta}\rangle} & A^\omega \times \overline{\Sigma}(\overline{Z} \times (F\overline{Z})^A) & \xrightarrow{\mathrm{id}\times\overline{\rho}_{\overline{Z}}} & A^\omega \times (F\overline{T}\,\overline{Z})^A \\ & & & & \downarrow{\scriptstyle e_{F\overline{T}\,\overline{Z}}} \\ & & A^\omega \times F\overline{T}\,\overline{Z} & \xrightarrow{\mathrm{st}^F_{A^\omega,\overline{T}\,\overline{Z}}} & F(A^\omega \times \overline{T}\,\overline{Z}). \end{array}$$

Note that in this definition all maps except the first are natural in $\overline{Z}$. This allows us to use a definition/proof principle similar to that used in (14), and infer that both sides of the lower part of (30) are equal if they are both 'maps' from γ to the final F-coalgebra ζ in the following sense:

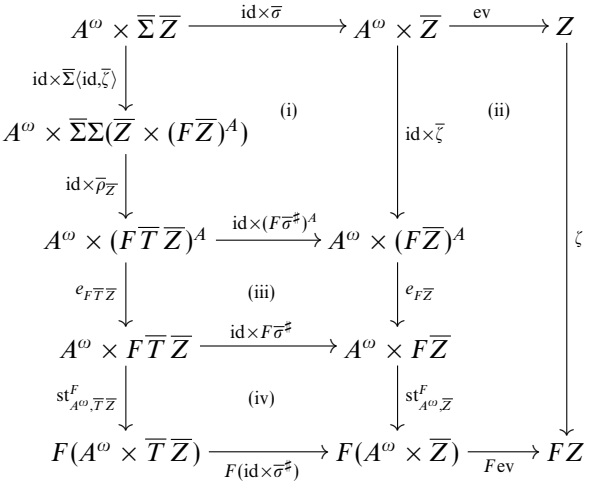
$$\begin{array}{ccccc} A^\omega \times \overline{\Sigma}\,\overline{Z} & \xrightarrow{\mathrm{id}\times\overline{\sigma}} & A^\omega \times \overline{Z} & \xrightarrow{\mathrm{ev}} & Z \\ {\scriptstyle\gamma}\downarrow & & & & \downarrow{\scriptstyle\zeta} \\ F(A^\omega \times \overline{T}\,\overline{Z}) & \xrightarrow[F(\mathrm{id}\times\overline{\sigma}^\sharp)]{} & F(A^\omega \times \overline{Z}) & \xrightarrow[F\mathrm{ev}]{} & FZ \end{array} \tag{31}$$

$$\begin{array}{ccccccc} A^\omega \times \overline{\Sigma}\,\overline{Z} & \xrightarrow{\theta_{\overline{Z}}} & \Sigma(A^\omega \times \overline{Z}) + (A^\omega \times \overline{Z}) & \xrightarrow{\Sigma\mathrm{ev}+\mathrm{ev}} & \Sigma Z + Z & \xrightarrow{[\sigma,\mathrm{id}]} & Z \\ {\scriptstyle\gamma}\downarrow & & & & & & \downarrow{\scriptstyle\zeta} \\ F(A^\omega \times \overline{T}\,\overline{Z}) & \xrightarrow[F\theta^\sharp_{\overline{Z}}]{} & FT(A^\omega \times \overline{Z}) & \xrightarrow[FT\mathrm{ev}]{} & FTZ & \xrightarrow[F\sigma^\sharp]{} & FZ \end{array} \tag{32}$$

where $\theta^\sharp : A^\omega \times \overline{T}- \Longrightarrow T(A^\omega \times -)$ is the obvious inductive extension of θ. The above two conditions can be understood equivalently as two coalgebra morphism diagrams to ζ from an F-coalgebra on $A^\omega \times \overline{T}\,\overline{Z}$, which is easily defined from γ, but they are a bit easier to prove in the present formulation.

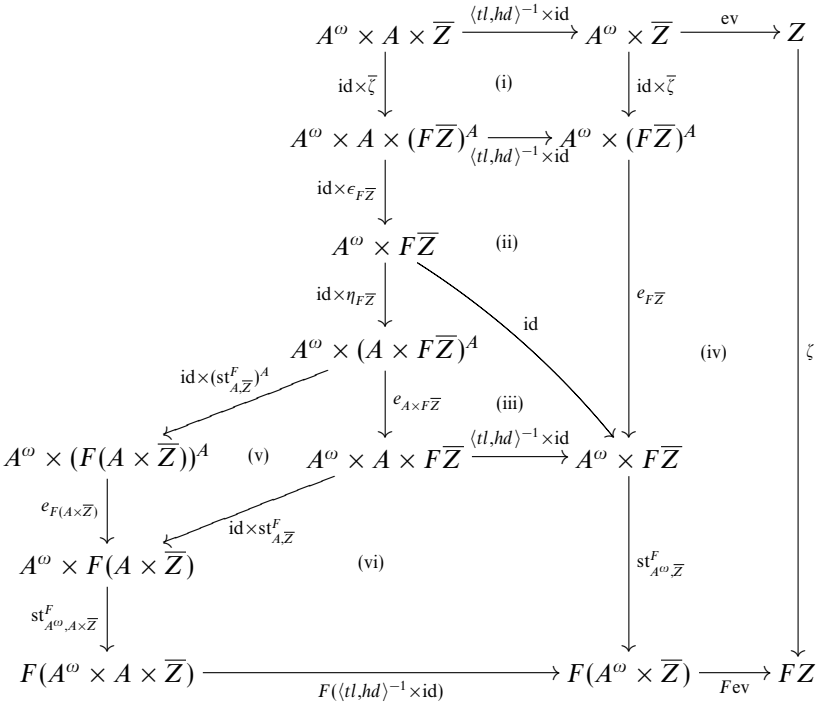
https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

The condition (31) is proved by analogy with the corresponding condition in the proof of Theorem 6.1. So we chase the diagram



where (i) commutes by (29), (ii) by (27), (iii) by naturality of e and (iv) by naturality of st^F. The condition (32) requires more care, and is best proved by case analysis of $\overline{\Sigma}$. We unfold the definition of $\overline{\rho}_{\overline{Z}}$ in the definition of γ, and, recalling the definition of θ by cases:

— for $\Sigma_\triangleright$ we chase the diagram

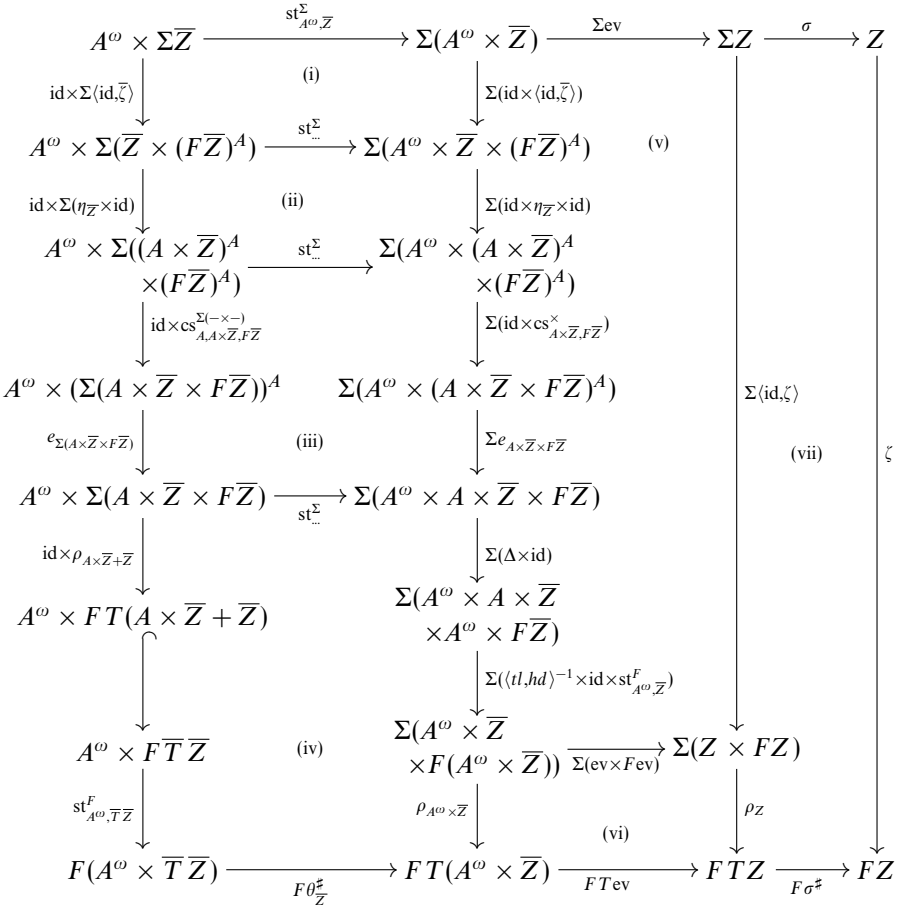


where (i) commutes trivially, (ii) and (iii) commute by the definition of e, (iv) commutes

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

by (27), (v) commutes by the naturality of e and (vi) commutes by (23) and the naturality of st^F.

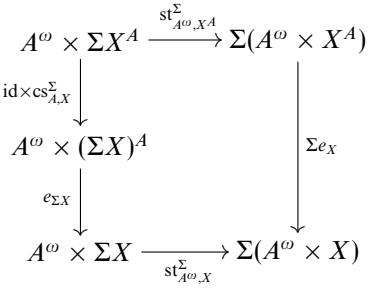
— for Σ, we chase the diagram



(33)

where in the top right-hand corner of (iv), $\Delta : A^\omega \to A^\omega \times A^\omega$ is the diagonal map, with rearrangement of products left implicit, (i) and (ii) commute by the naturality of st^Σ, (vi) commutes by naturality of ρ and (vii) commutes by (14). We still need to show that (iii)–(v) also commute.

Proof that rectangle (iii) in (33) *commutes.* First, for any X, the following diagram commutes by (22), by the definition (26) of e and by (23):



https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

For $X = U \times V$, this becomes the bottom part of

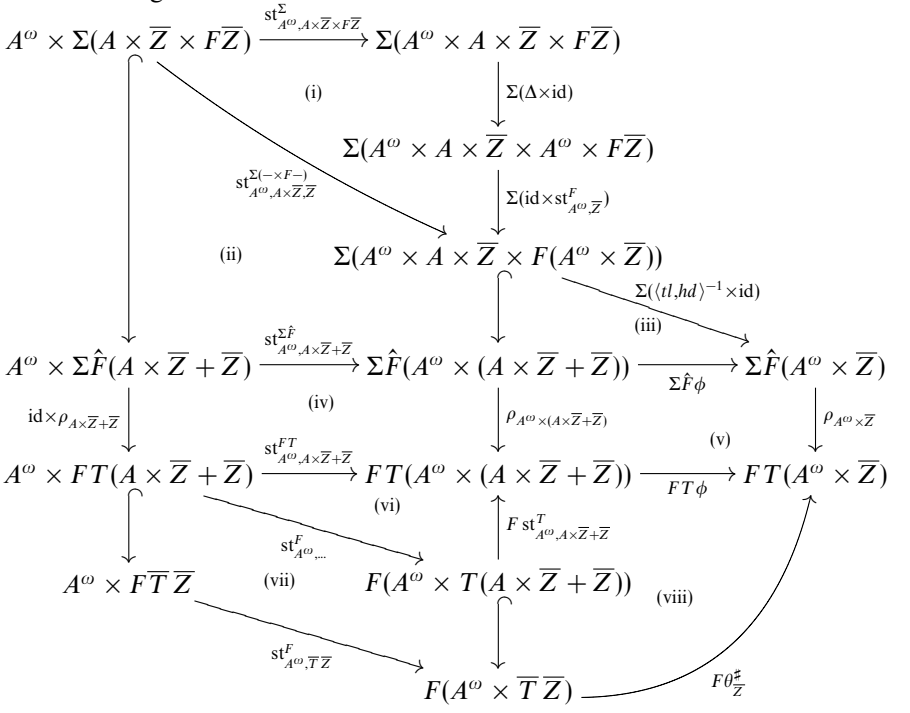
$$\begin{array}{ccc}
A^\omega \times \Sigma(U^A \times V^A) & \xrightarrow{\mathrm{st}^\Sigma_{A^\omega, U^A \times V^A}} & \Sigma(A^\omega \times U^A \times V^A) \\
\downarrow{\scriptstyle \mathrm{id} \times \Sigma\, \mathrm{cs}^\times_{U,V}} & & \downarrow{\scriptstyle \Sigma(\mathrm{id} \times \mathrm{cs}^\times_{U,V})} \\
A^\omega \times \Sigma(U \times V)^A & \xrightarrow{\mathrm{st}^\Sigma_{A^\omega, (U\times V)^A}} & \Sigma(A^\omega \times (U \times V)^A) \\
\downarrow{\scriptstyle \mathrm{id} \times \mathrm{cs}^\Sigma_{A, U\times V}} & & \downarrow{\scriptstyle \Sigma e_{U\times V}} \\
A^\omega \times (\Sigma(U \times V))^A & & \\
\downarrow{\scriptstyle e_{\Sigma(U\times V)}} & & \\
A^\omega \times \Sigma(U \times V) & \xrightarrow[\mathrm{st}^\Sigma_{A^\omega, U\times V}]{} & \Sigma(A^\omega \times (U \times V)),
\end{array}$$

where the top part commutes by naturality of st^Σ. Substituting $U = A \times \overline{Z}$ and $V = F\overline{Z}$, we obtain (iii) of (33). □

Proof that rectangle (iv) in (33) *commutes.* We use the denotation $\hat{F} = Id \times F$ and define a function $\phi : A^\omega \times (A \times \overline{Z} + \overline{Z}) \to A^\omega \times \overline{Z}$ by

$$A^\omega \times (A \times \overline{Z} + \overline{Z}) \xrightarrow{\cong} A^\omega \times A \times \overline{Z} + A^\omega \times \overline{Z} \xrightarrow{[\langle tl, hd\rangle^{-1}, \mathrm{id}]} A^\omega \times \overline{Z}.$$

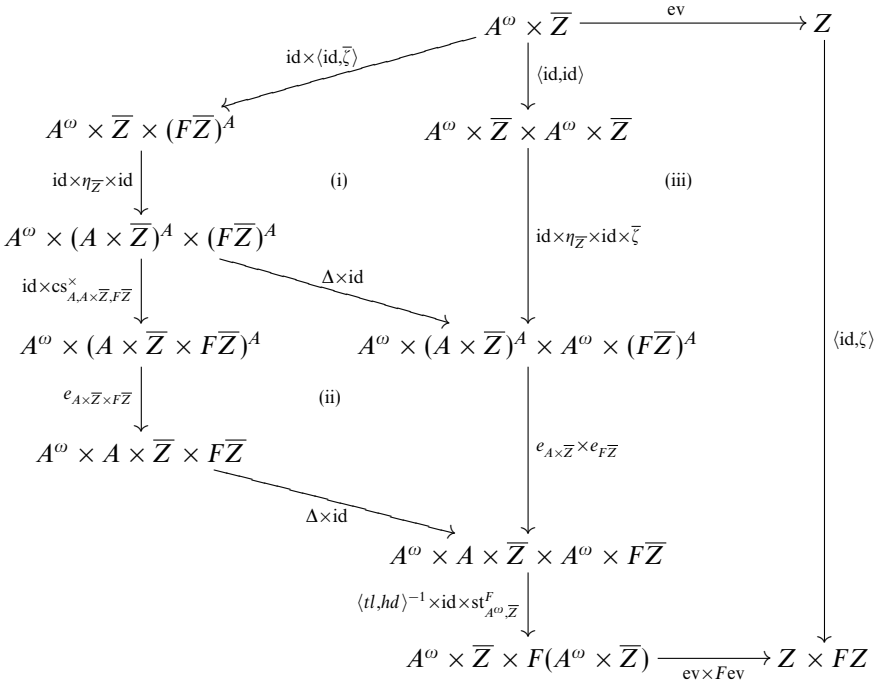
We chase the diagram



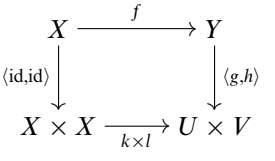
https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

where (i) commutes by (24) (note that $\Delta \times \mathrm{id} = \mathrm{st}^{\times}_{A^\omega, A\times\overline{Z}, F\overline{Z}}$), (ii) commutes by (25) and by the naturality of the two inclusions, (iii) commutes by the definition of ϕ, (iv) commutes by (25), (v) commutes by the naturality of ρ, (vi) commutes by (24), (vii) commutes by the naturality of st^F and (viii) follows easily from the definition of θ and $\theta^\sharp$, once it is noticed that st^T is the inductive extension of st^Σ. The outer shape of this diagram is (iv) of (33). □

Proof that rectangle (v) in (33) *commutes.* First, we chase the diagram

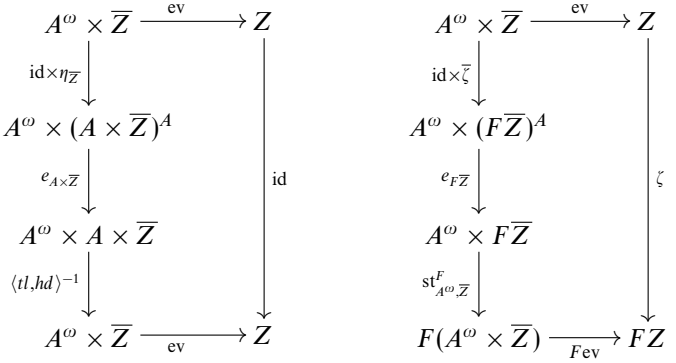


whose outer shape, mapped along Σ, is (v) of (33). Here, (i) commutes trivially and (ii) commutes by the definition (26) of e and by (22). Rectangle (iii) has the general form



https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

(put $X = A^\omega \times \overline{Z}$, $Y = U = Z$, $V = FZ$), which commutes if and only if $k = g \circ f$ and $l = h \circ f$, so it is enough to check that the following two diagrams commute:



It is easy to show that the diagram on the left-hand side commutes by the definition (26) of e, and the one on the right-hand side is (27). □

This completes the proof of Theorem 6.3. □

Acknowledgments. The authors are grateful to an anonymous referee for thorough comments, and for spotting several mistakes in a previous version of this paper.

References

Aceto, L., Fokkink, W. J. and Verhoef, C. (2001) Structural operational semantics. In: Bergstra, J. A., Ponse, A. and Smolka, S. (eds.) *Handbook of Process Algebra*, Elsevier 197–292.

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

Bartels, F. (2004) *On Generalised Coinduction and Probabilistic Specification Formats*, Ph.D. thesis, Vrije Universiteit Amsterdam.

Carton, O. (2000) Wreath product and infinite words. *Journal of Pure and Applied Algebra* **153** 129–150.

Hansen, H. and Klin, B. (2010) Pointwise extensions of GSOS-defined operations (extended abstract). In: CMCS 10 – Short Contributions. CWI Technical Report, SEN-1004, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands 10–11.

Hansen, H. H. and Rutten, J. J. M. M. (2010) Symbolic synthesis of Mealy machines from arithmetic bitstream functions. *Scientific Annals of Computer Science* **XX** 97–130.

Hehner, E. and Horspool, R. (1979) A new representation of the rational numbers for fast easy arithmetic. *SIAM Journal on Computing* **8** 124–134.

Jacobs, B. (2006) A bialgebraic review of deterministic automata, regular expressions and languages. In: Futatsugi, K., Jouannaud, J.-P. and Meseguer, J. (eds.) Algebra, Meaning and Computation: Essays dedicated to Joseph A. Goguen on the Occasion of his 65th Birthday. *Springer-Verlag Lecture Notes in Computer Science* **4060** 375–404.

Klin, B. (2009) Bialgebraic methods and modal logic in structural operational semantics. *Information and Computation* **207** 237–257.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press

Koblitz, N. (1984) *P-adic numbers, p-adic Analysis, and Zeta-Functions*, second edition, Graduate Texts in Mathematics, Springer-Verlag.

Lambek, J. (1968) A fixpoint theorem for complete categories. *Mathematische Zeitschrift* **103** (2) 151–161.

Lenisa, M., Power, J. and Watanabe, H. (2004) Category theory for operational semantics. *Theoretical Computer Science* **327** (1-2) 135–154.

Pin, J.-E. and Weil, P. (2002) The wreath product principle for ordered semigroups. *Communications in Algebra* **30** 5677–5713.

Rutten, J. (1991) Coalgebra, concurrency and control. CWI Research Report SEN-9921, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

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

Rutten, J. (2003) Behavioural differential equations: a coinductive calculus of streams, automata and power series. *Theoretical Computer Science* **308** (1) 1–53.

Rutten, J. (2006) Algebraic specification and coalgebraic synthesis of Mealy machines. In: Proceedings of the 2nd International Workshop on Formal Aspects of Component Software (FACS 2005). *Electronic Notes in Theoretical Computer Science* **160** 305–319.

Rutten, J. and Turi, D. (1994) Initial algebra and final coalgebra semantics for concurrency. In: de Bakker, J., de Roever, W.-P. and Rozenberg, G. (eds.) A Decade of Concurrency, Reflections and Perspectives, REX School/Symposium. *Springer-Verlag Lecture Notes in Computer Science* **803** 530–582.

Straubing, H. (1989) The wreath product and its applications. In: Formal Properties of Finite Automata and Applications. *Springer-Verlag Lecture Notes in Computer Science* **386** 15–24.

Turi, D. and Plotkin, G. (1997) Towards a mathemathical operational semantics. In: *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (LICS 1997)*, IEEE Computer Society 280–291.

Worrell, J. (2005) On the final sequence of a finitary set functor. *Theoretical Computer Science* **338** 184–199.

https://doi.org/10.1017/S096012951000054X Published online by Cambridge University Press