Coalgebraic Semantics for Timed Processes

Kick, M., Power, J., & Simpson, A. (2006). Coalgebraic Semantics for Timed Processes. Information and Computation, 204(4), 588-609doi: 10.1016/j.ic.2005.11.003

Coalgebraic Semantics for Timed Processes

Marco Kick, John Power [1,*], Alex Simpson [2]

Laboratory for the Foundations of Computer Science,
University of Edinburgh, King's Buildings, Edinburgh, EH9 3JZ, Scotland.

Abstract

We give a coalgebraic formulation of timed processes and their operational semantics. We model time by a monoid called a "time domain", and we model processes by "timed transition systems", which amount to partial monoid actions of the time domain or, equivalently, coalgebras for an "evolution comonad" generated by the time domain. All our examples of time domains satisfy a partial closure property, yielding a distributive law of a monad for total monoid actions over the evolution comonad, and hence a distributive law of the evolution comonad over a dual comonad for total monoid actions. We show that the induced coalgebras are exactly timed transition systems with delay operators. We then integrate our coalgebraic formulation of time qua timed transition systems into Turi and Plotkin's formulation of structural operational semantics in terms of distributive laws. We combine timing with action via the more general study of the combination of two arbitrary sorts of behaviour whose operational semantics may interact. We give a modular account of the operational semantics for a combination induced by that of each of its components. Our study necessitates the investigation of products of comonads. In particular, we characterise when a monad lifts to the category of coalgebras for a product comonad, providing constructions with which one can readily calculate.

Key words: time domains, timed transition systems, evolution comonads, delay operators, structural operational semantics, modularity, distributive laws

1 Introduction

The goal of this paper is to analyse the timed behaviour of processes with an eye towards integrating it into the study of coalgebra and its body of theory.

* Corresponding author.

[1] Supported by EPSRC grant GR/586372/01: A Theory of Effects for Programming Languages.

[2] Supported by an EPSRC Advanced Research Fellowship.

Such a coalgebraic formulation would and does allow uniform treatments of bisimulation [6], of operations on processes [21], and of operational semantics [21].

We first consider the extent to which timed behaviour alone can be expressed in coalgebraic terms. This involves a succession of concepts. First, a *time domain* is a monoid $(\mathcal{T}, +, 0)$ subject to two conditions (see Definition 2.1). A motivating example is given by the set of natural numbers with addition. A *timed transition system* is then a labelled transition system $(P, T, \rightsquigarrow)$, where P is a set of processes, T is a *time domain*, and $\rightsquigarrow \subseteq P \times T \times P$ is a *time transition relation*, i.e., it satisfies axioms of determinacy, zero-delay, and continuity. The concept of timed transition system was at the heart of the first author's thesis [11], was summarised in [10], and was synthesised from various accounts of time in the literature, such as [7,17,22]. The central result of Section 2 is that a timed transition system amounts exactly to a coalgebra for what we call the evolution comonad $E_{\mathcal{T}}$ on **Set** generated by the time domain $\mathcal{T}$. The evolution comonad has a natural and succinct description. When the time domain is the set of natural numbers with addition, $E_{\mathcal{T}}$ is the cofree comonad on an endofunctor. In general, however, the evolution comonad is not cofreely generated.

We then investigate, in coalgebraic terms, how time interacts with a concept of delay. This involves the formulation of a notion of *delay operator* (see Definition 3.5), reflecting the natural properties of delay in a timed setting. In order to define the concept at all, we need to define *closedness*, more generally *partial closedness*, for a time domain, and we need some analysis of partially closed timed domains, as well as checking that our main examples of time domains are partially closed.

These definitions in hand, in Section 4 we prove that timed transition systems together with a delay operator amount to the bialgebras for a distributive law of the monad for total left $\mathcal{T}$-actions for a time domain $\mathcal{T}$ over the evolution comonad $E_{\mathcal{T}}$. The category of $\mathcal{T}$-actions is not only algebraic over **Set** but also coalgebraic over **Set**, with comonad $(-)^{\mathcal{T}}$. A subtle use of Currying allows us to reformulate the distributive law of the monad $\mathcal{T} \times (-)$ over the comonad $E_{\mathcal{T}}$ as a distributive law of $E_{\mathcal{T}}$ over the comonad $(-)^{\mathcal{T}}$ and then to see the bialgebras as coalgebras for the induced composite comonad.

In Section 5, we incorporate time with action. To do so, we introduce and study the notion of *heterogeneous transition systems*. Given a finite set A of actions, a time domain $\mathcal{T}$ and a set P, an heterogeneous transition system $(P, A, T, \rightarrow, \rightsquigarrow)$ on P is given by

- an image-finite transition system $(P, A, \rightarrow)$ and
- a timed transition system $(P, T, \rightsquigarrow)$

The first transition system is a coalgebra for an endofunctor B on **Set** [6]. Thus, by the above, if D is the cofree comonad on B, we have a pair of comonads D and D' on **Set** together with a D-coalgebra structure and a D'-coalgebra structure on the same set. So we consider how to combine such comonads. If the product $D \times D'$ of comonads exists, it is the combined comonad we require. But products of comonads do not always exist, and when they do exist, they are typically awkward to calculate. So we give simple general conditions that imply existence, and we characterise the product in terms with which one can readily calculate, providing that one of the comonads is cofree on an endofunctor. There is an important special case, where D' too is cofree on an endofunctor B', e.g., when the time domain is the set of natural numbers. When both comonads are cofree on endofunctors, life becomes simpler, as the product of comonads is then the cofree comonad on the product of endofunctors, which in turn is given pointwise.

In Section 6, following the work of Turi, Plotkin and later authors on distributive laws [14,15,18,19,21], we study the combination of operational semantics generated by two sorts of behaviour, our leading class of examples having one sort of behaviour generated by time with the other sort of behaviour generated by action. Time, as well as being of fundamental interest in its own right, illustrates some, albeit not all, of the intricacies involved with combining operational behaviours in general. An important delicacy that arises with time is as follows [11]: it is not always the case that one has independent pairs of behaviour, i.e., one might not have distributive laws

$$TD \Rightarrow DT$$

and

$$TD' \Rightarrow D'T$$

that one seeks to combine into one of the form

$$T(D \times D') \Rightarrow (D \times D')T$$

That simple situation sometimes does appear in practice, so we do address it. But time behaviour typically interacts with action behaviour: one most generally starts with data of the form

$$T(D \times D') \Rightarrow DT$$

and

$$T(D \times D') \Rightarrow D'T$$

rather than with a pair of distributive laws. So the bulk of Section 6, which is devoted to the derivation of a combined operational semantics, gives necessary and sufficient conditions for the combination, necessarily allowing for the possibility of parametrised starting data.

Although the results in this paper give a fairly comprehensive account of the most fundamental way of combining different forms of transition system, namely taking products, there are situations in which it is appropriate to consider more sophisticated methods of combination in which, for example, non-trivial distributivity or commutativity relations between behviours are incorporated. The investigation of such interactions is left as an interesting task for future research.

We also do not explicitly consider bisimulation for time in this paper. For this, the interested reader is referred to the first author's thesis [11], where it is seen to follow routinely from the analysis here. We also refer to the thesis for combined rule formats in special cases. We regard the work of this paper as a natural development of [18,15], which was a first attempt to take Turi and Plotkin's *definition* of a mathematical operational semantics and start to develop a *theory* of mathematical operational semantics based on it. This paper is an extended version of the CMCS 2004 workshop paper [12] by the first two authors, extended primarily by incorporation of the work of the third author in his invited talk at the same workshop.

2 Timed transition systems

In this section, we briefly develop an account of timed processes as explained more fully in the first author's thesis [11] and in a conference paper summarising part of the thesis [10]. Our analysis is consistent with and generalises much of the literature on time, for instance [7,17,22].

The primary feature of a timed process is that it may evolve, under the passage of time, to another process. We write such *timed transitions*

$$p \overset{t}{\rightsquigarrow} p',$$

where p, p' are processes and t is the time taken by the evolution. To cater for different possible notions of time, we shall ask for t to be an element of a *time domain*, a notion which captures relevant abstract properties of time.

Definition 2.1 (Time domain) A *time domain* is a (not necessarily commutative) monoid $(\mathcal{T}, +, 0)$ satisfying axioms of irreversibility and left-cancellation as follows:

$$(\forall s, t, u \in \mathcal{T}).\ s = s + t + u \Rightarrow s = s + t$$

$$(\forall s, t, u \in \mathcal{T}).\ s + t = s + u \Rightarrow t = u.$$

Note that, in the presence of left-cancellation, the irreversibility axiom is equiv-

alent to

$$(\forall t, u \in \mathcal{T}).\ s + t = 0 \Rightarrow s = t = 0$$

In the above definition, elements of $\mathcal{T}$ are to be understood as representing durations of time. The monoid addition $s+t$ represents the duration s followed immediately by the duration t. Irreversibility states that if s is followed by t to reach a new time $s + t \neq s$, then, whatever further time u is allowed to pass, it always holds that $s + t + u \neq s$, i.e., it is impossible ever to return to s. The left-cancellation property asserts that time is homogeneous in the sense that the future $\{s + t \mid t \in \mathcal{T}\}$ from s looks the same irrespective of s.

One expects any abstract notion of time to come with an associated temporal order. The reason for not including such an order among the primitive data of a time domain is that a natural order can be derived from the monoid structure.

Definition 2.2 (Temporal order) For a monoid $(\mathcal{T}, +, 0)$ define

$$s \leq t \ \Leftrightarrow\ (\exists u \in \mathcal{T}).\ s + u = t.$$

Proposition 2.3 *For any monoid* $(\mathcal{T}, +, 0)$

(1) $\leq$ *is a preorder with minimum element* 0*, and*
(2) for every $s \in \mathcal{T}$*, the function* $s + (\cdot)\colon \mathcal{T} \to \mathcal{T}$ *preserves the order.*

Proposition 2.4 *A monoid* $(\mathcal{T}, +, 0)$ *is a time domain if and only if*

(1) $\leq$ *is a partial order, and*
(2) for every $s \in \mathcal{T}$ *the function* $s + (\cdot)\colon \mathcal{T} \to \mathcal{T}$ *reflects the order.*

As well as directly expressing intuitive general properties of time, as discussed thoroughly in [10,11], the definition of time domain is further motivated by the existence of naturally occurring examples.

Example 2.5 (Discrete time) $\mathcal{T} = \mathbb{N}$. Here 0, $+$ and $\leq$ are all as expected.

Example 2.6 (Real time) $\mathcal{T} = \mathbb{R}_{\geq 0}$, the set of non-negative reals. Again 0, $+$ and $\leq$ are as expected.

Example 2.7 (Qualitative/branching time) $\mathcal{T} = C^*$, the free monoid over a set C. Think of C as a set of independent global clocks, and a word $\alpha \in C^*$ as representing a sequence of ticks from the clocks in the order in which they occur. In the free monoid, 0 is the empty word ε and $+$ is concatenation of words, corresponding to the natural composition of time durations in this context. The derived order is the prefix ordering: $\alpha \leq \beta$ if and only if α is a prefix of β. When $|C| = 1$, i.e., for a single clock, qualitative time is equivalent

to discrete time. When $|C| \geq 2$, in contrast to the previous examples, C^* is not a commutative monoid and $\leq$ is not a total order.

Example 2.8 (Product/local time) If $\{\mathcal{T}_i\}_{i\in I}$ is a family of time domains, then $\prod_{i\in I}\mathcal{T}_i$ is a time domain under the pointwise monoid structure. The derived order is also pointwise. Product time domains naturally model local time, where each $i \in I$ models a clock corresponding to the time domain $\mathcal{T}_i$, local in the sense that the actions of different clocks do not interfere with each other, e.g., with each clock influencing a distinct component of the system being modelled. When $|I| \geq 2$ and the time domains are non-trivial, the temporal order is not total.

We now introduce the notion of *timed transition system*, which captures intuitive properties of the evolution of processes under time.

Definition 2.9 (Timed transition system) A *timed transition system* over a time domain $\mathcal{T}$ is a labelled transition system $(P, \rightsquigarrow)$ where P is a set of *processes*, and $\rightsquigarrow \subseteq P \times \mathcal{T} \times P$ is called the *time transition relation*, satisfying axioms of determinacy, zero-delay and continuity, which, respectively, are given by

$$p \overset{t}{\rightsquigarrow} p' \wedge p \overset{t}{\rightsquigarrow} p'' \Rightarrow p' = p''$$

$$p \overset{0}{\rightsquigarrow} p$$

$$p \overset{t+u}{\rightsquigarrow} p' \Leftrightarrow (\exists p''). \, p \overset{t}{\rightsquigarrow} p'' \overset{u}{\rightsquigarrow} p'$$

The key point to observe here is that this is *only* about time: it does not involve any other possible behaviour. Rather, our strategy is first to study properties of timed behaviour in isolation, and then to consider how timed behaviour interacts with other forms of behaviour such as nondeterministic behaviour. This approach motivates, in particular, the determinacy axiom. The idea is that nondeterminism in computation needs to be specifically triggered and should not merely be a by-product of the evolution of a process under time. Later, we shall model nondeterministic timed processes by combining deterministic timed behaviour as above with explicit nondeterministic behaviour. Such an approach to nondeterminism is standard in the literature on timed processes. For further discussion, see [10,11].

One of the basic observations of [10,11] is that timed transition systems over $\mathcal{T}$ are equivalent to partial actions of the monoid $\mathcal{T}$. To see this, with more elegant proofs, we proceed as follows. We denote a partial function from X to Y by $X \rightharpoonup Y$, and we denote Kleene equality of terms t and t', i.e., t being defined if and only if t' is, and in that case their being equal, by $t \simeq t'$.

Definition 2.10 Given a monoid $(M, +, 0)$ and a set X, a *partial right monoid action* of M on X is a partial function $* : X \times M \rightharpoonup X$ such that for all x in

X and m, n in M,

$$x * 0 \;=\; x$$
$$x * (m + n) \;\simeq\; x * m * n\,.$$

Proposition 2.11 *To give a timed transition system $(P, \rightsquigarrow)$ over $\mathcal{T}$ is to give a partial right $\mathcal{T}$-action $*$ on P, with the equivalence given by*

$$p \overset{t}{\rightsquigarrow} p' \;\;\Leftrightarrow\;\; p' \simeq p * t$$

In view of this result, we henceforth consider timed transition systems and partial right actions as being interchangeable.

There are several possible ways to make partial M-actions into a category $\mathbf{pAct}_M$, but the one that proves most useful in this setting is by defining a map from $(X, *)$ to $(X', *')$ to be a total function $f : X \to X'$ such that for all x in X and m in M

$$f(x * m) \;\simeq\; f(x) *' m$$

The central result of [11] is Theorem 4.1, which asserts that for a time domain $\mathcal{T}$, the category $\mathbf{pAct}_{\mathcal{T}}$, which, by Proposition 2.11, amounts to a category of timed transition systems, is comonadic over **Set** with comonad given as follows.

Definition 2.12 (Evolution comonad) Given a time domain $(\mathcal{T}, +, 0)$ and a set X, a $\mathcal{T}$-*evolution* is partial function $e : \mathcal{T} \rightharpoonup X$ satisfying the following two axioms:

$$e(0)\downarrow$$
$$(\forall t, u \in \mathcal{T}).\, e(t + u)\downarrow \Rightarrow\; e(t)\downarrow$$

We denote the set of all $\mathcal{T}$-evolutions on X by $E_{\mathcal{T}}X$, omitting the subscript $\mathcal{T}$ when no confusion can arise. This extends to a functor on **Set** sending $f : X \to X'$ to $Ef : EX \to EX'$ defined by

$$E(f)(e)(t) \;\simeq\; f(e(t))\,.$$

The functor extends to a comonad with counit $\epsilon : E \Rightarrow Id$ and comultiplication $\delta : E \Rightarrow E^2$ defined as follows:

$$\epsilon(e) \;=\; e(0)$$

$$\delta(e)(s)\downarrow \;\;\text{whenever}\;\; e(s)\downarrow$$
$$\delta(e)(s)(t) \;\simeq\; e(s + t)\,.$$

Theorem 2.13 ([11, Theorem 4.1]) *For any time domain $(\mathcal{T}, +, 0)$, the forgetful functor $\mathbf{pAct}_{\mathcal{T}} \to \mathbf{Set}$ is comonadic, with comonad given by E_T.*

Given an endofunctor B on a category C, if the forgetful functor $U : B\text{-}Coalg \longrightarrow C$ has a right adjoint, we call the induced comonad the cofree comonad on B. In general, for example in the case of real time $\mathcal{T} = \mathbb{R}_{\geq 0}$, the comonad $E_{\mathcal{T}}$ does not appear to be cofreely generated by an endofunctor. But for discrete and qualitative time, i.e., for Examples 2.5 and 2.7, it is cofreely generated:

Theorem 2.14 (cf. [11, Theorem 4.2],[4]) *The comonad E_{C^*} is cofreely generated by the endofunctor on* **Set** *given by $1 + (C \times -)$. In particular, E_N is the cofree comonad on the endofunctor $1 + -$ on* **Set**.

We shall need one more basic result about the comonad $E_{\mathcal{T}}$.

Proposition 2.15 (cf. [11, Propositions 4.7 & 4.8]) *For any time domain $\mathcal{T}$, the functor E_T preserves pullbacks and is accessible.*

PROOF. The preservation of pullbacks is straightforward (see [11, Proposition 4.7]). For accessibility, first observe that the functor $\mathcal{T} \rightharpoonup (-)$ is accessible as it is isomorphic to $(1 + (-))^{\mathcal{T}}$ and accessible functors on **Set** are closed under limits in general, hence products in particular (see, for instance, [1]). Next, obtain $E_{\mathcal{T}}$ as the equaliser of natural transformations $\rho, \rho' : (\mathcal{T} \rightharpoonup (-)) \Rightarrow 2 \times 2^{\mathcal{T} \times \mathcal{T}}$, where we write 2 for the set $\{\text{true}, \text{false}\}$ of truth values, defined by

$$\rho_X(e) \;=\; (e(0)\!\downarrow,\ \lambda(t,u).\,(e(t+u)\!\downarrow))$$

$$\rho'_X(e) \;=\; (\text{true},\ \lambda(t,u).\,(e(t+u)\!\downarrow \wedge e(t)\!\downarrow)\,.$$

This exhibits $E_{\mathcal{T}}$ as a limit of accessible endofunctors, hence accessible. □

We remark that it follows from the above, see e.g. [5], that $\mathbf{pAct}_{\mathcal{T}}$ is a topos. In fact it can be shown to be a presheaf topos.

3 Delay operators

In this paper, we use coalgebra to provide a principled treatment of operations on timed processes. Most interesting operations concern the interaction of time with other types of behaviour, for example, nondeterministic behaviour. We consider such heterogeneous behaviour in Section 5. But first we address the only natural example we know of an interesting operation on timed processes whose behaviour concerns time alone: the delay operator. The delay operator interacts with timed transition systems in a way that is quite different to that of other operations, as we shall see later.

A *delay operator* on a timed transition system will be a binary operation, mapping a time t and process p to a process $t\,.\,p$, the process that delays for duration t and then proceeds as p. There are intuitive properties that a delay operator should satisfy. First,

$$0\,.\,p \;=\; p$$

$$(s+t)\,.\,p \;=\; s\,.\,t\,.\,p$$

which together say that delay is a total left action of the time domain monoid on processes. Second, one expects, as minimum, the following interaction with process evolution,

$$(t\,.\,p) * t = p$$

These equations, together with the monoid action laws for $*$, have the following two natural consequences:

$$t \leq s \;\Rightarrow\; (s\,.\,p) * t \;=\; (s-t)\,.\,p$$

$$s \leq t \;\Rightarrow\; (s\,.\,p) * t \;\simeq\; p * (t-s)$$

where, if $s \leq t$, we write $t - s$ for the unique u such that $t = s + u$. The last two equations can equivalently be expressed as a single equation

$$(s\,.\,p) * t \;\simeq\; (s \dot{-} t)\,.\,(p * (t \dot{-} s)) \tag{1}$$

by introducing the notation $s \dot{-} t$ for truncated subtraction, which satisfies the following conditions:

$$t \leq s \;\Rightarrow\; s \dot{-} t \;=\; s - t\,,$$
$$s \leq t \;\Rightarrow\; s \dot{-} t \;=\; 0\,.$$

When the temporal order on $\mathcal{T}$ is total, the above properties completely determine the evolution of $s\,.\,p$ to $(s\,.\,p)*t$ for any time t. However, for time domains with a partial temporal order, the value of $(s\,.\,p)*t$ is not yet specified when s and t are incomparable. It turns out that there is an elegant general solution to determining a natural value for $(s\,.\,p) * t$ based on taking equation (1) as the defining property of delay operators, and identifying conditions on a time domain under which there exists a uniquely determined truncated subtraction operation $s \dot{-} t$ that is as well behaved as possible on incomparable values.

To define the general notion of truncated subtraction, we adapt the approach of Lawvere, who observed that, by viewing the partial order $(\mathbb{R}_{\geq 0}, \leq)$ as a category, truncated subtraction $(\cdot) \dot{-} t$ is left adjoint to the functor $t + (\cdot)$ [13]. Thus truncated subtraction shows that, when $+$ is taken as the monoidal product, the category $(\mathbb{R}_{\geq 0}, \geq)$ is monoidal closed: order inversion arises because monoidal closure requires $(\cdot) \dot{-} t$ to be a *right* adjoint; we shall ignore that convention, using the standard order $\leq$ and accepting $(\cdot) \dot{-} t$ as a left adjoint. For a general time domain $\mathcal{T}$, monoidal closure is not the correct abstraction

because $+$ is not generally monoidal: the function $(\cdot) + t$ need not preserve the order, e.g., consider C^* for $|C| \geq 2$. Nevertheless, as observed in Proposition 2.3(2), the function $t + (\cdot)$ does always preserve the order. Thus, one can ask whether the functor $t + (\cdot)$ has a left adjoint. That gives rise to the following definition.

Definition 3.1 (Closed time domain) A time domain $\mathcal{T}$ is *closed* if there exists a function $\dot{-} : \mathcal{T} \times \mathcal{T} \to \mathcal{T}$ satisfying

$$(\forall s, t, u \in \mathcal{T}).\ s \dot{-} t \leq u \ \Leftrightarrow\ s \leq t + u\,.$$

We shall show in Proposition 3.4 that this definition captures intuitive properties of truncated subtraction. But it is not as general as we should like, excluding time domains for qualitative time (Example 2.7). So, for a more general notion, we allow truncated subtraction to be a partial function.

Definition 3.2 (Partially closed time domain) A time domain $\mathcal{T}$ is *partially closed* if there exists a partial function $\dot{-} : \mathcal{T} \times \mathcal{T} \rightharpoonup \mathcal{T}$ satisfying

$$(\forall s, t, u \in \mathcal{T}).\ ((s \dot{-} t)\downarrow \text{ and } s \dot{-} t \leq u) \ \Leftrightarrow\ s \leq t + u\,.$$

This definition can be formulated in category-theoretic terms by introducing a notion of partial left adjoint for the functor $t + (-)$. But we shall not develop that idea here as it distracts from our main line of argument.

A fortiori, every closed time domain is partially closed. It is also easy to see that the value $s \dot{-} t$ is uniquely determined in a (partially) closed time domain as the least u such that $s \leq t + u$. That observation motivates the following simple characterization of (partially) closed time domains in alternative order-theoretic terms.

A subset of a partial order is *bounded* if it has an upper bound. A partial order has *finite bounded joins* if every bounded finite subset has a least upper bound (lub). A partial order with least element has finite bounded joins if and only if, for every s, t with $\{s, t\}$ bounded (denoted $s \uparrow t$), a least upper bound $s \vee t$ exists.

Proposition 3.3 *The following are equivalent for a time domain $\mathcal{T}$:*

(1) $\mathcal{T}$ is closed (resp. partially closed)
(2) $\leq$ has finite joins (resp. finite bounded joins).

PROOF. To show that 1 implies 2, suppose $\mathcal{T}$ is partially closed. For finite bounded joins, suppose $s \uparrow t$. Then there exists u with $s \leq t + u$, so $(s \dot{-} t)\downarrow$.

We show that $t + (s \dot{-} t)$ is the lub of $\{s, t\}$. Trivially, $t \leq t + (s \dot{-} t)$. That $s \leq t + (s \dot{-} t)$ holds follows from Definition 3.2, because $(s \dot{-} t)\downarrow$ and $s \dot{-} t \leq s \dot{-} t$. Thus $t + (s \dot{-} t)$ is an upper bound for $\{s, t\}$. For minimality, suppose $s \leq u \geq t$. Then $s \leq t + (u - t)$. So, by Definition 3.2, $s \dot{-} t \leq u - t$. So $t + (s \dot{-} t) \leq t + (u - t) = u$ as required.

For the converse, suppose that $\mathcal{T}$ has finite bounded joins. Define

$$s \dot{-} t \simeq \begin{cases} (s \vee t) - t & \text{if } s \uparrow t \\ \text{undefined} & \text{otherwise} \end{cases}$$

To verify Definition 3.2, suppose $s \leq t + u$, then $s \uparrow t$, so $(s \dot{-} t)\downarrow$. Then $t + (s \dot{-} t) = t + ((s \vee t) - t) = s \vee t \leq t + u$, because $t + u$ is an upper bound for $\{s, t\}$. So, by order reflection, $(s \dot{-} t) \leq u$. Conversely, suppose $(s \dot{-} t)\downarrow$ and $(s \dot{-} t) \leq u$, i.e. $s \uparrow t$ and $((s \vee t) - t) \leq u$. Then $s \leq s \vee t = t + ((s \vee t) - t) \leq t + u$, as required. $\Box$

It follows routinely from the proposition that all our main examples of time domains are partially closed. In the case of discrete time and real time, i.e., Examples 2.5 and 2.6, the time domains are closed, as are all time domains for which $\leq$ is a total order. In the case of qualitative time, i.e., Example 2.7, when $|C| \geq 2$, the time domain C^* is partially closed but not closed. In that case, $(s \dot{-} t)\downarrow$ if and only if either $s \leq t$ or $t \leq s$. For products, Proposition 3.3 yields a simple proof that if all $\mathcal{T}_i$ are (partially) closed then so is $\prod_{i \in I} \mathcal{T}_i$.

The proposition below summarises the main properties of partially closed time domains. The properties concern the interaction between $\dot{-}$ and the temporal order, properties arising from the relationship between $\dot{-}$ and bounded lubs, and the interaction between $\dot{-}$ and the monoid structure.

Proposition 3.4 *If $\mathcal{T}$ is a partially closed time domain, then*

(1) $s \leq t$ *implies* $s \dot{-} t = 0$.
(2) $t \leq s$ *implies* $s \dot{-} t = s - t$.
(3) $(s \dot{-} t)\downarrow$ *if and only if* $s \uparrow t$ *if and only if* $(t \dot{-} s)\downarrow$.
(4) $s + (t \dot{-} s) \simeq s \vee t \simeq t + (s \dot{-} t)$.
(5) $s \dot{-} 0 = s$.
(6) $s \dot{-} (t + u) \simeq (s \dot{-} t) \dot{-} u$.
(7) $0 \dot{-} s = 0$.
(8) $(s + t) \dot{-} u \simeq (s \dot{-} u) + (t \dot{-} (u \dot{-} s))$.

PROOF. Properties (1) and (2) are easy, and (5) and (7) follow immediately.

Properties (3) and (4) are direct consequences of the proof of Proposition 3.3.

For property (6), suppose $(s\dot{-}(t+u))\downarrow$. Then $s\dot{-}(t+u) \leq s\dot{-}(t+u)$. So, repeatedly applying Definition 3.2, first $s \leq t+u+(s\dot{-}(t+u))$, then $(s\dot{-}t)\downarrow$ and $s\dot{-}t \leq u+(s\dot{-}(t+u))$, whence finally $((s\dot{-}t)\dot{-}u)\downarrow$ and $(s\dot{-}t)\dot{-}u \leq s\dot{-}(t+u)$. A similar argument establishes that if $((s\dot{-}t)\dot{-}u)\downarrow$, then both $(s\dot{-}(t+u))\downarrow$ and $s\dot{-}(t+u) \leq (s\dot{-}t)\dot{-}u$.

Finally, for (8), the desired equation follows by left cancellation from:

$$\begin{array}{ll} u+((s+t)\dot{-}u) & \\ \quad \simeq s+t+(u\dot{-}(s+t)) & \text{by (4)} \\ \quad \simeq s+t+((u\dot{-}s)\dot{-}t) & \text{by (6)} \\ \quad \simeq s+(u\dot{-}s)+(t\dot{-}(u\dot{-}s)) & \text{by (4)} \\ \quad \simeq u+(s\dot{-}u)+(t\dot{-}(u\dot{-}s)) & \text{by (4).} \end{array}$$

□

Definition 3.5 (Delay operator) A delay operator on a time transition system $(P, \rightsquigarrow)$ over $\mathcal{T}$, where $\mathcal{T}$ is a partially closed time domain, is a total left $\mathcal{T}$-action " . " on P satisfying equation (1) above.

4 Delay operators and coalgebra

In this section, we describe a distributive law of the monad for total left $\mathcal{T}$-actions over the evolution comonad $E_{\mathcal{T}}$, then make subtle use of Currying, in order to incorporate delay operators into our coalgebraic analysis of timed transition systems. First we deal with the total-left-action aspect of delay operators.

Proposition 4.1 *The functor $S_{\mathcal{T}} = \mathcal{T} \times (-)$ carries a monad structure with unit $\eta : Id \Rightarrow S$ and multiplication $\mu : S^2 \Rightarrow S$ defined as follows:*

$$\eta(x) \;=\; (0, x)$$

$$\mu(s, (t, x)) \;=\; ((s+t), x).$$

Also, the functor $(-)^{\mathcal{T}}$ carries a dual comonad structure. And $S_{\mathcal{T}}$-Alg, equally $(-)^{\mathcal{T}}$-Coalg, is the category of total left $\mathcal{T}$-actions $(\,.\,) : \mathcal{T} \times X \to X$.

Recall that a *distributive law* of a monad (T, η, μ) over a comonad (D, ϵ, δ) is a natural transformation

$$\lambda : TD \Rightarrow DT$$

subject to commutativity of four diagrams expressing coherence with respect to each of η, μ, ϵ and δ. A λ-*bialgebra* is a pair of maps $h : TX \longrightarrow X$ and $k : X \longrightarrow DX$ such that (X, h) is a T-algebra, (X, k) is a D-coalgebra, and the diagram below commutes.

$$\begin{array}{ccccc} TX & \xrightarrow{h} & X & \xrightarrow{k} & DX \\ {\scriptstyle Tk}\downarrow & & & & \uparrow{\scriptstyle Dh} \\ TDX & & \xrightarrow[\lambda_X]{} & & DTX \end{array} \tag{2}$$

One may similarly define a distributive law of a comonad D over a comonad D' and bicoalgebras for such.

Theorem 4.2 *If $\mathcal{T}$ is a partially closed time domain, the following is a distributive law $\lambda : S_{\mathcal{T}}E \Rightarrow ES_{\mathcal{T}}$ of the monad $S_{\mathcal{T}} = \mathcal{T} \times (-)$ over the comonad E:*

$$\lambda_X(s, e)(t) \;\simeq\; (s \dot{-} t,\, e(t \dot{-} s)).$$

Moreover, λ-bialgebras are exactly pairs

$$(* : X \times \mathcal{T} \rightharpoonup X,\; (\,.\,) : \mathcal{T} \times X \to X)$$

consisting of a timed transition system together with a delay operator.

PROOF. We give a sketch of the proof in order to show how the result follows from Proposition 3.4. First, in order to see that λ is a distributive law, we check the equation

$$E\mu \circ \lambda \circ S_{\mathcal{T}}\lambda = \lambda \circ \mu_E : S_{\mathcal{T}}^2 E \Rightarrow ES\,,$$

which follows by:

$$\begin{array}{lll} E\mu_X(\lambda_X(S\lambda_X(s,(t,e))))(u) & & \\ \quad \simeq E\mu_X(\lambda_X(s,(v \mapsto (t \dot{-} v,\, e(v \dot{-} t)))))(u) & & \text{(def. of } \lambda) \\ \quad \simeq E\mu_X(w \mapsto (s \dot{-} w,\, (t \dot{-} (w \dot{-} s)),\, e((w \dot{-} s) \dot{-} t))))(u) & & \text{(def. of } \lambda) \\ \quad \simeq (w \mapsto ((s \dot{-} w) + (t \dot{-} (w \dot{-} s)),\, e((w \dot{-} s) \dot{-} t)))(u) & & \text{(def. of } \mu) \\ \quad \simeq ((s \dot{-} u) + (t \dot{-} (u \dot{-} s)),\, e((u \dot{-} s) \dot{-} t))) & & \\ \quad \simeq ((s + t) \dot{-} u,\, e(u \dot{-} (s + t))) & & \text{(by Prop. 3.4)} \\ \quad \simeq (w \mapsto ((s + t) \dot{-} w,\, e(w \dot{-} (s + t))))(u) & & \\ \quad \simeq \lambda_X(s + t,\, e)(u) & & \text{(def. of } \lambda) \\ \quad \simeq \lambda_X(\mu_{EX}(s,\, (t,\, e)))(u) & & \text{(def. of } \mu)\,. \end{array}$$

For the bialgebra claim, we know that E-coalgebras are precisely timed transition systems, and S-algebras are precisely total left actions. Thus it just remains to verify that diagram (2) corresponds to equation (1). Suppose then that we have a coalgebra $k : X \longrightarrow EX$, which we shall write as the partial right action "$*$", and an algebra $h : SX \longrightarrow X$, which we shall write as the total left action "$.$". Then, the composite $k \circ h \colon SX \longrightarrow EX$ is by definition

$$(k(h(s,x)))(t) \simeq (s\,.\,x) * t\,.$$

Moreover, we have

$$\begin{aligned}
&(Eh(\lambda_X(Sk(s,x))))(t) \\
&\quad \simeq (Eh(\lambda_X(s,\ (u \mapsto x * u))))(t) \\
&\quad \simeq (Eh(v \mapsto (s \dot{-} v,\ x * (v \dot{-} s))))(t) && \text{(def. of } \lambda) \\
&\quad \simeq (v \mapsto (s \dot{-} v)\,.\,(x * (v \dot{-} s)))(t) \\
&\quad \simeq (s \dot{-} t)\,.\,(x * (t \dot{-} s))\,.
\end{aligned}$$

Thus diagram (2) commutes if and only if equation (1) holds. □

The relationship between the monad $S_{\mathcal{T}} = \mathcal{T} \times (-)$ and the comonad $(-)^{\mathcal{T}}$ as explained in Proposition 4.1 allows us to reformulate this result, by making two uses of Currying, in terms of comonads alone. Recall the following result, explored for instance in [19]:

Proposition 4.3 *The following are equivalent:*

(1) a distributive law $\lambda : TD \Rightarrow DT$ of a monad T over a comonad D
(2) a lifting of the comonad D to a comonad D_T on T-Alg.
(3) a lifting of the monad T to a monad T_D on D-Coalg

Moreover, given any of the above, the following are isomorphic:

(1) the category of bialgebras for the distributive law λ
(2) the category of coalgebras for the comonad D_T on T-Alg.
(3) the category of algebras for the monad T_D on D-Coalg

Now recall the corresponding situation for coalgebras (see [2] for the dual):

Proposition 4.4 *The following are equivalent:*

(1) a distributive law of a comonad D over a comonad D'
(2) a lifting of the comonad D to a comonad $D_{D'}$ on D'-Coalg.

Given these equivalent conditions, the composite of functors DD' possesses a canonical comonad structure. Further, the category of D, D'-bicoalgebras is isomorphic to $D_{D'}$-Coalg and to DD'-Coalg.

Combining Propsitions 4.1, 4.3, and 4.4, we have the following result:

Proposition 4.5 *To give a distributive law λ of the monad $S_{\mathcal{T}} = \mathcal{T} \times (-)$ over an arbitrary comonad D is equivalent to giving a distributive law of D over the comonad $(-)^{\mathcal{T}}$. Moreover, the category of λ-bialgebras is isomorphic to the category $D(-)^{\mathcal{T}}$-Coalg.*

Applying this result to Theorem 4.2 yields the characterisation we seek, as follows:

Corollary 4.6 *If $\mathcal{T}$ is partially closed, the formula*

$$\lambda_X(s, e)(t) \;\simeq\; (s \dot{-} t,\, e(t \dot{-} s)).$$

corresponds to a distributive law of the comonad E over the comonad $(-)^{\mathcal{T}}$. Moreover, $E(-)^{\mathcal{T}}$-Coalg is isomorphic to the category of pairs

$$(* : X \times \mathcal{T} \rightharpoonup X,\; (\,.\,) : \mathcal{T} \times X \to X)$$

consisting of a timed transition system together with a delay operator.

5 Heterogeneous transition systems and product comonads

In this section, we consider the combination of timed behaviour with behaviour relative to actions as studied extensively in the coalgebra literature [6]. This allows us, in the succeeding section, to study operational semantics for the combination. For both generality and elegance, we make our theoretical analysis in terms of an arbitrary pair of comonads, sometimes specializing to the case in which one or both are cofree on endofunctors. As a leading example, for simplicity of exposition, we restrict our attention to timed transition systems as in Section 2 rather than the combination of timed transition systems with delay operators of Sections 3 and 4, but the generalities apply equally.

Definition 5.1 [11, Definition 7.1] Let A be a finite set of actions, let $\mathcal{T}$ be a time domain, and let P be a set. An *heterogeneous transition system* $(P, \to, \rightsquigarrow)$ over $\mathcal{T}$ and A consists of

- an image-finite labelled transition system $(P, \{\xrightarrow{a}\}_{a \in A})$ and
- a timed transition system $(P, \rightsquigarrow)$ over $\mathcal{T}$

We saw in Section 2 that a timed transition system amounts to an $E_{\mathcal{T}}$-coalgebra for the evolution comonad $E_{\mathcal{T}}$. We have long known that an image-finite labelled transition system is given by a B-coalgebra for an endofunctor B, and, moreover, we may regard it as a D-coalgebra for the cofree comonad

D on B, (which exists for all other leading examples of behaviour functors too [6]). So an heterogeneous transition system amounts to a set together with a pair of coalgebra structures for comonads D and D', the former given by the cofree comonad on an endofunctor B. So, given comonads D and D', we seek to exhibit an heterogeneous transition system as a coalgebra for a comonad derived from D and D'. In fact, if it exists, the combined comonad must be the product $D \times D'$ of comonads (see Theorem 5.7 below). Such product comonads are somewhat subtle. In general they need not exist, and when they do they are typically not given pointwise. (Note that, for a category C with finite products, there need not be any naturally induced comonad structure on the pointwise product of two comonads D and D' on C. Also, the product comonad $D \times D'$ may exist even when C itself does not have finite products.) The next few results are working towards Theorem 5.7 and Corollary 5.8, which give general conditions under which the product comonad does exist.

Definition 5.2 Given comonads D and D' on a category C, define the category (D, D')-$Coalg$ to be the pullback in the large category of categories given by

$$\begin{array}{ccc} (D,D')-Coalg & \longrightarrow & D'-Coalg \\ \downarrow & & \downarrow{\scriptstyle U_{D'}} \\ D-Coalg & \xrightarrow[U_D]{} & C \end{array}$$

where U_D and $U_{D'}$ are the forgetful functors.

Proposition 5.3 *If the forgetful functor $U : (D, D')$-$Coalg \longrightarrow C$ has a right adjoint G, then (D, D')-$Coalg$ is comonadic over C with comonad given by G.*

PROOF. By the dual of Beck's monadicity theorem [2], it suffices to prove that U reflects isomorphisms and that (D, D')-$Coalg$ has and U preserves the equalisers of U-split equaliser pairs. Reflection of isomorphisms is trivial: a map in (D, D')-$Coalg$ is simply a map in C that preserves both coalgebra structures, and if that map in C is an isomorphism, its inverse must preserve both coalgebra structures. And for the second condition, any U-split equaliser pair is sent to a U_D-split equaliser pair in D-$Coalg$ and a $U_{D'}$-split equaliser pair in D'-$Coalg$. So the split equaliser in C must lift, by the converse (easy) part of Beck's theorem to an equaliser in both D-$Coalg$ and D'-$Coalg$, and so the equalising map in C is a map in (D, D')-$Coalg$ and satisfies the equalising property there. The functor U preserves it by construction. □

It is not easy to give a direct proof of the existence of a right adjoint to the forgetful functor $U : (D, D')\text{-}Coalg \longrightarrow C$, thus yielding comonadicity by the proposition, under general conditions. But an indirect route is readily available to us via a subtle use of results about accessible categories [16], cf [5].

Theorem 5.4 *If C is a locally presentable category and D and D' are accessible, the category (D, D')-$Coalg$ is locally presentable and the forgetful functor to C has a right adjoint.*

PROOF. First observe that coalgebra structure transports along isomorphism, i.e., given a D-coalgebra (X, d) and an isomorphism $f : X \longrightarrow X'$ in C, it follows that X' possesses a (unique) D-structure making f an isomorphism in D-$Coalg$. It follows that the category (D, D')-$Coalg$ is equivalent to the following category: an object consists of a D-coalgebra (X, d), a D'-coalgebra (X', d'), and an isomorphism in C between X and X'. This latter category is an iso-comma object

$$\begin{array}{ccc} P & \longrightarrow & D' - Coalg \\ \downarrow & \cong & \downarrow U_{D'} \\ D - Coalg & \xrightarrow[U_D]{} & C \end{array}$$

in the large category of categories. But the large category of accessible categories is closed under taking the category of coalgebras for an accessible comonad (see [5]), and under iso-comma objects [16], and under equivalence of categories. So (D, D')-$Coalg$ is an accessible category. Moreover, since D and D' are both accessible, D-$Coalg$ and D'-$Coalg$ are cocomplete and U_D and $U_{D'}$ preserve colimits. So (D, D')-$Coalg$ is also cocomplete and the forgetful functor to C preserves colimits. Thus (D, D')-$Coalg$ is a locally presentable category and the forgetful functor to C preserves colimits; so the latter has a right adjoint. $\Box$

Thus, for all examples of primary interest to us, e.g. for $C = \mathbf{Set}$ and D and D' being any of our leading examples, we do have a comonad. More analysis of the significance of accessibility and the fact that it includes all examples of substantial interest to us appears in [5]. Unusually, but fortunately, the fact that we know we have a comonad allows us to characterise it as the product of D and D'.

Assume we have an arbitrary category C with small copowers. (This assumption specialises the $\mathcal{V}$-tensors required in the $\mathcal{V}$-enriched context of [9].) Given an object X of C, consider the functor $\coprod_{C(X,-)} X : C \longrightarrow C$. It sends an

object Y to the coproduct of $C(X,Y)$ copies of X. For an arbitrary endofunctor $H : C \longrightarrow C$, it follows from the Yoneda lemma that to give a natural transformation

$$\chi : \coprod_{C(X,-)} X \Rightarrow H$$

is equivalent to giving a map $x : X \longrightarrow HX$. One can readily prove that the functor $\coprod_{C(X,-)} X$ possesses a natural comonad structure, and one has the following equivalence, as used extensively for instance in the dual setting in [9].

Proposition 5.5 *For a comonad D on C, to give a map of comonads*

$$\chi : \coprod_{C(X,-)} X \Rightarrow D$$

is equivalent to giving a D-coalgebra structure (X, x) on the object X.

Using that proposition, one can immediately prove the following.

Proposition 5.6 *For comonads D and D' on C, if the product of comonads $D \times D'$ exists, the category of coalgebras $(D \times D')$-$Coalg$ is canonically isomorphic to the pullback*

$$\begin{array}{ccc} (D,D') - Coalg & \longrightarrow & D - Coalg \\ \downarrow & & \downarrow U_{D'} \\ D - Coalg & \xrightarrow[U_D]{} & C \end{array}$$

PROOF. To give a $D \times D'$-coalgebra is equivalent to giving a map of comonads of the form

$$\chi : \coprod_{C(X,-)} X \Rightarrow D \times D'$$

but that, by definition of product, is equivalent to giving a pair of maps

$$\chi : \coprod_{C(X,-)} X \Rightarrow D \qquad\qquad \chi' : \coprod_{C(X,-)} X \Rightarrow D'$$

which in turn is equivalent to giving an object of (D, D')-$Coalg$. All these equivalences are natural, yielding the result. □

Comonads are characterised by their categories of coalgebras, and the canonical isomorphism of the proposition commutes with the underlying functors to C. So the proposition has a converse as follows.

Theorem 5.7 *If the forgetful functor from (D, D')-$Coalg$ to C is comonadic with comonad G, then the product of comonads $D \times D'$ exists and is given by G.*

PROOF. The canonical functor

$$(D, D') - Coalg \longrightarrow D - Coalg$$

commutes with the forgetful functors to C. So, if (D, D')-$Coalg$ is of the form G-$Coalg$, the functor must be of the form

$$\delta - Coalg : G - Coalg \longrightarrow D - Coalg$$

for a map of comonads $\delta : G \Rightarrow D$ (see [2] for the dual result). Thus we have projections δ and δ'. Now, given a comonad W, to give a comonad map $\omega : W \Rightarrow D$ is equivalent to giving a functor from W-$Coalg$ to D-$Coalg$ that commutes with the forgetful functors. Using the definition of (D, D')-$Coalg$ as a pullback, we obtain a unique functor from S-$Coalg$ to G-$Coalg$ that commutes with the forgetful functors and with δ-$Coalg$ and δ'-$Coalg$, and hence the desired unique map of comonads. □

Combining Proposition 5.3, Theorem 5.4, and Theorem 5.7, we can now deduce the result we seek.

Corollary 5.8 *If C is a locally presentable category and D and D' are accessible comonads on C, the product $D \times D'$ exists and is given by the right adjoint to the forgetful functor from (D, D')-$Coalg$ to C, exhibiting (D, D')-$Coalg$ as $(D \times D')$-$Coalg$.*

This corollary includes all examples that are likely to be of much interest to us. But it does not give us a construction of the product $D \times D'$ that we can readily calculate. However, in the cases of primary interest to us, one of the comonads, that given by the action behaviour, is the cofree comonad on an endofunctor. And in that case, the dual of a result for monads in [3] does give us a reasonable construction as follows.

Theorem 5.9 *For any category C and any endofunctor B and comonad D for which the cofree comonads B^∞ and $(BD)^\infty$ on B and BD exist, the product $B^\infty \times D$ also exists and is given by the functor $D(BD)^\infty$ with a canonical comonad structure.*

The dual of this theorem appears in [3], and this theorem directly appears in [11, Theorem 7.1]. We shall not include the detailed derivation here, although it does contain results of independent interest. Instead, we refer to the development of Chapter 7 of the first author's thesis [11]. Constructions

of cofree comonads on endofunctors abound in the coalgebraic literature, for instance in [23] but see also [8,18]. In particular, if C is locally presentable and B and D are accessible, the cofree comonads B^∞ and $(BD)^\infty$ exist [5], and so the theorem holds.

More specifically still, recall that in the cases of discrete and qualitative time (Examples 2.5 and 2.7), the comonad $E_{\mathcal{T}}$ is itself the cofree comonad on an endofunctor. So there is some interest in the situation in which both D and D' are cofree comonads on endofunctors B and B'. That is a particularly simple case, because then, the category B-$Coalg$ of coalgebras for the endofunctor B is isomorphic to the category D-$Coalg$ of coalgebras for the comonad D, and so, by a variant of the above analysis, we have the following result.

Corollary 5.10 *Given endofunctors B and B' on a category C with finite products such that the cofree comonads B^∞ and B'^∞ exist, the product of comonads $B^\infty \times B'^\infty$ exists and is given by $(B \times B')^\infty$, where $B \times B'$ is the pointwise product of endofunctors, providing the cofree comonad on $B \times B'$ exists. Moreover, whether or not the cofree comonad on $B \times B'$ exists, the category (B^∞, B'^∞)-$Coalg$ is equivalent to the category $(B \times B')$-$Coalg$ of coalgebras for the endofunctor $B \times B'$.*

6 Structural operational semantics for a combination of behaviours

In this section, we incorporate time into Turi and Plotkin's coalgebraic formulation of structural operational semantics [21] as expressed in terms of distributive laws [14,15,18]. They considered a category C with finite products, a "syntax" endofunctor Σ on C, and a "behaviour" endofunctor B on C, and they modelled a $GSOS$ rule by an abstract operational rule, which they defined to be a natural transformation $\Sigma(B \times Id) \Rightarrow BT$, where T is the free monad on Σ. It was shown in [14] and further explained and exploited in [18,15] that to give an abstract operational rule is equivalent to giving a distributive law of the monad T over the cofree copointed endofunctor on B, from which one can deduce a distributive law of T over D.

Here, extending the work of Section 5, we combine the operational semantics generated by two sorts of behaviour, i.e., start with endofunctors B and B' or more generally with comonads D and D' and try, using more primitive data, to induce a distributive law of T over the cofree copointed endofunctor on $B \times B'$ or more generally over the comonad $D \times D'$ if the latter exists. We shall not develop the example of time in detail in this section, as we have explained its role in detail in previous sections. We refer to the thesis [11] for explanation of exactly how operational semantics works in the case of time, with examples of combined rule formats.

For a simple first result, consider the following:

Theorem 6.1 *Given a monad T, comonads D and D', and distributive laws $\lambda : TD \Rightarrow DT$ and $\lambda' : T'D \Rightarrow DT'$, there is a canonical distributive law of T over $D \times D'$ if the product of comonads $D \times D'$ exists.*

PROOF. This follows from [19] together with Proposition 5.6. By the former, the two distributive laws give liftings of T to D-$Coalg$ and D'-$Coalg$ respectively. By the latter, these liftings yield a monad on $(D \times D')$-$Coalg$, as it is the pullback category P. So by the converse part of [19], we have the distributive law of T over $D \times D'$ that we seek. □

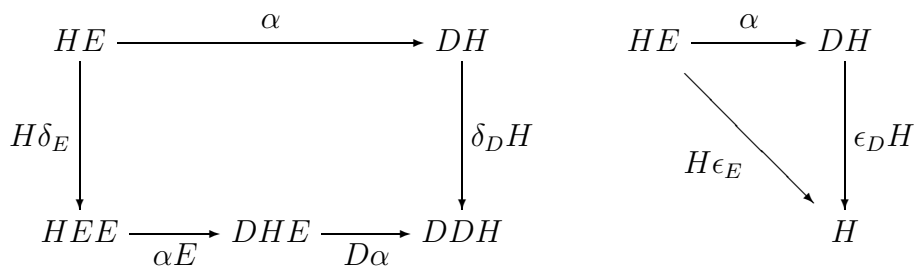
This result is less general than one would like because one does not always start with distributive laws $\lambda : TD \Rightarrow DT$ and $\lambda' : T'D \Rightarrow DT'$ or with anything that induces them [11]. The reason is that, in the leading examples, the time information and the action information typically interact with each other [11, Section 7.3.2]. So we consider the following question: given a monad T and comonads D and D', what are necessary and sufficient data that separate D and D' to some extent yet yield a lifting of the monad T to the category (D, D')-$Coalg$?

The following result was not explicitly stated in [19], but does follow from the analysis therein, which in turn was based on the characterisation of D-$Coalg$ as a limit in [20].

Proposition 6.2 *Given comonads $(D, \delta_D, \epsilon_D)$ and $(E, \delta_E, \epsilon_E)$ on a category C and a functor $H : C \longrightarrow C$, to give a lifting of H to a functor from E-$Coalg$ to D-$Coalg$ is equivalent to giving a natural transformation*

$$\alpha : HE \Rightarrow DH$$

subject to commutativity of the following two diagrams:

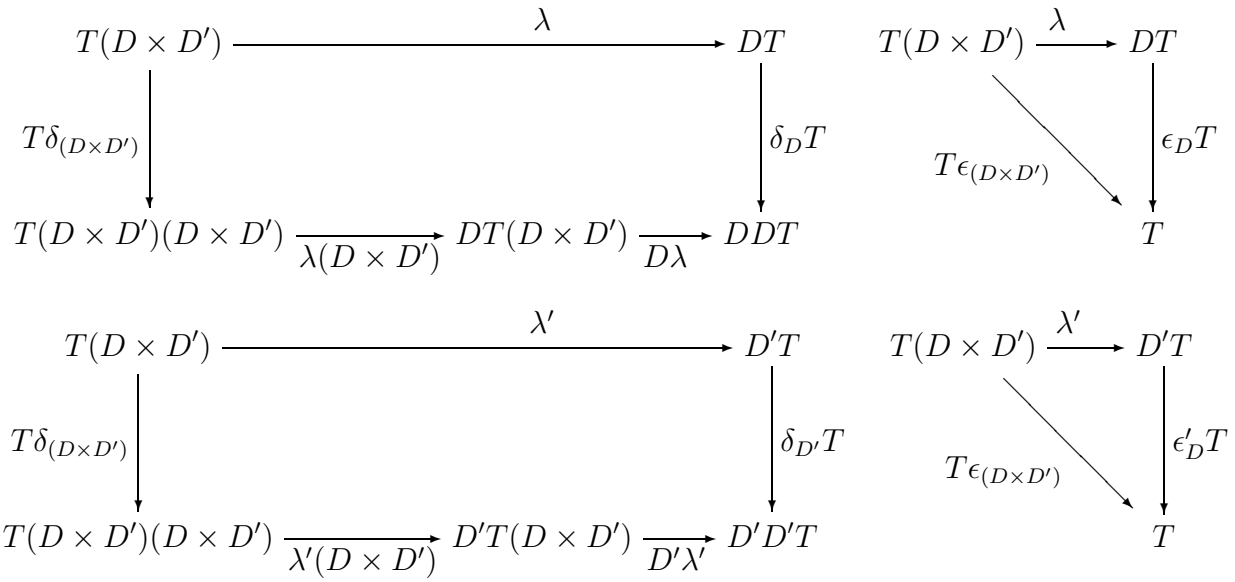


Using two copies of this proposition and our characterisation of $(D \times D')$-$Coalg$ as (D, D')-$Coalg$ in Proposition 5.6, we can readily deduce the following.

Proposition 6.3 *Given comonads D and D' on a category C such that the product comonad $D \times D'$ exists, and given a functor $T : C \longrightarrow C$, to give a lifting of T to an endofunctor on $(D \times D')$-Coalg is equivalent to giving natural transformations*

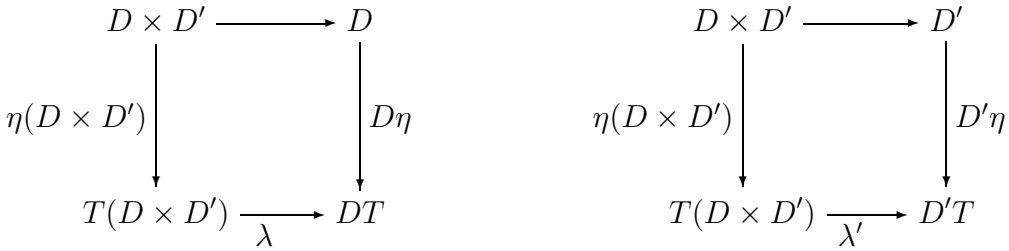
$$\lambda : T(D \times D') \Rightarrow DT \qquad \lambda' : T(D \times D') \Rightarrow D'T$$

subject to the following four axioms:



Now suppose one has not just an endofunctor T but a pointed endofunctor (T, η) that one wants to lift.

Proposition 6.4 *Given comonads D and D' on a category C such that the product $D \times D'$ exists, and given a pointed endofunctor (T, η) on C together with a lifting of the endofunctor T to $(D \times D')$-Coalg (or equivalently with the data of Proposition 6.3 subject to the axioms of the proposition), the unit η of T lifts if and only if the following two diagrams commute:*



PROOF. We already have the data for the unit, and the naturality condition is trivial. The only question is of finding necessary and sufficient conditions for each component of the natural transformation to be a map in (D, D')-$Coalg$; but we know that the maps in (D, D')-$Coalg$ are given by maps in C that

respect both coalgebra structures. That the necessary and sufficient condition is as stated appears in [19] (essentially gleaned from [20]), but it is also easy to check directly. □

Finally, given a monad (T, μ, η), we seek necessary and sufficient conditions for the multiplication to lift. This is the one point that is not so easy: the lifting of the functor part of the monad, as in Proposition 6.3, yields a distributive law $\tau : T(D \times D') \Rightarrow (D \times D')T$. And one needs to use that induced distributive law in the diagrams required to commute in order to make the multiplication lift. The conditions are easy to write if one is willing to use that distributive law, but it does make for potentially tricky calculation in verifying that examples satisfy the condition, as that distributive law is induced by more primitive data. Fortunately, in particular cases, commutativity of the diagrams is fairly routine to verify.

Proposition 6.5 *Given comonads D and D' on a category C such that the product $D \times D'$ exists, and given a monad (T, μ, η) on C together with a lifting of the pointed endofunctor (T, η) to $(D \times D')$-Coalg (or equivalently with the data of Propositions 6.3 and 6.4 subject to the axioms of the propositions), the multiplication μ of T lifts if and only if the following two diagrams commute:*

$$\begin{array}{ccccc} TT(D \times D') & \xrightarrow{T\tau} & T(D \times D')T & \xrightarrow{\lambda T} & DTT \\ \downarrow{\scriptstyle \mu(D \times D')} & & & & \downarrow{\scriptstyle D\mu} \\ T(D \times D') & & \xrightarrow[\lambda]{} & & DT \end{array}$$

$$\begin{array}{ccccc} TT(D \times D') & \xrightarrow{T\tau} & T(D \times D')T & \xrightarrow{\lambda' T} & D'TT \\ \downarrow{\scriptstyle \mu(D \times D')} & & & & \downarrow{\scriptstyle D'\mu} \\ T(D \times D') & & \xrightarrow[\lambda']{} & & D'T \end{array}$$

PROOF. The proof is similar to that for Proposition 6.4, using the universal property of the pullback (D, D')-*Coalg* and giving a necessary and sufficient condition for the components of a natural transformation to lift from C to each of D-*Coalg* and D'-*Coalg*. The above diagrams emerge fairly routinely, but one does need to think directly in terms of liftings, as the use of τ in both diagrams implies. □

The presence of τ in Proposition 6.5 but not in Proposition 6.4 means that the lifting of multiplication, as opposed to the lifting of the unit of a monad, depends upon both λ and λ' for each lifting, i.e., for lifting to each of D-$Coalg$ and D'-$Coalg$.

Evidently, Propositions 6.3, 6.4 and 6.5 can be combined to yield:

Corollary 6.6 *Given comonads D and D' on a category C such that the product $D \times D'$ exists, and given a monad (T, μ, η) on C, to give a lifting of the monad (T, μ, η) to $(D \times D')$-$Coalg$ is equivalent to giving natural transformations*

$$\lambda : T(D \times D') \Rightarrow DT \qquad \lambda' : T(D \times D') \Rightarrow D'T$$

subject to the commutativity of the eight diagrams in Propositions 6.3, 6.4 and 6.5.

There are special cases of Corollary 6.6. It is common to start with a distributive law of the form

$$TD \Rightarrow DT$$

for the action behaviour while only having a natural transformation of the form

$$T(D \times D') \Rightarrow D'T$$

for the time behaviour. That reduces the complexity of some of the diagrams a little, with the distributive law above often coming via the well-trodden paths of [21,14,18], but one still needs the second of our two diagrams in Proposition 6.5 involving τ.

In a slightly different direction, one can consider cases in which one or both of D and D' is the cofree comonad on an endofunctor, say D cofree on B. Then D-$Coalg$ is determined by the simpler universal property that characterises B-$Coalg$, and so one can avoid the coherence axioms required for the lifting of a functor. Life is also simpler because one can use our characterisation of the product of comonads. So, to lift a functor T to B-$Coalg$, one merely needs any natural transformation, subject to no axioms at all, of the form

$$\alpha : TD'(BD')^\infty \Rightarrow BT$$

and such can be readily constructed, for instance, from any natural transformation of the form

$$\beta : T(Id \times B)D' \Rightarrow BD'T$$

For applying D' to the counit $(BD')^\infty \Rightarrow Id$ yields a natural transformation $D'(BD')^\infty \Rightarrow D'$. And one has the canonical composite

$$D'(BD')^\infty \Rightarrow (BD')^\infty \Rightarrow BD'$$

and thus a natural transformation of the form $D'(BD')^\infty \Rightarrow (Id \times B)D'$, as products of endofunctors are given pointwise (assuming of course that C has products), and so applying T to this, and composing with β and with another embedded counit yields a natural transformation of the form α as above: $TD'(BD')^\infty \Rightarrow T(Id \times B)D' \Rightarrow BD'T \Rightarrow BT$.

Acknowledgements

The research in this paper extends that in the first author's PhD thesis, supervised by Gordon Plotkin and Daniele Turi, who are both thanked for their contributions to the work.

References

[1] Adamek, J., and J. Rosicky, "Locally Presentable and Accessible Categories," Cambridge University Press, 1994.

[2] Barr, M., and C. Wells, "Toposes, Triples, and Theories," Grundlehren der math. Wissenschaften **278**, 1985.

[3] Hyland, M., G.D. Plotkin, and A.J. Power, *Combining effects: sum and tensor*, Theoretical Computer Science, to appear.

[4] Jacobs, B., *Monocongruences and cofree coalgebras*, "Proceedings AMAST 95," Lecture Notes in Computer Science **936**, 1995, 245–260.

[5] Johnstone, P., A.J. Power, T. Tsujishita, H. Watanabe, and J. Worrell, *The structure of categories of coalgebras*, Theoretical Computer Science **260** (2001) 87–117.

[6] Jacobs, B., and J. Rutten. *A tutorial on (co)algebras and (co)induction*, Bulletin of the European Association of Theoretical Computer Science **62** (1997), 222–259.

[7] Jeffrey, A.S.A., S.A. Schneider, and F.W. Vaandrager, "A comparison of additivity axioms in timed transition systems," CWI Technical Report CS-R9366, 1993.

[8] Kelly, G.M., *A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on*, Bulletin Australian Mathematical Society **22** (1980), 1–83.

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

[10] Kick, M., *Bialgebraic modelling of timed processes*, "Proceedings of ICALP 2002," Lecture Notes in Computer Science **2380**, 2002, 525–536.

[11] Kick, M., "A Mathematical Model of Timed Processes" Ph.D. dissertation, University of Edinburgh, 2003.

[12] Kick, M., and A.J. Power, *Modularity of behaviours for mathematical operational semantics*, Electronic Notes in Theoretical Computer Science (2004).

[13] Lawvere, F.W., *Metric spaces, generalized logic, and closed categories*, Rend. del Sem. Mat. e Fis. di Milano **43** (1973), 135–166.

[14] Lenisa, M., A.J. Power, and H. Watanabe, *Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads*, Electronic Notes in Theoretical Computer Science **33** (2000).

[15] Lenisa, M., A.J. Power, and H. Watanabe, *Category theory for operational semantics*, Theoretical Computer Science (to appear).

[16] Makkai, M., and R. Pare, "Accessible categories: the foundations of categorical model theory," Contemporary Mathematics **104**, 1989.

[17] Nicollin, X., and J. Sifakis, *An overview and synthesis of timed process algebras*, "Proceedings CAV 1991", Lecture Notes in Computer Science **575**, 1991, 376–398.

[18] Power, A.J., *Towards a theory of mathematical operational semantics*, Electronic Notes in Theoretical Computer Science **82.1** (2003).

[19] Power, A.J., and H. Watanabe, *Combining a monad and a comonad*, Theoretical Computer Science **280** (2002), 137–162.

[20] Street, R., *The formal theory of monads*, Journal of Pure and Applied Algebra **2** (1972), 149–168.

[21] Turi, D., and G.D. Plotkin, *Towards a mathematical operational semantics*, "Proceedings of 12th Symposium on Logic in Computer Science," IEEE Computer Science Press, 1997, 280–291.

[22] Wang, Y., *Real-time behaviour of asynchronous agents*, "Proceedings of CONCUR 1990", Lecture Notes in Computer Science **458**, 1990, 502–520.

[23] Worell, James, *Terminal sequences for accessible endofunctors*, Electronic Notes in Theoretical Computer Science **19** (1999).