Electronic Notes in Theoretical Computer Science 68 No. 1 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume68.html 20 pages

Marco Kick [1]

LFCS, Division of Informatics, University of Edinburgh
Edinburgh EH9 3JZ, Scotland, UK
Email: mk@dcs.ed.ac.uk

Rule Formats for Timed Processes

Abstract

Building on previous work [15,8], this paper describes two syntactic ways of defining 'well-behaved' operational semantics for timed processes. In both cases, the semantic rules are derived from abstract operational rules for *behaviour comonads* and thus ensure congruence results. The first of them, a light-weight attempt using *schematic* rules, is shown to be *sound*, i.e., to indeed induce abstract rules as introduced in [8]. Then a second format, based on a new and very general kind of abstract rules, *comonadic SOS (CSOS)*, is presented which uses *meta rules* and is also *complete*, i.e., it characterises all possible CSOS rules for timed processes.

1 Introduction

In their paper [15], Turi and Plotkin presented a mathematical theory for Plotkin's *structural operational semantics (SOS)* [12], using the categorical notions of *distributive law* of a monad over a comonad [13], and *bialgebras* [15] of such a law. To model *abstract operational rules*, they considered natural transformations of type

$$\Sigma(Id \times B) \Rightarrow BT \tag{1}$$

for functorial notions Σ and B of *signature* and (one-step) *behaviour*, respectively, and T the monad freely generated by Σ (corresponding to terms over the signature Σ). It was shown that abstract rules as in (1) induce a distributive law of T over the cofree comonad on B (for the definition see, e.g., [14]). Using the bialgebras of this law, they were able to obtain an abstract congruence result for coalgebraic B-bisimulation [1] (under the technical assumption that B preserves weak pullbacks).

For a particular choice of behaviour functor, they also showed that the resulting abstract rules (1) correspond precisely to concrete rules in the well-studied *GSOS* [4] format. Thus their approach offers a conceptual explanation

[1] This work is supported by EPSRC grant GR/M56333

why bisimulation is a congruence for GSOS languages. This illustrates that, under auspicious circumstances, it is possible to derive syntactic formats from abstract rules by careful analysis of the constraints expressed in (1), stemming from the chosen behaviour and the naturality of the rules. Very recently, this has led to the discovery of a GSOS-style format for probabilistic transition systems [3].

With an analogous format as the ultimate goal, [8] showed how to adapt the approach of [15] to accommodate *timed processes* and the time rules of timed process calculi, with *Temporal CCS* (TeCCS) [10] as the principal example. This was achieved by first distilling from the literature abstract accounts of time and timed processes as *time domains* (special kinds of monoids, e.g., $\mathbb{N}$ or $\mathbb{R}_{\geq 0}$ with addition) and *timed transition systems* (TTSs, labelled transition systems with suitably restricted transition relations as to account for specific properties of time passing), respectively. One important result was that TTSs are the same as the (Eilenberg-Moore) coalgebras for a novel *evolution comonad* E. More abstractly, the desire to accommodate coalgebras for the evolution comonad within the bialgebraic framework of [15] necessitated finding abstract rules for general *behaviour comonads*, corresponding to 'global' behaviour, or 'big-step' semantics, as opposed to 'local' behaviour, or 'small-step' semantics, described by behaviour functors.

The solution presented in [8] employed abstract rules of type

$$\Sigma E \Rightarrow E(Id + \Sigma) \tag{2}$$

respecting the structure of the comonad E, i.e., validating two diagrams relating the rules (2) to the counit and the comultiplication of E, respectively. These requirements became necessary since E is not cofreely generated, giving a *complete* account of *all* the time transitions of a process *at once*, unlike the cofree case where atomic steps are iterated without producing additional constraints. On the level of concrete rules, these new constraints correspond to *global* conditions on *sets* of rules, in contrast to the GSOS conditions operating on a *local*, i.e., per-rule, basis. Abstract rules (2) satisfying these conditions then indeed induced a distributive law of the free monad on Σ over E.

As an application, it was shown that the (time) rules for TeCCS can be expressed in this way, and so a congruence result for E-bisimulation (given by *time bisimulation*, a very natural notion of equivalence for TTSs) was obtained. Furthermore, it was shown in [8] that for the special case of *discrete* time (with time domain $\mathbb{N}$), the evolution comonad is actually cofreely generated from a functor. Hence, applying the machinery of [15], a syntactic format for at least this case was obtained, leaving the derivation of a format for an arbitrary time domain as an open problem. The objective of the present paper is to close this gap by presenting two different ways of syntactically specifying well-behaved operational semantics for timed processes in the general case. In doing so, it is assumed (as in [8]) that there is no interference between time rules and action rules, and so again only the time rules are discussed since action rules

are commonly just GSOS rules already fitting in the framework of [15] and thus need not be considered here.

The paper is organised as follows. After §2, which contains a brief exposition of the necessary background from [8], §3 introduces *schematic rule shapes* with *time variables* allowing to uniformly derive time transitions by instantiating the variables with concrete time values (as done, e.g., in the operational semantics of TeCCS), and *admissible operators* consisting of specific combinations of such shapes. These constitute a *sound* way of describing the behaviour of timed processes, i.e., admissible operators do indeed induce abstract rules of type (2) which additionally respect the structure of E. Therefore, time bisimulation is automatically established as a congruence for such operators, which are powerful enough to encompass the time rules of TeCCS. Hence, this yields a new proof of the congruence result obtained in [8] (and also in the original paper [10]). However, they are *not complete* since there are easy examples of abstract rules as in (2) respecting the structure of E, yet not induced by admissible operators. Even so, the approach provides an easy-to-use, systematic way of describing well-behaved rules for timed processes, expressive enough to include most interesting operators from the literature.

Next, §4 presents very powerful abstract operational rules for an arbitrary behaviour comonad D, based on natural transformations of type

$$\Sigma D \Rightarrow DT \tag{3}$$

This *comonadic SOS (CSOS)* combines elements of (1) and (2) but is strictly more expressive than either. In order to guarantee that CSOS rules induce a distributive law of T over the D as before, the rules again have to respect the structure of D, which, due to the increased expressivity, results in more complex global conditions on the abstract rules (3). The generality of CSOS is then demonstrated by showing that CSOS rules respecting the structure of D are already *equivalent* to giving the full distributive law $TD \Rightarrow DT$, the most general kind of abstract rules for a freely generated monad T, thereby extending results from [13,9].

Finally, §5 presents a sound and complete format for timed processes derived from CSOS rules instantiated with the evolution comonad, establishing a one-to-one correspondence between concrete and abstract rules at the price of using infinite sets of infinitary *meta rules*. This is essentially due to the evolution comonad describing timed processes as evolutions with, in general, infinite domain. To give a complete characterisation of such processes (fitting with the 'global behaviour' interpretation of behaviour comonads), one needs infinitely many premises, and the infinite number of possible domains requires infinitely many rules. The admissible operators from §3 are contained in the more general format, thus providing a sufficiently concrete description for most, if not all, important cases.

2 Bialgebraic Semantics for Timed Processes

This section briefly reviews some of the definitions and results from [8].

Definition 2.1 (i) A *time domain* is a commutative monoid $\mathcal{T} = (\mathcal{T}, +, 0)$ which, for all $s, t, u \in \mathcal{T}$, satisfies the *cancellation rule*

$$s + t = s + u \Rightarrow t = u$$

and whose *induced preorder* $\leq$, defined as

$$t \leq u \Leftrightarrow (\exists s).\ t + s = u$$

is a linear order.

(ii) A *timed transition system (TTS)* is a labelled transition system (LTS) $(P, \mathcal{T}, \rightsquigarrow)$ where P is a set of *processes*, $\mathcal{T}$ is a time domain, and the *transition relation* $\rightsquigarrow \subseteq P \times \mathcal{T} \times P$ satisfies the following axioms, where $p \overset{t}{\rightsquigarrow} p'$ denotes $(p, t, p') \in \rightsquigarrow$:

(Determinacy)	$p \overset{t}{\rightsquigarrow} p' \wedge p \overset{t}{\rightsquigarrow} p'' \Rightarrow p' = p''$
(Zero-Delay)	$p \overset{0}{\rightsquigarrow} p$
(Continuity)	$p \overset{t+u}{\rightsquigarrow} p' \Leftrightarrow (\exists p'').\ p \overset{t}{\rightsquigarrow} p'' \overset{u}{\rightsquigarrow} p'$

(iii) Given TTSs $(P_i, \mathcal{T}, \rightsquigarrow_i)$, $i \in \{1, 2\}$, a relation $R \subseteq P_1 \times P_2$ is a *(strong) time bisimulation (over $\mathcal{T}$)* if $(p_1, p_2) \in R$ implies for all $t \in \mathcal{T}$ that

$$p_1 \overset{t}{\rightsquigarrow}_1 p_1' \Rightarrow (\exists p_2').\ p_2 \overset{t}{\rightsquigarrow}_2 p_2' \wedge (p_1', p_2') \in R$$
$$p_2 \overset{t}{\rightsquigarrow}_2 p_2' \Rightarrow (\exists p_1').\ p_1 \overset{t}{\rightsquigarrow}_1 p_1' \wedge (p_1', p_2') \in R$$

To obtain a coalgebraic description of timed processes represented by TTSs, the following notion of *evolution* is introduced:

Definition 2.2 Let $\mathcal{T}$ be a time domain and X a set. A *$\mathcal{T}$-evolution on X* is a partial function $e : \mathcal{T} \rightharpoonup X$ satisfying the two axioms, for $t, u \in \mathcal{T}$:

$$e(0)\downarrow \tag{4}$$
$$e(t + u)\downarrow \Rightarrow\ e(t)\downarrow \tag{5}$$

where $(\ldots)\downarrow$ denotes that the partial expression $(\ldots)$ is defined (and, dually, $(\ldots)\uparrow$ denotes undefinedness of the expression). The set of all $\mathcal{T}$-evolutions on X is written as $EX = E_{\mathcal{T}}X$. Given a function $f : X \to Y$, define the map $Ef : EX \to EY$ by $e \mapsto f \circ e$.

This last definition makes $E = E_{\mathcal{T}}$ into an endofunctor on **Set**, the category of sets and total functions, yet more is true:

Proposition 2.3 *The functor E is a comonad on* **Set**, *with counit ε and comultiplication δ given by*

$$\varepsilon_X : EX \to X,\ e \mapsto e(0)$$

$$\delta_X : EX \to E^2X,\ e \mapsto \lambda t.\begin{cases} e+t \stackrel{\mathsf{df}}{=} \left(\lambda u.\begin{cases} e(t+u) & \text{if } e(t+u)\downarrow \\ \text{undef} & \text{if } e(t+u)\uparrow \end{cases}\right) & \text{if } e(t)\downarrow \\ \text{undef} & \text{if } e(t)\uparrow \end{cases}$$

and the (Eilenberg-Moore) coalgebras for E are precisely TTSs. Moreover, coalgebraic E-bisimulation [1] (extended to the case of coalgebras for comonads) is time bisimulation, and E preserves pullbacks.

Abstract rules for the behaviour comonad E are then obtained as follows:

Theorem 2.4 *Let Σ be an endofunctor on* **Set** *such that the free monad T on Σ exists. Furthermore, suppose $\rho : \Sigma E \Rightarrow E(Id+\Sigma)$ is a natural transformation respecting the structure of E, i.e., satisfying the ε- and δ-diagrams*

(6)
$$\begin{array}{ccc} \Sigma E & \stackrel{\rho}{\Longrightarrow} & E(Id+\Sigma) \\ {\scriptstyle \Sigma\varepsilon}\Downarrow & & \Downarrow{\scriptstyle \varepsilon_{Id+\Sigma}} \\ \Sigma & \stackrel{}{\underset{inr}{\Longrightarrow}} & Id+\Sigma \end{array} \qquad \begin{array}{ccccc} \Sigma E & & \stackrel{\rho}{\Longrightarrow} & & E(Id+\Sigma) \\ {\scriptstyle \Sigma\delta}\Downarrow & & & & \Downarrow{\scriptstyle \delta_{Id+\Sigma}} \\ \Sigma E^2 & \stackrel{}{\underset{\rho_E}{\Longrightarrow}} & E(E+\Sigma E) & \stackrel{}{\underset{f_\rho}{\Longrightarrow}} & E^2(Id+\Sigma) \end{array}$$

for $f_\rho \stackrel{\mathsf{df}}{=} E[Einl, \rho]$. Then ρ induces a distributive law $\ell : TE \Rightarrow ET$ of the monad T over the comonad E.

Theorem 2.5 *The time rules of* TeCCS *can be described by a natural transformation ρ as in Thm. 2.4 which respects the structure of E.*

Since E in particular preserves weak pullbacks, the theory of ℓ-bialgebras from [15] applies to the TeCCS-case, allowing to automatically obtain the following well-established result:

Corollary 2.6 *Time bisimulation is a congruence for* TeCCS

3 Schematic Rules for Timed Processes

This section presents a simple way of specifying a well-behaved operational semantics of timed processes over an arbitrary time domain, using *schematic* rules: the rules only contain time transitions which are labelled by *time variables*, rather than concrete time values. Then, in order to derive concrete time transitions, the time variables in such a rule must be instantiated with the actual values, subject to applicability of the rule.

To this end, certain schematic *rule shapes* for defining time transitions are introduced, and only certain combinations of these are allowed as *admissible operators* for describing concrete operators. Hence instead of a 'format',

this approach really just yields a collection of 'operator blueprints'. In order to describe timed processes, admissible operators have to include *time-parameterised* operators, i.e., operators which have time(s) as parameters, in addition to the usual parameters for processes. For simplicity, the set of admissible operators only considers the case of at most one time parameter.

In deriving the schematic rules, the time rules of TeCCS served as the guiding example, in particular time prefixing $(t).p$ of TeCCS, for a time $t \neq 0$ and a process p, is the prototype of a time-parameterised operator. Consequently, the admissible operators encompass the time rules of TeCCS but also additional rules from [11]. Soundness of the admissible operators is established by showing that admissible operators indeed induce natural transformations as in Thm. 2.4, while the failure of completeness is then demonstrated by presenting a simple example of abstract rules not expressible by an admissible operator. Finally, a way to make the schematic shapes a bit more permissive is discussed by considering them relative to a time domain.

In the remainder of this section, let $\mathcal{V}$ be a countable set of variables, and let $\mathcal{T}$ be an arbitrary time domain, writing $\mathcal{T}_> \stackrel{\mathsf{df}}{=} \mathcal{T} \setminus \{0\}$. Note that the time variables in the following rule shapes are only allowed to range over $\mathcal{T}_>$. It is therefore impossible to derive $\overset{0}{\rightsquigarrow}$-transitions. The reason for this restriction is that the ε-diagram in (6) already determines the targets such transitions (see the proof of Prop. 3.4). Note that also potential time parameters cannot have value 0. Furthermore, all variables occurring in a rule must be distinct, hence the restriction that each rule shape should be a GSOS rule:

Definition 3.1 Let Σ be a signature and $\sigma \in \Sigma$ be a function symbol; write $\boldsymbol{x} = (x_1, \ldots, x_n)$ for distinct $x_i \in \mathcal{V}$ and the appropriate arity $n \in \mathbb{N}$, s, t time variables ranging over $\mathcal{T}_>$, and $I \subseteq \{1, \ldots, n\}$, $1 \leq j \leq n$ such that $j \notin I$. Then the possible *rule shapes* are GSOS rules of the following kinds:

(i) Standard operators defined by rules of the shapes

(A) $$\frac{-}{\sigma(\boldsymbol{x}) \overset{t}{\rightsquigarrow} \sigma(\boldsymbol{x})}$$

(B) $$\frac{x_1 \overset{t}{\rightsquigarrow} x_1' \ \cdots \ x_n \overset{t}{\rightsquigarrow} x_n'}{\sigma(\boldsymbol{x}) \overset{t}{\rightsquigarrow} \sigma(\boldsymbol{x}')}$$

(C_j) $$\frac{x_j \overset{t}{\rightsquigarrow} x_j', (\forall 1 \leq k \leq n).\ k \neq j \Rightarrow x_k \overset{t}{\not\rightsquigarrow}}{\sigma(\boldsymbol{x}) \overset{t}{\rightsquigarrow} x_j'}$$

(ii) Time-parameterised operators defined by rules of the shapes

(tA_I) $$\frac{\{x_i \overset{t}{\rightsquigarrow} x_i' \mid i \in I\},\ (\forall 1 \leq k \leq n).\ y_k = \begin{cases} x_k' & \text{if } k \in I \\ x_k & \text{if } k \notin I \end{cases}}{\sigma(\boldsymbol{x}, t+s) \overset{t}{\rightsquigarrow} \sigma(\boldsymbol{y}, s)}$$

$$(\mathrm{tB}_{I,j}) \qquad \frac{\{x_i \overset{t}{\rightsquigarrow} x'_i \mid i \in I\}}{\sigma(\boldsymbol{x}, t) \overset{t}{\rightsquigarrow} x_j}$$

$$(\mathrm{tC}_{I,j}) \qquad \frac{\{x_i \overset{t}{\rightsquigarrow} x'_i \mid i \in I\},\ x_j \overset{s}{\rightsquigarrow} x'_j}{\sigma(\boldsymbol{x}, t) \overset{t+s}{\rightsquigarrow} x'_j}$$

Note that for a constant c, i.e., a function symbol c of arity 0, the two shapes (A) and (B) become equal, and there can be no rule of shape (C_j).

Definition 3.2 Let Σ be a signature and let $\sigma \in \Sigma$ be a function symbol with arity $n \in \mathbb{N}$. Then, in addition to the case of no rules at all, the *admissible operators* are given as follows. For standard operators:

(i) for arity $n = 1$
- one rule of shape (A), or
- one rule of shape (B), or
- one rule of shape (C_j) for $j = 1$

(ii) for arity $n \neq 1$
- one rule of shape (A), or
- one rule of shape (B), or
- one rule of shape (B), and for each $1 \leq j \leq n$ one rule of shape (C_j)

For time-parameterised operators of arbitrary arity, the following operators are admissible:

- one rule of shape (tA_I) for some $I \subseteq \{1, \ldots, n\}$, or
- one rule for each of the shapes (tA_I), ($\mathrm{tB}_{I,j}$), and ($\mathrm{tC}_{I,j}$), with matching $I \subseteq \{1, \ldots, n\}$ and $1 \leq j \leq n$ such that $j \notin I$

Again for the case of a constant c, note that the only non-trivial (i.e., consisting of at least one rule shape) admissible operator is given by the case of one rule of shape (A), or equivalently (B), cf. the previous remark.

Example 3.3 All the operators of TeCCS can be modelled using the above admissible operators, e.g.:

(i) Nil process 0 and action prefix $\alpha.\_$: no rules;

(ii) Delay prefix $\delta.\_$: the unary case of one rule of shape (A);

(iii) Strong choice $+$ and parallel composition $|$: the binary case of one rule of shape (B)

(iv) Weak choice $\oplus$: the binary case of one rule of shape (B) and two rules of shape (C_j), one each for $j = 1$ and $j = 2$;

(v) Time-prefixing $(t).\_$ for $t \in \mathcal{T}_>$: the unary case of one rule each of the shapes (tA_I), ($\mathrm{tB}_{I,j}$), and ($\mathrm{tC}_{I,j}$) for $I = \emptyset$ and $j = 1$.

The main result of this section is showing that the admissible operators of Def. 3.2 provide a sound operational semantics for timed processes:

Proposition 3.4 *Let Σ be a signature and $\sigma \in \Sigma$ an n-ary function symbol. If the time rules for σ can be described by of the above admissible operators they induce a map $[\![\sigma]\!] : (E\mathcal{V})^n \to E(\mathcal{V} + \Sigma\mathcal{V})$ natural in $\mathcal{V}$ that respects the structure of E.*

Proof (Sketch) *The proof is exemplified for the case of* TeCCS*'s strong choice $+$. As stated in Ex. 3.3, its semantics is defined by one binary rule of shape* (B)*, translating to*

$$[\![e_1 + e_2]\!] \;=\; \lambda t \in \mathcal{T}_{>}.\; \begin{cases} e_1(t) + e_2(t) & \text{if } e_1(t)\!\downarrow \wedge\, e_2(t)\!\downarrow \\ \text{undef} & \text{if } e_1(t)\!\uparrow \vee\, e_2(t)\!\uparrow \end{cases}$$

for two evolutions $e_1, e_2 \in E\mathcal{V}$. Note again that this map is not defined for $t = 0$ since the rule shapes cannot be instantiated with that value. However, the ε-diagram of (6) *expresses the validity of the equation*

$$[\![e_1 + e_2]\!](0) = e_1(0) + e_2(0)$$

Thus, simply take this as the definition *of $[\![e_1 + e_2]\!](0)$, automatically making the ε-diagram commute. Put together, this yields the well-defined map*

$$[\![+]\!] : (E\mathcal{V})^2 \to E(\mathcal{V} + \Sigma\mathcal{V}) \;,$$
$$[\![e_1 + e_2]\!] \;=\; \lambda t.\; \begin{cases} e_1(t) + e_2(t) & \text{if } e_1(t)\!\downarrow \wedge\, e_2(t)\!\downarrow \\ \text{undef} & \text{if } e_1(t)\!\uparrow \vee\, e_2(t)\!\uparrow \end{cases}$$

Also the δ-diagram of (6) *commutes for $[\![+]\!]$, as shown by the following calculations. First, the result of the map $\delta_{X+\Sigma X} \circ [\![+]\!]$, the right-down path around the diagram, is computed as follows:*

$$[\![e_1 + e_2]\!] \;=\; \lambda t.\; \begin{cases} e_1(t) + e_2(t) & \text{if } e_1(t)\!\downarrow \wedge\, e_2(t)\!\downarrow \\ \text{undef} & \text{if } e_1(t)\!\uparrow \vee\, e_2(t)\!\uparrow \end{cases} \overset{\delta_{X+\Sigma X}}{\longmapsto}$$
$$\lambda t.\; \begin{cases} \lambda u.\; \begin{cases} e_1(t+u) + e_2(t+u) & \text{if } e_1(t+u)\!\downarrow \wedge e_2(t+u)\!\downarrow \\ \text{undef} & \text{if } e_1(t+u)\!\uparrow \vee e_2(t+u)\!\uparrow \end{cases} & \text{if } e_1(t)\!\downarrow \wedge e_2(t)\!\downarrow \\ \text{undef} & \text{if } e_1(t)\!\uparrow \vee e_2(t)\!\uparrow \end{cases}$$

The map $E[Einl, [\![+]\!]] \circ [\![+]\!]_{EX} \circ (\delta + \delta)$, corresponding to the other path around the diagram, results in the following map, using the fact that $e(t)\!\downarrow \Leftrightarrow (\delta e)(t)\!\downarrow$:

$$[\![e_1 + e_2]\!] \overset{\delta+\delta}{\longmapsto} [\![\delta e_1 + \delta e_2]\!] = \;\lambda t.\; \begin{cases} \delta e_1(t) + \delta e_2(t) & \text{if } \delta e_1(t)\!\downarrow \wedge\, \delta e_2(t)\!\downarrow \\ \text{undef} & \text{if } \delta e_1(t)\!\uparrow \vee\, \delta e_2(t)\!\uparrow \end{cases}$$
$$= \;\lambda t.\; \begin{cases} (e_1 + t) + (e_2 + t) & \text{if } e_1(t)\!\downarrow \wedge\, e_2(t)\!\downarrow \\ \text{undef} & \text{if } e_1(t)\!\uparrow \vee\, e_2(t)\!\uparrow \end{cases} \overset{E[Einl,[\![+]\!]]}{\longmapsto}$$
$$\lambda t.\; \begin{cases} \lambda u.\; \begin{cases} (e_1 + t)(u) + (e_2 + t)(u) & \text{if } (e_1 + t)(u)\!\downarrow \wedge (e_2 + t)(u)\!\downarrow \\ \text{undef} & \text{if } (e_1 + t)(u)\!\uparrow \vee (e_2 + t)(u)\!\uparrow \end{cases} & \text{if } e_1(t)\!\downarrow \wedge e_2(t)\!\downarrow \\ \text{undef} & \text{if } e_1(t)\!\uparrow \vee e_2(t)\!\uparrow \end{cases}$$

$$= \lambda t.\ \begin{cases} \lambda u.\ \begin{cases} e_1(t+u) + e_2(t+u) & \text{if } e_1(t+u)\downarrow \wedge e_2(t+u)\downarrow \\ \text{undef} & \text{if } e_1(t+u)\uparrow \vee e_2(t+u)\uparrow \end{cases} & \text{if } e_1(t)\downarrow \wedge e_2(t)\downarrow \\ \text{undef} & \text{if } e_1(t)\uparrow \vee e_2(t)\uparrow \end{cases}$$

Hence the two maps are equal, and the diagram commutes. Finally, naturality follows from the fact that all the rule shapes obey the GSOS variable conditions, cf. the considerations in the proof of [15, Thm. 1.1]. □

Note that the intuitive meaning of the δ-diagram is illustrated in this proof: applying the rules once to derive a $\overset{t+u}{\rightsquigarrow}$-transition (expressed in the right-down path in the diagram) leads to the same process as applying the rules twice, first deriving a $\overset{t}{\rightsquigarrow}$-transition to an intermediate state, and then deriving a $\overset{u}{\rightsquigarrow}$-transition from there (the other path). In other words, the diagram represents a rule-, or derivation-based version of time continuity.

Soundness now follows by combining the maps $[\![\sigma]\!]$ for all operators $\sigma \in \Sigma$:

Theorem 3.5 *If the time rules of a language only use the above admissible operators they induce a natural transformation $\Sigma E \Rightarrow E(Id + \Sigma)$ respecting the structure of E.*

In addition to the time rules of TeCCS there are other operators in the literature that can be described using admissible operators. For example, consider the *time-out* operator $p \overset{t}{\triangleright} q$ from [11], with time rules

$$\frac{p \overset{t'}{\rightsquigarrow} p',\ t' < t}{p \overset{t}{\triangleright} q \overset{t'}{\rightsquigarrow} p' \overset{t-t'}{\triangleright} q} \qquad \frac{p \overset{t}{\rightsquigarrow} p'}{p \overset{t}{\triangleright} q \overset{t}{\rightsquigarrow} q} \qquad \frac{p \overset{t}{\rightsquigarrow} p',\ q \overset{t'}{\rightsquigarrow} q'}{p \overset{t}{\triangleright} q \overset{t+t'}{\rightsquigarrow} q'}$$

Intuitively, $p \overset{t}{\triangleright} q$ behaves like p strictly before time t; then at time t the control switches to q, simply discarding p: if p waits too long, viz., does not perform its intended task within t units of time, it gets preempted by the 'time-out handler' q. However, note that p really must wait until the point of preemption for the time-out to become effective: if p, for some reason, cannot idle at least for t units of time, q never gets activated.

The fact that $t' < t$ implies there is a unique $t'' \in \mathcal{T}_>$ such that $t + t'' = t$ (a property of time domains, cf. [8]), so the first rule can be rewritten as

$$\frac{p \overset{t'}{\rightsquigarrow} p'}{p \overset{t'+t''}{\triangleright} q \overset{t'}{\rightsquigarrow} p' \overset{t''}{\triangleright} q}$$

Then it fits the rule shapes as the binary case of shape (tA_I) with $I = \{1\}$; so do the other two rules, fitting shape ($\mathrm{tB}_{I,j}$) and shape ($\mathrm{tC}_{I,j}$), respectively, both with $I = \{1\}$ and $j = 2$. Since this is one of the allowed combinations for admissible operators in Def. 3.2, the time-out operator induces a map which respects the structure of E.

In contrast to that, the rules of the *start-delay* operator $\lfloor p \rfloor^t(q)$ from [11] do *not* fit any operator format:

$$\frac{p \overset{u}{\rightsquigarrow} p',\ u < t}{\lfloor p \rfloor^t q \overset{u}{\rightsquigarrow} \lfloor p' \rfloor^{t-u} q} \qquad \frac{(\forall s < t).\ p \overset{s}{\not\rightarrow},\ u < t}{\lfloor p \rfloor^t q \overset{u}{\rightsquigarrow} \lfloor p \rfloor^{t-u} q} \qquad \frac{p \overset{t}{\rightsquigarrow} p'}{\lfloor p \rfloor^t q \overset{t}{\rightsquigarrow} q} \qquad \frac{p \overset{t}{\rightsquigarrow} p',\ q \overset{u}{\rightsquigarrow} q'}{\lfloor p \rfloor^t q \overset{t+u}{\rightsquigarrow} q}$$

Intuitively, $\lfloor p \rfloor^t q$ is very similar to $p \overset{t}{\triangleright} q$, if p can idle for at least t units of time the two processes even behave in exactly the same way (as expressed in the first, third, and fourth rule). Yet if p cannot idle long enough there is a subtle difference: where the time-out $p \overset{t}{\triangleright} q$ simply cannot idle either, the start-delay $\lfloor p \rfloor^t q$, as stated in the second rule, allows further progress, provided p cannot perform *any* time transition whatsoever, and preserves p's action potential for longer than it would have been present originally.

Moreover, this is essentially the reason why the operator violates time continuity which is at the very heart of the TTS-based approach: for $\mathcal{T} = \mathbb{N}$ and any q, the rules allow the derivation $\lfloor (1).0 \rfloor^3 q \overset{1}{\rightsquigarrow} \lfloor 0 \rfloor^2 q \overset{1}{\rightsquigarrow} \lfloor 0 \rfloor^1 q$; yet $\lfloor (1).0 \rfloor^3 q \overset{2}{\not\rightarrow}$, since $(1).0 \overset{2}{\not\rightarrow}$ but $(1).0 \overset{1}{\rightsquigarrow} 0$, and so neither the first nor the second rule applies. Hence this particular operator is *not* compatible with the abstract model of timed processes used here, and its exclusion is actually *desirable* rather than problematic. On the level of abstract rules, the failure of time continuity is mirrored by the δ-diagram failing for the induced map $[\![\lfloor \_ \rfloor^t_=]\!]$ which consequently does *not* respect the structure of E, conceptually underpinning the decision not to include the operator.

Although the admissible operators exclude some undesirable operators, they do not provide a complete description of all possible well-behaved rules. A simple counter-example is given by the following function

$$[\![\sigma(e)]\!] = \lambda t. \begin{cases} \sigma(e(0)) & \text{if } t = 0 \\ \sigma'(e(0)) & \text{if } t > 0 \end{cases}$$

or, spelled out as a schematic time rule (again with t only ranging over $\mathcal{T}_>$):

$$\frac{-}{\sigma(x) \overset{t}{\rightsquigarrow} \sigma'(x)}$$

It is trivial to check that $[\![\sigma]\!]$ is natural and respects the structure of E, yet, for $\sigma \neq \sigma'$, the rule does not fit any of the rule shapes since the top-most operator is changed in the conclusion of the rule (from σ to σ').

The reason for this restriction on admissible operators is the δ-diagram. Intuitively demanding that the rules satisfy continuity, changes in the topmost operator would cause problems, especially in the case of *cyclic dependencies* between operators. Assume, for instance, that there is a time transition from a term with σ as its topmost operator to a term with σ' on top, and vice versa; once including such cases, it seems almost impossible to *syntactically* guarantee continuity. Besides, all relevant operators from the literature fit

the format anyway, so it seems general enough as it is. Furthermore, after repeated attempts, it seems that more permissive rule shapes or operator formats invariably invalidate well-definedness or the δ-diagram.

3.1 Refining the Rule Shapes

Even though arguably expressive enough, the admissible operators are by no means as general as they could be. As an illustrative example consider the following 'speed-halving' operator, as suggested by an anonymous referee:

$$\frac{p \overset{t}{\rightsquigarrow} p'}{\sigma(p) \overset{t+t}{\rightsquigarrow} \sigma(p')} \tag{7}$$

This operator can be described by the function

$$\begin{aligned} [\![\sigma]\!] : \Sigma E X &\to E(X + \Sigma X) \\ e &\mapsto \lambda t. \begin{cases} \sigma(e(t)) & \text{if } \exists u.\ u + u = t \ \wedge\ e(u)\downarrow \\ \text{undef} & \text{otherwise} \end{cases} \end{aligned}$$

which, in general, is not well-defined: for $\mathcal{T} = \mathbb{N}$, $[\![\sigma]\!]$ potentially allows the derivation of a $\overset{2}{\rightsquigarrow}$-transition (in case $e(1)\downarrow$), yet never of a $\overset{1}{\rightsquigarrow}$-transition (there is no $u \in \mathbb{N}$ such that $u + u = 1$) and therefore, axiom (5) of evolutions would not hold for $[\![\sigma]\!](e)$. However, when considered over the time domain $\mathbb{R}_{\geq 0}$, $[\![\sigma]\!]$ is natural, it fits the type (2), and it respects the structure of E. Thus, for the time domain $\mathbb{R}_{\geq 0}$ and the specific *time transformation* $t \mapsto t + t$ on it, (7) results in a well-behaved operator. This can be generalised as sketched in the following tentative development, allowing rule shapes parameterised by a time domain $\mathcal{T}$ and 'well-behaved' transformations $\mathcal{T} \to \mathcal{T}$.

Let $f : \mathcal{T} \to \mathcal{T}$ be a monoid homomorphism. The *image* of f is the set $f[\mathcal{T}] \overset{\text{df}}{=} \{s \in \mathcal{T} \mid \exists t \in \mathcal{T}.\ s = f(t)\}$; it is necessarily a submonoid of $\mathcal{T}$, hence non-empty, always containing 0. An *order ideal* on $\mathcal{T}$ is a *downward closed* subset I of $\mathcal{T}$ (with respect to the induced pre-order $\leq$ on $\mathcal{T}$, cf. Def. 2.1), i.e., $t \in I$ and $u \leq t$ imply $u \in I$.

Assume now that $f : \mathcal{T} \to \mathcal{T}$ is a monoid homomorphism which is injective, and whose image is an order ideal. Spelled out in concrete terms, the second requirement states that if there exists $s_t \in \mathcal{T}$ such that $f(s_t) = t$ and $u \leq t$ then there also exists $s_u \in \mathcal{T}$ with $f(s_u) = u$. Given such an f, the following rule shape then constitutes an admissible operator *with respect to* $\mathcal{T}$:

$$\frac{x \overset{t}{\rightsquigarrow} x'}{\sigma(x) \overset{f(t)}{\rightsquigarrow} \sigma(x')}$$

By the injectivity of f, its inverse function $f^{-1} : f[\mathcal{T}] \to \mathcal{T}$ exists and, by the additional properties, is also an injective homomorphism, so the above rule

can be translated into the well-defined map

$$\begin{aligned}[\![\sigma]\!] : EX &\to E(X + \Sigma X),\\ e &\mapsto \lambda t.\ \begin{cases}\sigma(e(f^{-1}(t))) & \text{if } f^{-1}(t)\downarrow \ \wedge\ e(f^{-1}(t))\downarrow\\ \text{undef} & \text{if } f^{-1}(t)\uparrow \ \vee\ e(f^{-1}(t))\uparrow\end{cases}\end{aligned}$$

which also respects the structure of E.

For $\mathbb{N}$, the only such f is the identity function, while for $\mathbb{R}_{\geq 0}$, we conjecture that only functions of the form $t \mapsto r \cdot t$ for $r \neq 0$ (including the identity for $r = 1$, and (7) for $r = 2$) satisfy all the conditions, i.e., that only speeding-up/slowing-down by a constant factor is allowed. It remains to be investigated in how far such functions f can be used in the previously presented rule shapes, including a necessary generalisation to the n-ary case.

4 Comonadic SOS

Taking a more abstract point of view, this section now presents a general account of abstract rules for an arbitrary behaviour comonad $D = (D, \varepsilon, \delta)$ and freely generated syntax [2], based on natural transformations ρ of type (3), and thus having greater expressivity than (2), which in turn formed the basis of the schematic format of the previous section. The increased expressive power of (3) necessitates stronger assumptions on the rules ρ in order for them to respect the operations of the comonad D.

Definition 4.1 Let Σ, F be endofunctors, Σ freely generating the monad $T = (T, \eta, \mu)$ with free Σ-algebra structure $\gamma : \Sigma T \Rightarrow T$. Given a natural transformation $\rho : \Sigma F \Rightarrow FT$, define $\ell_\rho : TF \Rightarrow FT$ as the unique map making the following diagram commute (obtained by the freeness of T):

(8)
$$\begin{array}{ccccc} F & \xRightarrow{\eta_F} & TF & \xLeftarrow{\gamma_F} & \Sigma TF \\ & \searrow^{F\eta} & \Downarrow \ell_\rho & & \Downarrow \Sigma\ell \\ & & FT & \xLeftarrow[F\mu]{} & FT^2 \xLeftarrow[\rho_T]{} \Sigma FT \end{array}$$

Call ℓ_ρ the *distributive law* induced by ρ.

Proposition 4.2 *Let Σ, F, T and ℓ_ρ be as in Def. 4.1. Then $\ell_\rho : TF \Rightarrow FT$ is a distributive law of the monad T over the functor F.*

Using ℓ_ρ, it is now possible to formulate conditions under which rules as in (3) respect the structure of D:

[2] Dualising this approach by using an arbitrary (syntax) monad T, e.g., terms over a signature modulo some equational theory, but only a one-step behaviour endofunctor B as in [15], seems closely related to [5] where so-called *structured* transition systems (where the set of states is endowed with some algebraic structure given by T and its algebras) were considered by using B-coalgebras on the category of T-algebras.

Definition 4.3 Let Σ, T be as in Def. 4.1, and define the natural transformation $\xi : \Sigma \Rightarrow T$ by $\xi \stackrel{\mathsf{df}}{=} \gamma \circ \Sigma\eta$. Let, furthermore, $D = (D, \varepsilon, \delta)$ be a comonad and $\rho : \Sigma D \Rightarrow DT$ be a natural transformation with induced distributive law $\ell_\rho : TD \Rightarrow DT$ of T over D (regarded as an endofunctor). Say then that ρ *respects the structure of the comonad* D if the following two diagrams (again referred to as the ε- and δ-diagram, respectively) commute:

(9)
$$\begin{array}{ccc} \Sigma D & \xRightarrow{\rho} & DT \\ {\scriptstyle \Sigma\varepsilon}\Downarrow & & \Downarrow{\scriptstyle \varepsilon_T} \\ \Sigma & \xRightarrow[\xi]{} & T \end{array} \qquad \begin{array}{ccccc} \Sigma D & & \xRightarrow{\qquad\rho\qquad} & & DT \\ {\scriptstyle \Sigma\delta}\Downarrow & & & & \Downarrow{\scriptstyle \delta_T} \\ \Sigma D^2 & \xRightarrow[\rho_D]{} & DTD & \xRightarrow[D\ell_\rho]{} & D^2T \end{array}$$

Abstract comonadic SOS (CSOS) rules for D are then given by a natural transformation $\rho : \Sigma D \Rightarrow DT$ respecting the structure of D.

Theorem 4.4 *Assume* $\rho : \Sigma D \Rightarrow DT$ *respects the structure of the comonad* $D = (D, \varepsilon, \delta)$*. Then the induced distributive law* $\ell_\rho : TD \Rightarrow DT$ *is a distributive law of the monad* T *over the comonad* D*.*

Thm. 4.4 states that abstract CSOS rules induce a distributive law of free syntax over the (arbitrary) behaviour comonad D. Hence it extends the result of Prop. 4.2 in the case that the endofunctor F is a comonad D such that the rules ρ respect the comonad structure. It is also a more general result than Thm. 2.4, since each natural transformation of type (2) can be extended to one of type (3), but not vice versa. The main result of [15], that abstract GSOS rules (1) induce a distributive law of free syntax over cofree behaviour, is also covered by Thm. 4.4 when instantiating D with the cofree comonad on a behaviour functor B, e.g., for timed processes over discrete time, in which case the evolution comonad is cofreely generated from $B_{\mathbb{N}} \stackrel{\mathsf{df}}{=} 1 + \_$ (see [8]). In the following, the generality of abstract CSOS rules is made even more precise: such rules are in fact already *equivalent* to a distributive law of a free monad over a comonad, providing the converse of Thm. 4.4.

Lemma 4.5 *Let* Σ, F*, and* T *be as above. Then natural transformations of type* $\rho : \Sigma F \Rightarrow FT$ *are in one-to-one correspondence with distributive laws* $\ell : TF \Rightarrow FT$ *of the monad* T *over the endofunctor* F*.*

Proof (Sketch) *The correspondence is defined as follows, leaving the proof that the two maps are mutually inverse to the reader. Given* ρ*, use the induced distributive law* $\ell_\rho : TF \Rightarrow FT$*; Prop. 4.2 then shows that* ℓ *respects the operations of the monad* T*. In the converse direction, given* $\ell : TF \Rightarrow FT$*, consider* $\rho_\ell = \ell \circ \xi_F : \Sigma F \Rightarrow FT$*.* □

Substituting D for F, and adding the requirement that ρ respects the structure of D, the one-to-one correspondence from Lemma 4.5 extends to distributing free monads over comonads:

Theorem 4.6 *Let Σ, T, and D be as above. Then there is a one-to-one correspondence between abstract CSOS rules $\rho : \Sigma D \Rightarrow DT$ and distributive laws $\ell : TD \Rightarrow DT$ of the freely generated monad T over the comonad D.*

Proof (Sketch) *The assumptions* (9) *allow to deduce that the induced distributive law ℓ_ρ is a distributive law of a monad over a comonad (not just an endofunctor), and in the other direction, if ℓ distributes a monad over a comonad, then ρ_ℓ satisfies both diagrams in* (9)*. So the correspondence from Prop. 4.5 extends to the case of freely generated monads and comonads.* □

Intuitively, Thm. 4.6 states that, in the case of T being freely generated, the conditions in (9) are precisely the necessary and sufficient conditions on abstract rules of type $\Sigma D \Rightarrow DT$ allowing the bialgebraic approach of [15] to be applied; in other words: there is no way to use strictly more expressive abstract rules and still obtain a distributive law $TD \Rightarrow DT$ in the case that T is freely generated.

Note that, as already stated in [8], abstract CSOS rules (3) and the rules (2) as in [8] form the extreme cases of a hierarchy of types for abstract rules from which, under global constraints like (9) or (6), ensure that one obtains a distributive law of T over D.

5 CSOS for Timed Processes

As an application of CSOS rules, this section presents a syntactic characterisation of CSOS rules for the evolution comonad E. The format is based on the notion of a *meta rule*, which will serve as a convenient abbreviation for infinite sets of infinitary rules. Even so, the format will still consist of infinitely many such meta rules to completely capture all possible abstract rules.

In the following, fix a non-trivial time domain $\mathcal{T}$, hence $\mathcal{T}$ has infinite cardinality, see [8]. Furthermore, let $\mathcal{V}$ be a set of variables with $|\mathcal{V}| = |\mathcal{T}|$, and for each $n \geq 1$ fix an *n-ary enumeration* of $\mathcal{V}$ without repetitions: disjoint subsets $\mathcal{V}_i = \{x_i^t \mid t \in \mathcal{T}\} \subseteq \mathcal{V}$ for each $1 \leq i \leq n$. For a set X and an evolution $e \in EX$, the *domain* and *range* of e are $\mathrm{dom}(e) = \{t \in \mathcal{T} \mid e(t)\downarrow\} \subseteq \mathcal{T}$ and $\mathrm{rng}(e) = e(\mathcal{T}) \subseteq X$, respectively, extended for $\boldsymbol{e} = (e_1, \ldots, e_n) \in (EX)^n$ by $\mathrm{dom}(\boldsymbol{e}) = (\mathrm{dom}(e_1), \ldots, \mathrm{dom}(e_n))$ and $\mathrm{rng}(\boldsymbol{e}) = \bigcup_{i=1}^n \mathrm{rng}(e_i)$. If $e \in ETX$ is an evolution on terms, the *variables* $\mathrm{vars}(e)$ of e are all the variables occurring in the terms in $\mathrm{rng}(e) \subseteq TX$. For tuples $\boldsymbol{e} \in (ETX)^n$, define $\mathrm{vars}(\boldsymbol{e}) = \bigcup_{i=1}^n \mathrm{vars}(e_i)$. A tuple $\boldsymbol{e} \in (EX)^n$ is *generic* if all the e_i are injective and have pairwise disjoint ranges. For $n \in \mathbb{N}$ and some n-ary domain, the n-ary *canonical* tuple for that domain is given by n evolutions ϵ_i such that the ϵ_i have the desired domain, and their values are given as $\epsilon_i(t) = x_i^t \in \mathcal{V}_i \subseteq \mathcal{V}$. In the following, canonical tuples are denoted by $\boldsymbol{\epsilon}$. Since none of the n-ary enumerations on $\mathcal{V}$ contains repetitions, canonical tuples are generic tuples with canonical names. Note that for $\boldsymbol{e} \in (EX)^n$, there is a *unique* corresponding canonical tuple $\boldsymbol{\epsilon} \in (E\mathcal{V})^n$ with the same domain.

Intuitively, generic tuples will serve the same purpose in the new format as using distinct variables did in GSOS rules, viz., ensuring that rules treat argument processes 'anonymously': although instantiating a rule with the same processes in different argument places is certainly possible, the rules cannot *demand* such identifications like, e.g., defining transitions only if two arguments are equal. As a consequence of this 'anonymity' for generic tuples $\boldsymbol{e}$, each element in $\mathrm{rng}(\boldsymbol{e})$ can be uniquely 'traced back' to some e_i, and canonical tuples $\boldsymbol{\epsilon}$ are truly canonical: they can be transformed into all other tuples of the same domain.

Lemma 5.1 (i) *If $\boldsymbol{e} \in (EX)^n$ is a generic tuple and $x \in \mathrm{rng}(\boldsymbol{e})$ then there exist unique $1 \leq i \leq n$ and $t \in \mathcal{T}$ such that $x = e_i(t)$.*

(ii) *Each tuple $\boldsymbol{e} \in (EX)^n$, with corresponding canonical tuple $\boldsymbol{\epsilon}$, induces a unique map $\varphi_{\boldsymbol{e}} : \mathrm{rng}(\boldsymbol{\epsilon}) \to \mathrm{rng}(\boldsymbol{e})$ such that $E\varphi_{\boldsymbol{e}}(\epsilon_i) = \varphi_{\boldsymbol{e}} \circ \epsilon_i = e_i$.*

Later on, $\varphi_{\boldsymbol{e}}$ will be extended to a function of type $\mathcal{V} \to X$ by arbitrarily assigning values to variables not contained in the range of $\boldsymbol{\epsilon}$.

Definition 5.2 Let Σ be a signature, $\sigma \in \Sigma$ be an n-ary function symbol, $\boldsymbol{e} \in (EX)^n$, and $\vartheta \in ETX$. Then an expression of the form

$$\sigma(e_1, \ldots, e_n) \Longrightarrow \vartheta \tag{10}$$

is a *meta rule for σ*. In the following, write $\theta_t = \vartheta(t) \in TX$ for $t \in \mathrm{dom}(\vartheta)$. The domain of a meta rule (10) is the same as $\mathrm{dom}(\boldsymbol{e})$ and it is generic (resp. canonical) if $\boldsymbol{e}$ is a generic (canonical) tuple of evolutions.

Each meta rule (10) is an abbreviation of the (infinite) set of infinitary time rules, ranging over $t \in \mathrm{dom}(\vartheta)$, of the form

$$\frac{\{e_i(0) \overset{t_i}{\rightsquigarrow} e_i(t_i) \,|\, t_i \in \mathcal{T} \wedge e_i(t_i)\downarrow\}_{1\leq i\leq n} \quad \{e_i(0) \overset{t_i}{\not\rightsquigarrow} \,|\, t_i \in \mathcal{T} \wedge e_i(t_i)\uparrow\}_{1\leq i\leq n}}{\sigma(e_1(0), \ldots, e_n(0)) \overset{t}{\rightsquigarrow} \theta_t} \tag{11}$$

occasionally abbreviated as $\sigma(\boldsymbol{e}) \overset{t}{\rightsquigarrow} \theta_t$, blurring the distinction between rule and rule conclusions. Note that each meta rule contains a *complete* (or *global*) description of the arguments' behaviour, not just local tests for the presence or absence of specific time transitions (like in the schematic format). This is in line with the interpretation of behaviour comonads modelling global behaviour. The following development will be based entirely on meta rules to make it more concise; all of it can also be carried out using standard time rules, via the correspondence (11).

Definition 5.3 A meta rule $\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta$ is a *GSOS meta rule* if it is canonical and if $\mathrm{vars}(\vartheta) \subseteq \mathrm{rng}(\boldsymbol{\epsilon})$. A set of meta rules over a signature Σ is *complete* (resp. *deterministic*) if for each n-ary function symbol $\sigma \in \Sigma$ and each n-ary domain, there is at least (at most) one meta rule for σ. A set of deterministic, complete GSOS meta rules is called *admissible*.

It follows immediately that an admissible set of meta rules contains precisely one meta rule for each σ and each appropriate domain. The terminology 'GSOS meta rule' is used because then each induced time rule (11) is an (infinitary) GSOS rule: all the ϵ_i have disjoint ranges, therefore variables occurring in the premises are distinct, in particular the $\epsilon_i(0)$; moreover, since $\mathrm{vars}(\vartheta) \subseteq \mathrm{rng}(\boldsymbol{\epsilon})$, each variable occurring in some θ_t must occur in the premises. Note that this property does not depend on $\boldsymbol{\epsilon}$ being canonical, it also holds for generic tuples $\boldsymbol{e}$ such that $\mathrm{vars}(\vartheta) \subseteq \mathrm{rng}(\boldsymbol{e})$. However, canonical names are used to guarantee an exact one-to-one correspondence. The following results show that admissible sets of meta rules are a correct characterisation of natural transformations $\Sigma E \Rightarrow ET$.

Proposition 5.4 *Each admissible set R of meta rules induces a natural transformation $[\![R]\!] : \Sigma E \Rightarrow ET$.*

Proof (Sketch) *Define a map $[\![R]\!]_{\mathcal{V}} : \Sigma E\mathcal{V} \to ET\mathcal{V}$ at all canonical tuples (possible because R is admissible) by $[\![R]\!]_{\mathcal{V}}(\sigma(\boldsymbol{\epsilon})) = \vartheta$ iff R contains the meta rule $\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta$. This is well-defined and extends to a natural transformation $[\![R]\!]$ because all meta rules in R are GSOS and then using Lemma 5.1(ii).* □

Proposition 5.5 *Let $\rho : \Sigma E \Rightarrow ET$ be a natural transformation. Then ρ induces an admissible set $\langle\!\langle \rho \rangle\!\rangle$ of meta rules.*

Proof (Sketch) *Derive the meta rules in $\langle\!\langle \rho \rangle\!\rangle$ from the values of $\rho_{\mathcal{V}}$, via $(\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta) \in \langle\!\langle \rho \rangle\!\rangle$ iff $\rho_{\mathcal{V}}(\sigma(\boldsymbol{\epsilon})) = \vartheta$. Since ρ is natural, $\langle\!\langle \rho \rangle\!\rangle$ is admissible.* □

The following theorem follows from using canonical tuples to describe admissible sets of meta rules.

Theorem 5.6 *The two constructions $R \mapsto [\![R]\!]$ and $\rho \mapsto \langle\!\langle \rho \rangle\!\rangle$ are mutually inverse. Hence, there is a one-to-one correspondence between admissible sets of meta rules and natural transformations of type $\Sigma E \Rightarrow ET$.*

Note that under the correspondence, $(\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta) \in R$ iff $[\![R]\!]_{\mathcal{V}}(\sigma(\boldsymbol{\epsilon})) = \vartheta$, and $\rho_{\mathcal{V}}(\sigma(\boldsymbol{\epsilon})) = \vartheta$ iff $(\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta) \in \langle\!\langle \rho \rangle\!\rangle$, for an admissible set R of meta rules and a natural transformation $\rho : \Sigma E \Rightarrow ET$, respectively. The next goal for the format is a (meta) rule-based characterisation of the ε-diagram in (9):

Definition 5.7 A meta rule $\sigma(\boldsymbol{e}) \Longrightarrow \vartheta$ is *co-pointed* if it satisfies

$$\theta_0 = \vartheta(0) = \sigma(e_1(0), \ldots, e_n(0))$$

Call a set R of meta rules co-pointed if each meta rule in R is.

Simple inspection yields:

Theorem 5.8 *Admissible co-pointed sets of meta rules are in one-to-one correspondence with natural transformations $\rho : \Sigma E \Rightarrow ET$ additionally satisfying the ε-diagram in* (9).

The final task is to produce a similar characterisation of the δ-diagram in (9) where the maps ρ_E and ℓ_ρ are used, the latter being defined in terms of ρ_T. In the following, the values of these maps, for specific arguments, are expressed in terms of $\rho_{\mathcal{V}}$. Since this last component of the rules is at the core of the correspondence established in Thm. 5.6, a match between the diagram and (conditions on) meta rules is obtained.

Definition 5.9 Let Σ be a signature and $\theta \in TX$ be a term over Σ and some set X. Let $f : X \rightharpoonup Y$ be partial function such that $\mathrm{vars}(\theta) \subseteq \mathrm{dom}(f)$. Then denote the simultaneous substitution of $f(x_i)$ for $x_i \in \mathrm{vars}(\theta)$ in θ by $\theta[f]$ or $\theta[f(x_i)/x_i]$. Extending this notion to evolutions $\vartheta \in ETX$, write $\vartheta[f]$ or $\vartheta[f(x_i)/x_i]$ to denote the evolution on TY whose value at each $t \in \mathrm{dom}(\vartheta)$ is the term $\theta_t[f]$, subject to the condition $\mathrm{vars}(\vartheta) \subseteq \mathrm{dom}(f)$.

Lemma 5.10 *Assume that: $\boldsymbol{\epsilon}$ is a canonical n-ary tuple and $\boldsymbol{\vartheta} \in (ET\mathcal{V})^n$ has the same domain as $\boldsymbol{\epsilon}$; $\varphi_{\boldsymbol{\delta\epsilon}} : \mathcal{V} \to E\mathcal{V}$ and $\varphi_{\boldsymbol{\vartheta}} : \mathcal{V} \to T\mathcal{V}$ are (extensions of) the functions obtained from Lemma 5.1(ii), mapping $\boldsymbol{\epsilon}$ to $\boldsymbol{\delta\epsilon}$ and $\boldsymbol{\vartheta}$, respectively; $\rho : \Sigma E \Rightarrow ET$ is a natural transformation with induced admissible set $R = \langle\!\langle \rho \rangle\!\rangle$ of meta rules; Σ is a signature and $\sigma \in \Sigma$ is an n-ary function symbol. Then, using the naturality of ρ, one obtains the following characterisations:*

(i) $\rho_{E\mathcal{V}}(\sigma(\boldsymbol{\delta\epsilon})) = \vartheta[\varphi_{\boldsymbol{\delta\epsilon}}] = \vartheta[\epsilon_i + t/x_i^t]$ *if and only if* $\rho_{\mathcal{V}}(\sigma(\boldsymbol{\epsilon})) = \vartheta$.

(ii) $\rho_T(\sigma(\boldsymbol{\vartheta})) = \vartheta[\varphi_{\boldsymbol{\vartheta}}] = \vartheta[\vartheta_i(t)/x_i^t]$ *if and only if* $\rho_{\mathcal{V}}(\sigma(\boldsymbol{\epsilon})) = \vartheta$.

Definition 5.11 Given a set R of canonical meta rules, define the notion of *R-derivation* as follows. For $\xi \in TEX$ and $\vartheta \in ETX$, write $R \vdash \xi \Longrightarrow \vartheta$ if there is a finite proof using only the two following rules (where, for simplicity, any reference to unit η and multiplication μ of T is omitted):

(i) If $\xi = e$ for some $e \in EX$ then $R \vdash e \Longrightarrow e$

(ii) If $\sigma \in \Sigma$ is an n-ary function symbol and $\xi_1, \ldots, \xi_n \in TEX$ then

$$\frac{\{R \vdash \xi_i \Longrightarrow \vartheta_i,\ \mathrm{dom}(\vartheta_i) = \mathrm{dom}(\epsilon_i)\}_{1 \le i \le n} \quad (\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta) \in R}{R \vdash \sigma(\boldsymbol{\xi}) \Longrightarrow \vartheta[\varphi_{\boldsymbol{\vartheta}}]}$$

In particular, if $(\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta) \in R$, then $R \vdash \sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta$, so one could call the expression $R \vdash \sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta$ an *axiom*. Note that it is not necessary that R is admissible for R-derivations to make sense, but one obtains:

Lemma 5.12 *Let R be an admissible set of meta rules and $\rho = [\![R]\!]$ be its corresponding natural transformation with induced distributive law ℓ_ρ. Then the following equivalence holds for $\xi \in TEX$ and $\vartheta \in ETX$:*

$$(\ell_\rho)_X(\xi) = \vartheta \ \Leftrightarrow\ R \vdash \xi \Longrightarrow \vartheta$$

Proof (Sketch) *By induction on the structure of ξ. Two cases arise, as in the definition of ℓ_ρ, and they are taken care of by the two corresponding rules for R-derivations.* □

Intuitively, R-derivations (or equivalently $\ell_{[\![R]\!]}$ for admissible R) capture the notion of *provability* from a set of rules. Meta rules only apply to 'simple' terms with exactly one function symbol. The *inductive extension* given by the R-derivations then determines the action of the rules on complex terms, iterating applications of the rules (subject to necessary substitutions).

For admissible R, R-derivations describe the natural transformation ℓ_ρ. Hence, whenever $R \vdash \xi \Longrightarrow \vartheta$ (or equivalently, $\ell_\rho(\xi) = \vartheta$), it must hold that $\mathrm{vars}(\vartheta) \subseteq \mathrm{vars}(\xi)$, with $\mathrm{vars}(\xi)$ the suitably defined set of all variables from X occurring in $\xi \in TEX$. Otherwise, ℓ_ρ could not be natural, by the same argument as showing that GSOS (meta) rules induce a natural transformation.

Definition 5.13 Let R be a set of canonical meta rules, Σ a signature, $\sigma \in \Sigma$ an n-ary function symbol, $\sigma(\boldsymbol{\epsilon}) \Longrightarrow \vartheta$ a meta rule in R, and $t, u \in \mathcal{T}$. Then R is called *continuous* if the following two statements are equivalent:

(i) $\sigma(\boldsymbol{\epsilon}) \overset{t+u}{\rightsquigarrow} \theta_{t+u}$, and

(ii) $\sigma(\boldsymbol{\epsilon}) \overset{t}{\rightsquigarrow} \theta_t \wedge\ R \vdash \theta_t[\varphi_{\boldsymbol{\delta\epsilon}}] \Longrightarrow \vartheta' \wedge\ \vartheta'(u) = \theta_{t+u}$

The terminology 'continuous' is used because the required equivalence in Def. 5.13 is a generalised, meta rule-based version of time continuity: if one rule application allows to derive a $\overset{t+u}{\rightsquigarrow}$-transition, it must be possible to first derive a $\overset{t}{\rightsquigarrow}$-transition in one step, followed by a derivation (of arbitrary finite length) of a $\overset{u}{\rightsquigarrow}$-transition. This use of derivations also precisely marks the difference between the two δ-diagrams in (6) and (9): the former specifies that the $\overset{u}{\rightsquigarrow}$-transition must be derivable at once, whereas the latter, as just stated, allows several steps to derive the transition. This is due to the fact that the abstract rules in (2) only allow terms with at most one function symbol in rule conclusions, so at most one rule application is necessary/possible, whereas in (3), arbitrary terms are allowed. Note that continuous sets of meta rules are not necessarily admissible, yet using Lemmas 5.10 and 5.12, one deduces:

Theorem 5.14 *There is a one-to-one correspondence between admissible continuous sets of meta rules and natural transformations $\Sigma E \Rightarrow ET$ satisfying the δ-diagram.*

Corollary 5.15 *There is a one-to-one correspondence between abstract CSOS rules for timed processes and admissible, co-pointed, and continuous sets of meta rules.*

As already shown, the schematic format induces CSOS rules for E, hence:

Corollary 5.16 *The schematic format induces an admissible, co-pointed and continuous set of meta rules.*

There is also a concrete way to derive the set of meta rules corresponding to one of the admissible operators from §3, again illustrated by the case of TeCCS's strong choice + operator with associated map

$$[\![+]\!] : (E\mathcal{V})^2 \to E(\mathcal{V} + \Sigma\mathcal{V}) \subseteq ET\mathcal{V}$$

which was already shown to respect the structure of E in Prop. 3.4. Let (ϵ_1, ϵ_2) be an arbitrary canonical tuple. Then $[\![+]\!]$ induces the following meta rule:

$$\epsilon_1 + \epsilon_2 \Longrightarrow \vartheta \stackrel{\mathsf{df}}{=} \lambda t. \begin{cases} x_1^t + x_2^t & \text{if } \epsilon_1(t)\downarrow \wedge\ \epsilon_2(t)\downarrow \\ \text{undef} & \text{if } \epsilon_1(t)\uparrow \vee\ \epsilon_2(t)\uparrow \end{cases}$$

Adding such a meta rule for each canonical tuple, $[\![+]\!]$, and more generally any admissible operator from §3, yields an admissible set of meta rules.

Future Work

We are currently investigating how to treat calculi where action and time transitions are not independent, using *products of comonads* together with abstract rules like (2) or (3). Apart from the remaining open problems mentioned in [8], most importantly how to deal with timed automata [2], the parameterised version of the schematic format from §3 should be extended, in particular in such a way as to recover the presently used rule shapes. Finally, there is the question whether one can find a different abstract model of timed processes compatible with the bialgebraic approach of [15], yet avoiding the various levels of infinity as exhibited by the complete format (using infinite sets of infinitary rules); very speculatively, examining potential connections with Fiore's work on hybrid systems [6] might prove useful in this direction.

Acknowledgements. The author would like to thank Daniele Turi and Gordon Plotkin for many helpful discussions.

References

[1] P. Aczel and P. F. Mendler. A final coalgebra theorem. In D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts, and A. Poigné, editors, *Proceedings of the Conference on Category Theory and Computer Science*, volume 389 of *LNCS*, pages 357–365, Berlin, September 1989. Springer-Verlag.

[2] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 25 April 1994. Fundamental Study.

[3] F. Bartels. GSOS for probabilistic transition systems (extended abstract). In Larry Moss, editor, *Proc. Coalgebraic Methods in Computer Science (CMCS 2002)*, volume 65 of *Electronic Notes in Theoretical Computer Science*, 2002.

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

[5] A. Corradini, R. Heckel, and U. Montanari. Compositional SOS and beyond: a coalgebraic view of open systems. *Theoretical Computer Science*, 280(1–2):163–192, May 2002. Special issue with selected papers from [7].

[6] M. Fiore. Fibred models of processes: Discrete, continuous, and hybrid systems. In *IFIP International Conference on Theoretical Computer Science*, volume 1872 of *Lecture Notes in Computer Science*, pages 457–473, 2000.

[7] B. Jacobs and J. Rutten, editors. *Coalgebraic Methods in Computer Science (CMCS'1999)*, volume 19 of *Electronic Notes in Theoretical Computer Science*, 1999.

[8] M. Kick. Bialgebraic modelling of timed processes. In P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo, editors, *Proceedings ICALP'02*, volume 2380 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002. Available from `http://www.dcs.ed.ac.uk/home/mk`.

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

[10] F. Moller and C. Tofts. A temporal calculus of communicating systems. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR '90 (Concurrency Theory)*, volume 458 of *Lecture Notes in Computer Science*, pages 401–415, Amsterdam, The Netherlands, August 1990. Springer-Verlag.

[11] X. Nicollin and J. Sifakis. An overview and synthesis on timed process algebras. In K.G. Larsen and A. Skou, editors, *Computer Aided Verification (CAV '91)*, volume 575 of *Lecture Notes in Computer Science*, pages 376–398, Aalborg, Denmark, July 1991. Springer-Verlag.

[12] G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, Aarhus, Denmark, September 1981.

[13] J. Power and H. Watanabe. Combining a monad and a comonad. *Theoretical Computer Science*, 280(1–2):137–162, May 2002. Special issue with selected papers from [7].

[14] D. Turi. *Functorial Operational Semantics and its Denotational Dual*. PhD thesis, Free University, Amsterdam, June 1996. Available from `http://www.dcs.ed.ac.uk/home/dt/`.

[15] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Twelfth Annual Symposium on Logic in Computer Science (LICS '97)*, pages 280–291, Warsaw, Poland, 29 June–2 July 1997. IEEE Computer Society Press.