Metric Semantics for True Concurrent Real Time

Christel Baier[a], Joost-Pieter Katoen[b] and Diego Latella[c]

[a] *Fakultät für Mathematik und Informatik, Universität Mannheim Seminargebäude A5, D-68159 Mannheim, Germany*

[b] *Lehrstuhl für Informatik VII, Friedrich-Alexander-Universität Erlangen-Nürnberg Martensstrasse 3, D-91058 Erlangen, Germany*

[c] *CNUCE Istituto del CNR, Via Santa Maria 36, I-56100 Pisa, Italy*

Abstract. This paper investigates the use of a complete metric space framework for providing denotational semantics to a real-time process algebra. The study is carried out in a non-interleaving setting and is based on a timed extension of Langerak's bundle event structures, a variant of Winskel's event structures. The distance function is based on the amount of time to which event structures do 'agree'. We show that this intuitive notion of distance is a pseudo metric (but not a metric) on the set of timed event structures. A generalisation to equivalence classes of timed event structures in which we abstract from event names and non-executable events (events that can never appear) is shown to be a complete ultra-metric space. We show that the resulting metric semantics is an abstraction of an existing cpo-based denotational and a related operational semantics for the considered language.

1 Introduction

In this paper we consider a metric denotational semantics for an algebraic specification language that besides concurrency, synchronisation, and non-determinism, encompasses the notion of real-time. The language that we consider is a real-time extension of a process algebra based on the standardised specification language LOTOS [7]. As semantic domain we take a timed extension (defined in [10]) of Langerak's bundle event structures, a variant of Winskel's event structures that has been shown to adequately deal with the operators of LOTOS (in particular, parallel composition and disruption) [11]. The suitability of this timed truly concurrent model for modelling time-critical systems is addressed in [10]. The metric approach of this paper can also be applied to timed variants of other brands of event structures, like prime and stable event structures.

The basic idea of this paper is to consider behaviours of event structures up to a certain time. This is in fact a continuous version of the idea in [12] to consider (untimed) event structures up to a certain depth (i.e. length of a causal chain). The distance function is based on the amount of time to which event structures do 'agree'. We show that this intuitive notion of distance is a pseudo metric (but not a metric) on TES, the set of timed event structures. As a first step towards obtaining a metric (rather than a pseudo metric), we consider TES modulo an isomorphism $\simeq_{iso}$ that abstracts from event names and from non-executable events, events that can never appear. Secondly, we refine this notion towards finitely approximable timed event structures module $\simeq_{iso}$ and show that this model is a complete ultra-metric space. The resulting domain is used as a semantic domain for time-guarded processes. A process is time-guarded if it cannot generate instantaneous recursive process instantiations. We show that the proposed metric semantics is an abstraction of the cpo-based semantics of [10].

2 A real-time process algebra

We assume a given set of observable actions Obs and an *invisible action* τ; $\tau \notin \mathsf{Obs}$. The action $\surd$ indicates the *successful termination* action of a process; $\surd \notin \mathsf{Obs}$ and $\surd \neq \tau$. In addition, let $\mathsf{Act} = \mathsf{Obs} \cup \{\tau, \surd\}$, $a \in \mathsf{Obs} \cup \{\tau\}$, $I \subseteq \mathbb{R}^+ \cup \{\infty\}$, $t \in \mathbb{R}^+ \cup \{\infty\}$, $A \subseteq \mathsf{Obs}$, $\lambda : \mathsf{Act} \longrightarrow \mathsf{Act}$ with $\lambda(\tau) = \tau$ and $\lambda(\surd) = \surd$, and Var a set of process variables with $x \in \mathsf{Var}$. The set of expressions defined in the following is denoted Expr.

$$P ::= \mathbf{0} \mid \mathbf{1} \mid a_I \,.\, P \mid P + P \mid P \,;\, P \mid P\,[> P \mid P \,||_A\, P \mid P \setminus A \mid P[\lambda] \mid P \rhd_t P \mid x.$$

$+$, $\setminus A$, and $[\lambda]$ are the usual process algebra operators choice, abstraction and relabelling, respectively.

- $\mathbf{1}$ represents the successful termination process; it can only perform action $\surd$ and then becomes $\mathbf{0}$, the process that cannot perform any action.
- $a_I \,.\, P$ denotes the prefix of a and P where a is allowed (but not forced) to occur at $t \in I$.
- $P \,;\, Q$ denotes the sequential composition of P and Q; the control is passed to Q by the termination of P as indicated by the occurrence of $\surd$.
- $P\,[> Q$ denotes the disruption of P by Q; i.e. P may at any point of its execution be disrupted by Q, unless P has terminated.
- $P \,||_A\, Q$ denotes the parallel composition of P and Q; P and Q execute actions not in A independently from each other, while actions in A (and successful termination actions) must be performed by both processes simultaneously.
- $P \rhd_t Q$ initially behaves like P, but if P does not perform an action before time t (since its enabling) then a timeout occurs and control is passed to Q.

Using these operators a timed interrupt, for instance, can easily be modelled: the process $P\,[> (\mathbf{0} \rhd_t Q)$ specifies that P is disrupted by Q at time t, unless P has terminated before. Various case studies have proven that the timed operators like $a_I \,.\, P$ and $P \rhd_t Q$ are convenient to specify practical real-time systems [1, 18].

Process variables are considered in the context of a set of process definitions of the form $x := P$, where P might contain occurrences of x or of other process variables. For process variable x let $decl(x)$ denote the body of x, i.e. $decl(x) = P$ for $x := P$. A *process* is a pair $\langle decl, P \rangle$ consisting of a declaration $decl : \mathsf{Var} \longrightarrow \mathsf{Expr}$ and an expression $P \in \mathsf{Expr}$. PA denotes the set of all processes.

3 Timed event structures

Event structures consist of *events* labelled with actions (an event modelling the occurrence of its action), together with relations of causality and conflict between events. We take Langerak's (extended bundle) event structures [11] and equip this with timing information. Event structures incorporate a *conflict* relation (denoted $\rightsquigarrow$) that—as opposed to what is common in other types of event structures—is not required to be symmetric and a *bundle* relation (denoted $\mapsto$) for modelling causality.

The meaning of $e \rightsquigarrow e'$ is that (i) if e' occurs it disables the occurrence of e, and (ii) if e and e' both occur in a single system run then e precedes e'. $e \rightsquigarrow e'$ and $e' \rightsquigarrow e$ is equivalent with $e \,\#\, e'$, the usual symmetric conflict in event structures. The reason for adopting $\rightsquigarrow$ rather than $\#$ is to model the disrupt operator $[>$ adequately.

Causality is represented by the bundle relation. For set X of events and an event e, $X \mapsto e$ means that if e happens in a system run, some event in X must have happened before. X is called the *bundle set* and we use $\mapsto$ to denote the set of bundles of an event structure. The reason for not having a binary causality relation between events (as in prime event structures [16]) is to model parallel composition $||_A$ in a less complex way.

Time is added to event structures in the following way [10]. Relative delays between events are attached to bundles, and delays relative to the start are attached to events. Delays determine when an event may happen, they do not specify that an event should happen at a particular time. For the latter purpose we use *urgent* events; an urgent event should happen as soon as it is enabled.

Definition 1. A *timed event structure* (tes) $\mathcal{E}$ is a tuple $(E, \rightsquigarrow, \mapsto, l, \mathcal{A}, \mathcal{R}, \mathcal{U})$ with E, a set of *events*, $\rightsquigarrow \subseteq E \times E$, the (irreflexive) *conflict* relation, $\mapsto \subseteq \mathcal{P}(E) \times E$, the *bundle* relation, $l : E \longrightarrow \mathsf{Act}$, the *labelling* function, $\mathcal{A} : E \longrightarrow \mathcal{P}(\mathbb{R}^+ \cup \{ \infty \})$, the *event delay* function, $\mathcal{R} : \mapsto \longrightarrow \mathcal{P}(\mathbb{R}^+ \cup \{ \infty \})$, the *bundle delay* function, and $\mathcal{U} \subseteq E$, the set of *urgent* events, such that:

(P1) $(X \times X) \setminus \mathrm{Id}_E \subseteq \rightsquigarrow$ for any bundle set X

and for all $e \in \mathcal{U}$:

(P2) $\forall\, e' \in E, X \subseteq E : ((e' \rightsquigarrow e \;\vee\; e \rightsquigarrow e') \;\wedge\; X \mapsto e) \;\Rightarrow\; (X \mapsto e' \;\vee\; X \rightsquigarrow e')$
(P3) $\exists\, t \in \mathbb{R}^+ : (\mathcal{A}(e) \in \{ \varnothing, \{ t \} \}) \;\vee\; (\exists\, X : X \mapsto e \;\wedge\; \mathcal{R}(X, e) \in \{ \varnothing, \{ t \} \})$.

Here, $\mathcal{P}(\cdot)$ denotes the power-set function, $X \rightsquigarrow e'$ denotes $(\forall\, e'' \in X : e'' \rightsquigarrow e')$ and Id_E denotes the identity relation on set E. Note that $\varnothing \rightsquigarrow e'$ for all e'.

Event structures are depicted as follows. Events are denoted as dots; near the dot the action label is given. If no confusion arises we often use action labels rather than event identities to denote events. $e \rightsquigarrow e'$ is indicated by a dotted arrow from e to e'; if also $e' \rightsquigarrow e$, then a dotted line is drawn instead. A bundle $X \mapsto e$ is indicated by an arrow to which each event in X is connected via a line. Bundle and event delays are depicted near to a bundle and event, respectively. Urgent events are denoted by open dots, other events by closed dots. A bundle

$X \mapsto e$ with $\mathcal{R}(X, e) = I$ is denoted by $X \overset{I}{\mapsto} e$. Delays $[t, \infty)$ are simply denoted by t; delays $[0, \infty)$ are usually omitted. Figure 1(a) shows an example tes with e.g. $\{ a \} \overset{[0,7]}{\mapsto} b$ and $\{ a \} \overset{[0,5]}{\mapsto} c$, $b \rightsquigarrow \tau$ and $\tau \rightsquigarrow b$. $\mathcal{U} = \{ \tau \}$ and $\mathcal{A}$ is 0 for all events.

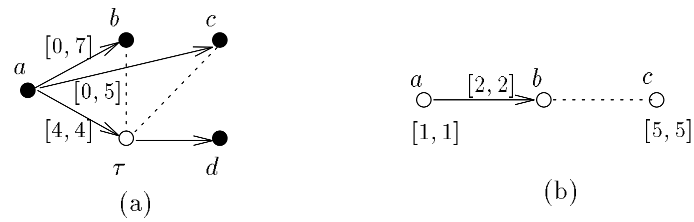


Fig. 1. A tes and a non-tes

The concept of a system run for tes's is captured by the notion of a *timed event trace*. For σ a sequence of distinct events let the set of events enabled in $\mathcal{E}$ after σ be defined as[1]

$$\mathsf{en}^{\mathcal{E}}(\sigma) \triangleq \{ e \in E \mid e \notin \sigma \;\wedge\; (\forall\, e_i \in \sigma : e \not\rightsquigarrow e_i) \;\wedge\; (\forall\, X \mapsto e : X \cap \sigma \neq \varnothing) \}.$$

The time instants at which an enabled event in $\mathcal{E}$ after $\sigma = (e_1, t_1) \ldots (e_n, t_n)$ could potentially happen is determined as

$$\mathsf{time}^{\mathcal{E}}_{\sigma}(e) \triangleq \mathcal{A}(e) \cap \bigcap_{e_i \rightsquigarrow e} [t_i, \infty) \cap \bigcap_{X \overset{I}{\mapsto} e,\, e_i \in X} t_i{+}I.$$

Definition 2. $\sigma = (e_1, t_1) \ldots (e_n, t_n)$ with $e_i \in E$ (all events being pairwise distinct) and $t_i \in \mathbb{R}^+$, is a *timed event trace* of $\mathcal{E} \in \mathsf{TES}$ iff for all $0 < i \leqslant n$:

1. $e_j \rightsquigarrow e_i \;\Rightarrow\; (j < i \;\wedge\; t_j \leqslant t_i)$ for all $0 < j \leqslant n$
2. $X \overset{I}{\mapsto} e_i \;\Rightarrow\; (\exists\, j : X \cap \{ e_1, \ldots, e_{i-1} \} = \{ e_j \} \;\wedge\; t_i \in t_j + I)$ for all $X \subseteq E$
3. $t_i \in \mathcal{A}(e_i)$
4. $(e_i \rightsquigarrow e \;\vee\; e \rightsquigarrow e_i) \;\Rightarrow\; t_i \leqslant \mathit{min}(\mathsf{time}^{\mathcal{E}}_{e_1 \ldots e_{i-1}}(e))$ for $e \in \mathcal{U} \cap \mathsf{en}^{\mathcal{E}}(e_1 \ldots e_{i-1})$.

The set of timed event traces of $\mathcal{E}$ is denoted by $\mathit{Traces}(\mathcal{E})$.

By convention we use $\mathit{min}\,\varnothing = \infty$. The last constraint takes care of the fact that urgent events may prevent the events that they disable (or by which they are disabled) to occur after a certain time. That is, event e_i can occur at time t_i provided there is no enabled urgent event e that disables e_i (or that is disabled by e_i) and that (if it occurs) must occur before t_i.

For example, for the following sequences of timed events the conditions are given under which they are timed event traces of Figure 1(a):

$$(a, t_a)\,(c, t_c)\,(b, t_b) \;\text{ if }\; 0 \leqslant t_a \leqslant t_c \leqslant t_b \;\wedge\; t_b \leqslant t_a{+}4 \;\wedge\; t_c \leqslant t_a{+}4$$
$$(a, t_a)\,(\tau, t_\tau)\,(d, t_d) \;\text{ if }\; 0 \leqslant t_a \leqslant t_\tau \leqslant t_d \;\wedge\; t_\tau = t_a{+}4.$$

[1] Often the set of events of a sequence is identified with the sequence itself.

Note that Figure 1(a) models a typical timeout scenario: if after the occurrence of a neither b nor c happen within 4 time units, then a timeout (τ) is forced to occur. It τ would not be urgent, the conditions for t_a and t_b in the first case would be $t_b \leqslant t_a+7$ and $t_c \leqslant t_a+5$, since τ is not forced to occur and time does not resolve the choice.

4 Operators for timed event structures

In this section we present some operators on timed event structures that are needed to define a compositional semantics for PA. They are basically adopted from [9, 10]. We start with some basic notions. Let Events be a set such that for all actions $a \in \mathsf{Act}$ there is an event $e_a \in \mathsf{Events}$, and (i) if $e \in \mathsf{Events}$ then $(e, *), (*, e) \in \mathsf{Events}$, and (ii) if $e, e' \in \mathsf{Events}$ then $(e, e') \in \mathsf{Events}$. Let TES denote the set of tes's $\mathcal{E}$ with $E \subseteq \mathsf{Events}$. Let $init(\mathcal{E})$ be the set of initial events of $\mathcal{E}$ and $exit(\mathcal{E})$ its set of successful termination events, i.e. $init(\mathcal{E}) \triangleq \{ e \in E \mid \neg(\exists X \subseteq E : X \mapsto e) \}$ and $exit(\mathcal{E}) \triangleq \{ e \in E \mid l(e) = \surd \}$.

In the rest of this section let $\mathcal{E} \in \mathsf{TES}$ and $\mathcal{E}_1 = (E_1, \rightsquigarrow_1, \mapsto_1, l_1, \mathcal{A}_1, \mathcal{R}_1, \mathcal{U}_1)$, $\mathcal{E}_2 = (E_2, \rightsquigarrow_2, \mapsto_2, l_2, \mathcal{A}_2, \mathcal{R}_2, \mathcal{U}_2)$ such that w.l.o.g. $E_1 \cap E_2 = \varnothing$. Let $\hat{\tau}$ denote the urgent variant of τ.

Definition 3. For $a \in \mathsf{Obs} \cup \{ \tau, \hat{\tau} \}$ and $I \subseteq [0, \infty)$, $a_I \,.\, \mathcal{E}_1 \triangleq (E_1 \cup \{ e_a \}, \rightsquigarrow_1, \mapsto, l_1 \cup \{ (e_a, a) \}, \mathcal{A}, \mathcal{R}, \mathcal{U})$ where for $e \notin E_1$

- $\mapsto = \mapsto_1 \cup (\{ \{ e \} \} \times E_1)$
- $\mathcal{A} = \{ (e, I) \} \cup (E_1 \times \{ [0, \infty) \})$
- $\mathcal{R} = \mathcal{R}_1 \cup \{ ((\{ e \}, e'), \mathcal{A}_1(e)) \mid e' \in E_1 \}$
- $\mathcal{U} = \mathsf{if}\ a = \hat{\tau}\ \mathsf{then}\ \mathcal{U}_1 \cup \{ e \}\ \mathsf{else}\ \mathcal{U}_1$.

$\hat{\tau}_I \,.\, \mathcal{E}$ denotes the prefixing of τ_I and $\mathcal{E}$ where e is declared to be urgent. The possibility $\hat{\tau}_I \,.\, \mathcal{E}$ is used to define the semantics of the timeout operator $\triangleright$ in a concise way. Notice that for $\hat{\tau}_I \,.\, \mathcal{E}$ set I must be either empty of equal to $[t, t]$ for some t in order to guarantee axiom (P3).

Definition 4. $\mathcal{E}_1 + \mathcal{E}_2 \triangleq (E_1 \cup E_2, \rightsquigarrow, \mapsto_1 \cup \mapsto_2, l_1 \cup l_2, \mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2, \mathcal{U}_1 \cup \mathcal{U}_2)$ where $\rightsquigarrow = \rightsquigarrow_1 \cup \rightsquigarrow_2 \cup (init(\mathcal{E}_1) \times init(\mathcal{E}_2)) \cup (init(\mathcal{E}_2) \times init(\mathcal{E}_1))$.

Definition 5. Let $A \subseteq \mathsf{Obs}$. Then $\mathcal{E} \setminus A \triangleq (E, \rightsquigarrow, \mapsto, l', \mathcal{A}, \mathcal{R}, \mathcal{U})$ where $(l(e) \in A \Rightarrow l'(e) = \tau) \wedge (l(e) \notin A \Rightarrow l'(e) = l(e))$.

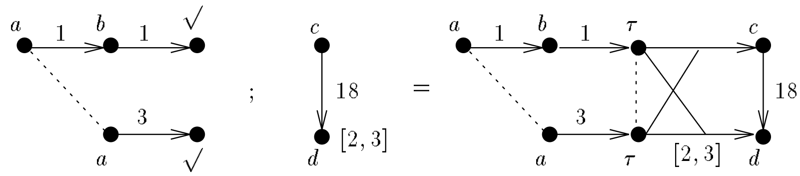
Definition 6. For $\lambda : \mathsf{Act} \longrightarrow \mathsf{Act}$ with $\lambda(\tau) = \tau$ and $\lambda(\surd) = \surd$ let $\mathcal{E}[\lambda] \triangleq (E, \rightsquigarrow, \mapsto, \lambda \circ l, \mathcal{A}, \mathcal{R}, \mathcal{U})$.

Definition 7. $\mathcal{E}_1 \,;\, \mathcal{E}_2 \triangleq (E_1 \cup E_2, \rightsquigarrow, \mapsto, l, \mathcal{A}, \mathcal{R}, \mathcal{U}_1 \cup \mathcal{U}_2)$ where

- $\rightsquigarrow = \rightsquigarrow_1 \cup \rightsquigarrow_2 \cup (exit(\mathcal{E}_1) \times exit(\mathcal{E}_1)) \setminus \mathrm{Id}_{E_1}$
- $\mapsto = \mapsto_1 \cup \mapsto_2 \cup (\{ exit(\mathcal{E}_1) \} \times E_2))$
- $l = ((l_1 \cup l_2) \setminus (exit(\mathcal{E}_1) \times \{ \surd \})) \cup (exit(\mathcal{E}_1) \times \{ \tau \})$
- $\mathcal{A} = \mathcal{A}_1 \cup (E_2 \times \{ [0, \infty) \})$

– $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \{ ((\mathit{exit}(\mathcal{E}_1), e), \mathcal{A}_2(e)) \mid e \in E_2) \}$.

As an example of how $\mathcal{E}_1 \,;\, \mathcal{E}_2$ is computed consider:



Notice that the delay of d in $\mathcal{E}_2$ now becomes relative to the termination of $\mathcal{E}_1$.

Definition 8. $\mathcal{E}_1 \,[\!>\, \mathcal{E}_2 \triangleq (E_1 \cup E_2, \rightsquigarrow, \mapsto_1 \cup \mapsto_2, l_1 \cup l_2, \mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2, \mathcal{U}_1 \cup \mathcal{U}_2)$ where $\rightsquigarrow = \rightsquigarrow_1 \cup \rightsquigarrow_2 \cup (E_1 \times \mathit{init}(\mathcal{E}_2)) \cup (\mathit{init}(\mathcal{E}_2) \times \mathit{exit}(\mathcal{E}_1))$.

The events of $\mathcal{E}_1 \,||_A\, \mathcal{E}_2$ are constructed in the following way: an event e of E_i (i=1, 2) that does not need to synchronise is paired with the auxiliary symbol $*$, and an event which is labelled with $\surd$ or with an action in A is paired with all events (if any) in the other tes that are equally labelled. Two events are put in conflict if any of their components are in conflict, or if different events have a common component different from $*$ (such events appear if two or more events in one tes synchronise with the same event in the other tes). A bundle is introduced iff when we take the projection on the component $\mathcal{E}_i$ of the bundle-set we obtain a bundle in $\mapsto_i$. Let for $A \subseteq \mathsf{Obs}$, $E_i^s \triangleq \{ e \in E_i \mid l_i(e) \in A \cup \{ \surd \} \}$ be the set of synchronising events and $E_i^f \triangleq E_i \setminus E_i^s$ the set of 'free' events.

Definition 9. Let $A \subseteq \mathsf{Obs}$. Then $\mathcal{E}_1 \,||_A\, \mathcal{E}_2 \triangleq (E, \rightsquigarrow, \mapsto, l, \mathcal{A}, \mathcal{R}, \mathcal{U})$ where

- $E = (E_1^f \times \{ * \}) \cup (\{ * \} \times E_2^f) \cup \{ (e_1, e_2) \in E_1^s \times E_2^s \mid l_1(e_1) = l_2(e_2) \}$
- $(e_1, e_2) \rightsquigarrow (e_1', e_2')$ iff
  - $(e_1 \rightsquigarrow_1 e_1') \vee (e_2 \rightsquigarrow_2 e_2')$ or
  - $(e_1 = e_1' \neq * \;\wedge\; e_2 \neq e_2') \vee (e_2 = e_2' \neq * \;\wedge\; e_1 \neq e_1')$
- $X \mapsto (e_1, e_2)$ iff
  - $(\exists X_1 : X_1 \mapsto_1 e_1 \;\wedge\; X = \{ (e, e') \in E \mid e \in X_1 \})$ or
  - $(\exists X_2 : X_2 \mapsto_2 e_2 \;\wedge\; X = \{ (e, e') \in E \mid e' \in X_2 \})$
- $l(e_1, e_2) = \mathsf{if}\ e_1 = *\ \mathsf{then}\ l_2(e_2)\ \mathsf{else}\ l_1(e_1)$
- $\mathcal{A}(e_1, e_2) = \mathcal{A}_1(e_1) \cap \mathcal{A}_2(e_2)$ with $\mathcal{A}_i(*) = [0, \infty)$.
- $\mathcal{R}(X, (e_1, e_2)) = \bigcap_{X_1 \in S_1} \mathcal{R}_1(X_1, e_1) \cap \bigcap_{X_2 \in S_2} \mathcal{R}_2(X_2, e_2)$ with
  - $S_1 = \{ X_1 \subseteq E_1 \mid X_1 \mapsto_1 e_1 \;\wedge\; X = \{ (e, e') \in E \mid e \in X_1 \} \}$ and
  - $S_2 = \{ X_2 \subseteq E_2 \mid X_2 \mapsto_2 e_2 \;\wedge\; X = \{ (e, e') \in E \mid e' \in X_2 \} \}$
- $(e_1, e_2) \in \mathcal{U}$ iff $e_1 \in \mathcal{U}_1 \vee e_2 \in \mathcal{U}_2$ with $* \notin \mathcal{U}_i$.

Parallel composition is illustrated by the following example where the left-hand tes is composed with the empty tes:

For $\mathcal{E}_1 \rhd_t \mathcal{E}_2$ a new urgent event e with delay $[t,t]$ is introduced that models the expiration of the timer. Since either the timer expires or $\mathcal{E}_1$ performs an initial event before (or at) t, e is put in mutual conflict with all initial events of $\mathcal{E}_1$.

Definition 10. For $t \in [0,\infty)$ let $\mathcal{E}_1 \rhd_t \mathcal{E}_2 \triangleq \mathcal{E}_1 + \hat{\tau}_{[t,t]} \,.\, \mathcal{E}_2$.

By straightforward proof one can establish that TES is closed under the operators $a_I \,.\,, +, \backslash A, [\lambda], \,;\,, [>, ||_A$, and $\rhd_t$.

5 A metric denotational semantics

The approach. We only give a brief account of our approach; see [2] for a full treatment, and [15, 5, 6] for more information on the use of metrics for denotational semantics. The semantic domain S for PA is equipped with a set Op' of operators that reflect the operators Op of Expr. For any fixed declaration $decl$, the function $P \mapsto \mathcal{M}(\langle decl, P\rangle)$ is a homomorphism from (Expr, Op) to (S, Op') such that the meaning of process variable x is given by $decl(x)$. Function $\mathcal{M}$ satisfies these conditions iff, for any fixed declaration $decl$, the function $P \mapsto \mathcal{M}(\langle decl, P\rangle)$ is a fixed point of $F_{decl} : [\mathsf{Expr} \longrightarrow S] \longrightarrow [\mathsf{Expr} \longrightarrow S]$, defined (in our case) by:

$$\begin{array}{rcll}
F_{decl}(\phi)(\mathbf{0}) & \triangleq & \mathbf{0}' & \\
F_{decl}(\phi)(\mathbf{1}) & \triangleq & \mathbf{1}' & \\
F_{decl}(\phi)(x) & \triangleq & \phi(decl(x)) & \\
F_{decl}(\phi)(op\ P) & \triangleq & op'\ F_{decl}(\phi)(P) & \text{for } op \in \{\, a_I \,.\,, \backslash A, [\lambda] \,\} \\
F_{decl}(\phi)(P\ op\ Q) & \triangleq & F_{decl}(\phi)(P)\ op'\ F_{decl}(\phi)(Q) & \text{for } op \in \{\, +, ||_A \,, \,;\,, [>, \rhd_t \,\}.
\end{array}$$

The semantics of PA is now obtained by $\mathcal{M}(\langle decl, P\rangle) = \phi_{decl}(P)$, where $\phi_{decl} : \mathsf{Expr} \longrightarrow S$ is the unique fixed point of F_{decl}. By Banach's fixpoint theorem, F_{decl} has a unique fixed point, provided that F_{decl} is contracting with respect to a distance function $\tilde{d}$ where $\langle [\mathsf{Expr} \longrightarrow S], \tilde{d}\rangle$ is a complete metric space (cms). $\tilde{d}$ is obtained from the cms $\langle S, d\rangle$ where

$$\tilde{d}(\phi_1, \phi_2) = sup\{\, d(\phi_1(P), \phi_2(P)) \mid P \in \mathsf{Expr} \,\}. \tag{1}$$

F_{decl} is contracting on $\langle [\mathsf{Expr} \longrightarrow S], \tilde{d}\rangle$ if its constituents $\,;\,, \rhd_t$ and so on, are non-distance increasing on $\langle S, d\rangle$ and contracting in certain arguments.

Time truncation. The basis of our distance function d is time truncation. The minimal time at which e can occur in $\mathcal{E}$ is defined by $mintime_{\mathcal{E}}(e) \triangleq inf\{\, t \in \mathbb{R}^+ \mid \exists\, \sigma \in Traces(\mathcal{E}) : (e,t) \in \sigma \,\}$, where $inf\, \varnothing \triangleq \infty$. For $t \in \mathbb{R}^+$ and $X \subseteq E$ let $X \restriction t \triangleq \{\, e \in X \mid mintime_{\mathcal{E}}(e) < t \,\}$ and $X \restriction \omega \triangleq \bigcup_{t \geqslant 0} X \restriction t$. Event e is called *executable* iff $e \in E \restriction \omega$, i.e. if $mintime_{\mathcal{E}}(e) < \infty$.

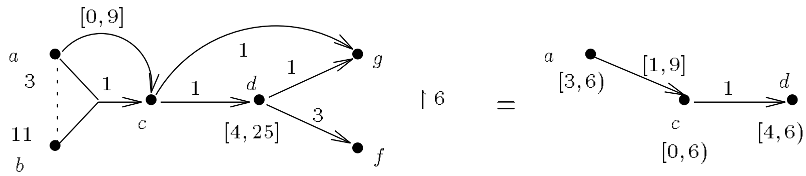
Definition 11. The *truncation* of $\mathcal{E}$ up to $t \in \mathbb{R}^+$ is defined by $\mathcal{E} \restriction t \triangleq (E \restriction t, \leadsto_t, \mapsto_t, l_t, \mathcal{A}_t, \mathcal{R}_t, \mathcal{U}_t)$ where $l_t(e) = l(e)$, $\mathcal{A}_t(e) = \mathcal{A}(e) \cap [0,t)$, $\mathcal{U}_t = \mathcal{U} \restriction t$, and

– $\leadsto_t = \leadsto \cap\, (E \restriction t \times E \restriction t)$

– $X \mapsto_t e$ iff there exists $Y \mapsto e$ with $Y \upharpoonright t = X$
– $\mathcal{R}_t(X, e) = \bigcap\{\, \mathcal{R}(Y, e) \mid Y \mapsto e, Y \upharpoonright t = X \,\}$.

It is not difficult to check that for $\mathcal{E} \in \mathsf{TES}$ we have $\mathcal{E} \upharpoonright t \in \mathsf{TES}$, for all $t \in \mathbb{R}^+$.

Example 1. Time truncation is illustrated by the following figure.



Events e_b, e_f and e_g are eliminated since the minimal time at which they can occur, time 11, 8 and 6, respectively, is at least 6. Note that $\{\, e_a \,\} \overset{[1,9]}{\mapsto}_t e_c$ for t=6, since $\{\, e_a, e_b \,\} \overset{1}{\mapsto} e_c$ and $\{\, e_a \,\} \overset{[0,9]}{\mapsto} e_c$ and $[1, \infty) \cap [0, 9] = [1, 9]$.

Lemma 12. *$Traces(\mathcal{E}) = \bigcup_{t \geqslant 0} Traces(\mathcal{E} \upharpoonright t)$.*

A complete ultra-metric space. The idea is to use time truncation as a basis for defining a distance d on TES. In particular, the distance between two tes's will be determined by the maximum amount of time they "agree", that is:

$$d(\mathcal{E}_1, \mathcal{E}_2) = inf\{\, 2^{-t} \mid \mathcal{E}_1 \upharpoonright t = \mathcal{E}_2 \upharpoonright t \,\}. \tag{2}$$

Remark that $\mathcal{E} \upharpoonright 0$ is the empty tes, so each pair of tes's agrees at least up to time 0. Although this basic idea is rather intuitive, it is, unfortunately, too naive. The problem is that some distinct tes's cannot be distinguished according to d. This means that d is a pseudo-metric rather than a metric. For instance, the tes consisting of a single event e with an empty bundle pointing to e is indistinguishable from the empty tes, since their time truncations are all empty. That is, according to (2) their distance is 0. The problem with this example is that tes's may contain events that can never appear. Such events can, for instance, appear in the semantics for expressions that contain circular causal dependencies, like in $a \,.\, b \,.\, \mathbf{0} \,||_{\{\, a,b \,\}}\, b \,.\, a \,.\, \mathbf{0}$, or timing constraints that avoid certain actions from happening, like in $a_2 \,.\, \mathbf{0} \rhd_1 b \,.\, \mathbf{0}$ where a will never happen. (Such events can be removed by applying the transformations exposed in [11, 9] that preserve timed event traces.)

A solution to this problem is to impose an equivalence relation, $\simeq$ say, on TES, while aiming at $d(\mathcal{E}_1, \mathcal{E}_2) = 0 \;\Leftrightarrow\; \mathcal{E}_1 \simeq \mathcal{E}_2$. Stated in other words, where $\mathbf{d}$ is the equivalent of d on $\mathsf{TES}/\!\simeq$ and $\mathbf{E}_i$ denotes the equivalence class of $\mathcal{E}_i$ under $\simeq$, we aim at $\mathbf{d}(\mathbf{E}_1, \mathbf{E}_2) = 0 \;\Leftrightarrow\; \mathbf{E}_1 = \mathbf{E}_2$. In order to obtain $\simeq$, the example suggests to abstract from events that can never be executed:

Definition 13. The *normal form* of $\mathcal{E}$, denoted $NF(\mathcal{E})$, is defined as $NF(\mathcal{E}) \triangleq (E \upharpoonright \omega, \rightsquigarrow_\omega, \mapsto_\omega, l_\omega, \mathcal{A}_\omega, \mathcal{R}_\omega, \mathcal{U}_\omega)$ where $l_\omega(e) = l(e)$, $\mathcal{A}_\omega(e) = \mathcal{A}(e)$, $\mathcal{U}_\omega = \mathcal{U} \cap (E \upharpoonright \omega)$, $\rightsquigarrow_\omega = \rightsquigarrow \cap\, (E \upharpoonright \omega \times E \upharpoonright \omega)$ and

– $X \mapsto_\omega e$ iff $X \restriction t \mapsto e$ is a bundle in $\mathcal{E} \restriction t$ for all $t > \mathit{mintime}_{\mathcal{E}}(e)$
– $\mathcal{R}_\omega(X, e) = \bigcup_{t > \mathit{mintime}_{\mathcal{E}}(e)} \{ \mathcal{R}(X \restriction t, e) \mid X \restriction t \mapsto e$ is a bundle of $\mathcal{E} \restriction t \}$.

For $\mathcal{E} \in \mathsf{TES}$ it follows by straightforward verification that $NF(\mathcal{E}) \in \mathsf{TES}$.

Lemma 14. *$Traces(NF(\mathcal{E})) = Traces(\mathcal{E})$.*

As in the untimed case [12] the metric approach also allows to abstract from the names of the events, i.e. to deal with isomorphism classes of tes's. The names of the events are only needed for technical reasons but they are meaningless for the semantics of a PA-process. The advantage of abstraction from event names is that the definitions of operators like $+$, $[>$, and so on, become less awkward.

Definition 15. $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i, \mathcal{A}_i, \mathcal{R}_i, \mathcal{U}_i)$ for i=1, 2 are *isomorphic* if there exists a bijection $f : E_1 \restriction \omega \longrightarrow E_2 \restriction \omega$ such that $l_2 \circ f = l_1$, $\mathcal{A}_2 \circ f = \mathcal{A}_1$ and

1. $e_1 \rightsquigarrow_1 e_2$ iff $f(e_1) \rightsquigarrow_2 f(e_2)$ for all $e_1, e_2 \in E_1 \restriction \omega$,
2. $X \overset{I}{\mapsto}_1 e$ iff $f(X) \overset{I}{\mapsto}_2 f(e)$ for all $e \in E_1 \restriction \omega, X \subseteq E_1 \restriction \omega$, and
3. $e \in \mathcal{U}_1 \restriction \omega$ iff $f(e) \in \mathcal{U}_2$.

Let $\mathcal{E}_1 \simeq_{iso} \mathcal{E}_2$ iff there exists an isomorphism from $\mathcal{E}_1$ to $\mathcal{E}_2$. Note that $\mathcal{E} \simeq_{iso} NF(\mathcal{E})$.

For $\mathcal{E} \in \mathsf{TES}$ let $\mathbf{E}_{\mathcal{E}}$ denote the equivalence class of $\mathcal{E}$ under $\simeq_{iso}$. For $\mathbf{E} \in \mathsf{TES}/{\simeq_{iso}}$ let $\mathbf{E} \restriction t \triangleq \mathbf{E}_{\mathcal{E} \restriction t}$, where $\mathcal{E}$ is a representative of $\mathbf{E}$. The distance between equivalence classes (under $\simeq_{iso}$) of tes's is given by:

$$\mathbf{d}(\mathbf{E}_1, \mathbf{E}_2) \triangleq \mathit{inf} \{ 2^{-t} \mid \mathbf{E}_1 \restriction t = \mathbf{E}_2 \restriction t \}. \tag{3}$$

Remark that $\mathbf{d}(\mathbf{E}, \mathbf{E} \restriction t) \leqslant 2^{-t}$ for all $t \geqslant 0$.

Example 2. Let $\mathcal{E}_i = (E_i, \varnothing, \mapsto_i, E_i \times \{ a \}, \mathcal{A}_i, \mathcal{R}_i, \varnothing)$, for i=1, 2 where

– $E_1 = \{ (k, j) \mid j \geqslant 1, 0 < k \leqslant j \}$ and $E_2 = E_1 \cup \{ (k, 0) \mid k \geqslant 1 \}$
– $\{ (k, j) \} \mapsto_i (k{+}1, j)$ for $0 < k < j$ and $\{ (k, 0) \} \mapsto_2 (k{+}1, 0)$ for $k \geqslant 1$
– $\mathcal{A}_i(k, j) = [k, k]$ for all $(k, j) \in E_i$, and
– $\mathcal{R}_i(\{ (k, j) \}, (k{+}1, j)) = [1, 1]$.

Then, $\mathcal{E}_1 \not\simeq_{iso} \mathcal{E}_2$ while $\mathcal{E}_1 \restriction t \simeq_{iso} \mathcal{E}_2 \restriction t$ for all $t \geqslant 0$. If we now would define $\mathbf{d}$ as suggested in (3) on $\mathsf{TES}/{\simeq_{iso}}$ then $\mathbf{d}(\mathbf{E}_1, \mathbf{E}_2) = 0$, although $\mathcal{E}_1$ and $\mathcal{E}_2$ are not isomorphic, thus yielding a pseudo-metric.

The problem with this example is that both tes's allow an infinite number of events to occur in a finite amount of time. This is avoided by considering finitely approximable tes's, a timed analogon of approximable event structures [12].

Definition 16. $\mathcal{E}$ is called *finitely approximable* iff $E \restriction t$ is finite for all $t \in \mathbb{R}^+$.

Let $\mathsf{TES}_{\mathit{fin}}/{\simeq_{iso}}$ denote the isomorphism classes of finitely approximable tes's. The main result that we need in order to define a metric semantics for PA is:

Theorem 17. *$\langle \mathsf{TES}_{fin}/{\simeq_{iso}}, \mathbf{d}\rangle$ is a complete ultra-metric space.*

A metric semantics for PA. We now give a metric semantics for (a subset of) PA based on equivalence classes (under $\simeq_{iso}$) of tes's. The main difference with the standard (untimed) case is the notion of 'guardedness' which ensures the well-definedness of recursive programs. While in the untimed case [12] guardedness requires that each process instantiation is preceded by an action we use a notion of *time guardedness* (like in timed CSP [17]) which requires that a recursive process instantiation can only happen after a positive amount of time. Let functions $\surd_{min} : \mathsf{Expr} \longrightarrow [0,\infty)$ and $tg : \mathsf{Expr} \longrightarrow [0,\infty)$ be defined as in Table 1. For declaration *decl* let $tg(decl) \triangleq inf\{\, tg(decl(x)) \mid x \in \mathsf{Var}\,\}$. *decl* is called *time-guarded* iff $tg(decl) > 0$. We give a metric semantics to TGPA,

P	$\surd_{min}(P)$	$tg(P)$
$\mathbf{0}$	∞	∞
$\mathbf{1}$	0	∞
x	0	0
$a_I \,.\, P$	$inf(I) + \surd_{min}(P)$	$inf(I) + tg(P)$
$P[\lambda], P \setminus A$	$\surd_{min}(P)$	$tg(P)$
$P + Q, P [> Q$	$min\{\, \surd_{min}(P), \surd_{min}(Q)\,\}$	$min\{\, tg(P), tg(Q)\,\}$
$P \,\vert\vert_A\, Q$	$max\{\, \surd_{min}(P), \surd_{min}(Q)\,\}$	$min\{\, tg(P), tg(Q)\,\}$
$P \,;\, Q$	$\surd_{min}(P) + \surd_{min}(Q)$	$min\{\, tg(P), \surd_{min}(P) + tg(Q)\,\}$
$P \rhd_t Q$	$min\{\, \surd_{min}(P), t + \surd_{min}(Q)\,\}$	$min\{\, tg(P), t + tg(Q)\,\}$

Table 1. Auxiliary functions $\surd_{min}$ and tg .

the set of *time-guarded processes*, i.e. the set of pairs $\langle decl, P\rangle$ where *decl* is a time-guarded declaration and P an expression. For the definition of the meaning function $\mathcal{M}_{cms} : \mathsf{TGPA} \longrightarrow \mathsf{TES}_{fin}/{\simeq_{iso}}$ we lift the semantic operators of Section 4 to operators on $\mathsf{TES}_{fin}/{\simeq_{iso}}$. Given that all operators defined in Section 4 preserve $\simeq_{iso}$ and finitely approximability (as can be shown by straightforward proof) we may define for $\mathbf{E}$, $\mathbf{F} \in \mathsf{TES}_{fin}/{\simeq_{iso}}$:

$$\begin{aligned} op\ \mathbf{E} &\triangleq \mathbf{E}_{op\ \mathcal{E}} \quad \text{for } op \in \{\, a_I\,.\,, \setminus A, [\lambda]\,\} \text{ and} \\ \mathbf{E}\ op\ \mathbf{F} &\triangleq \mathbf{E}_{\mathcal{E}\ op\ \mathcal{F}} \text{ for } op \in \{\, +, \,;\,, \,\vert\vert_A\,, [>, \rhd_t\,\} \end{aligned}$$

where $\mathcal{E}$, $\mathcal{F}$ are representatives of $\mathbf{E}$ and $\mathbf{F}$, respectively. Let $\mathbf{E_0}$ be the equivalence class of the empty tes and $\mathbf{E_1}$ the equivalence class of the tes

$$\mathcal{E}_{\mathbf{1}} \triangleq (\{\, e\,\}, \varnothing, \varnothing, \{\, (e, \surd)\,\}, \{\, (e, [0,\infty))\,\}, \varnothing, \varnothing).$$

Together with these semantic operators, $\mathsf{TES}_{fin}/{\simeq_{iso}}$ constitutes a PA-algebra.

Lemma 18. *For $\mathbf{E}, \mathbf{E}', \mathbf{F}, \mathbf{F}' \in \mathsf{TES}_{fin}/{\simeq_{iso}}$ we have*

1. $\mathbf{d}(a_I \,.\, \mathbf{E}, a_I \,.\, \mathbf{E}') = 2^{-inf(I)} \cdot \mathbf{d}(\mathbf{E}, \mathbf{E}')$

2. $\mathbf{d}(\mathbf{E}\, op\, \mathbf{F}, \mathbf{E}'\, op\, \mathbf{F}') \leqslant max\,\{\, \mathbf{d}(\mathbf{E}, \mathbf{E}'), \mathbf{d}(\mathbf{F}, \mathbf{F}')\,\}$ *for* $op \in \{\, +, ||_A\,, [>\,\}$
3. $\mathbf{d}(op\, \mathbf{E}, op\, \mathbf{E}') \leqslant \mathbf{d}(\mathbf{E}, \mathbf{E}')$ *for* $op \in \{\, \backslash A, [\lambda]\,\}$
4. $\mathbf{d}(\mathbf{E}\,;\,\mathbf{F}, \mathbf{E}'\,;\,\mathbf{F}') \leqslant max\,\{\, \mathbf{d}(\mathbf{E}, \mathbf{E}'), 2^{-\surd_{min}(\mathbf{E})} \cdot \mathbf{d}(\mathbf{F}, \mathbf{F}')\,\}$
5. $\mathbf{d}(\mathbf{E} \rhd_t \mathbf{F}, \mathbf{E}' \rhd_t \mathbf{F}') \leqslant max\,\{\, \mathbf{d}(\mathbf{E}, \mathbf{E}'), 2^{-t} \cdot \mathbf{d}(\mathbf{F}, \mathbf{F}')\,\}$.

where $\surd_{min}(\mathbf{E}_{\mathcal{E}}) \triangleq inf\,\{\, mintime_{\mathcal{E}}(e) \mid e \in E, l(e) = \surd \}$.

Lemma 19. *For each decl and homomorphisms* ϕ_1, $\phi_2 : \mathsf{Expr} \longrightarrow \mathsf{TES}_{\mathit{fin}}/\simeq_{iso}$*:*

$$\tilde{d}(F_{decl}(\phi_1), F_{decl}(\phi_2)) \;\leqslant\; 2^{-tg(decl)} \cdot \tilde{d}(\phi_1, \phi_2).$$

where $\tilde{d}$ *is defined in equation (1).*

This result says that F_{decl} is contracting with contraction coefficient $2^{-tg(decl)}$ provided that *decl* is time-guarded. Thus, for time-guarded declaration *decl*, F_{decl} has a unique fixed point, say ϕ_{decl}. The metric semantics $\mathcal{M}_{cms} : \mathsf{TGPA} \longrightarrow \mathsf{TES}_{\mathit{fin}}/\simeq_{iso}$ is now defined by $\mathcal{M}_{cms}(\langle decl, P\rangle) \triangleq \phi_{decl}(P)$.

6 Concluding remarks

Relation with untimed case. This paper defines a metric semantics $\mathcal{M}_{cms}$ for an expressive real-time process algebra PA that contains delay and timeout operators. The distance d measures the amount of time in which two processes coincide, i.e., $d(\mathcal{M}_{cms}(P_1), \mathcal{M}_{cms}(P_2)) \leqslant 2^{-t}$ iff P_1 and P_2 have the same behaviour upto time t. This notion of distance is a timed analogon of the distance proposed in [12] which is based on the number of steps processes coincide.

Consistency. [10] defines a cpo-based denotational semantics $\mathcal{M}_{cpo}$ and an (event-based) operational semantics for PA such that $\mathit{Traces} \circ \mathcal{M}_{cpo}$ are precisely the timed event traces that are generated operationally. The formal relationship between our cpo and metric semantics is as follows. Let $\mathsf{TES}_{\mathit{fin}}$ be the set of timed event structures that are finitely approximable. For time-guarded $\langle decl, P\rangle$ it follows that $\mathcal{M}_{cpo}(\langle decl, P\rangle)$ is finitely approximable. Function $f : \mathsf{TES}_{\mathit{fin}} \longrightarrow \mathsf{TES}_{\mathit{fin}}/\simeq_{iso}$ with $f(\mathcal{E}) \triangleq \mathbf{E}_{\mathcal{E}}$ is a homomorphism between the PA-algebras $\mathsf{TES}_{\mathit{fin}}$ and $\mathsf{TES}_{\mathit{fin}}/\simeq_{iso}$. Then, according to the results of [3], we obtain for any time-guarded process $\langle decl, P\rangle$: $f\,(\mathcal{M}_{cpo}(\langle decl, P\rangle)) \;=\; \mathcal{M}_{cms}(\langle decl, P\rangle)$. This entails that the presented metric semantics is significantly more abstract than the cpo-based, and consequently, the operational semantics of TGPA.

Related work. Several real-time extensions of process algebras have been proposed in the literature; for an overview see [14]. Usually, timed process algebras are provided with an operational semantics in the style of Plotkin that is based on some notion of (timed) transition system. Notably exceptions are the works on timed CSP by Reed & Roscoe [17] who use a metric denotational semantics based on timed refusals, and real-time LOTOS by Bryans, Davies & Schneider [8] who use a (non-standard) fixed point semantics based on an advanced form of timed refusals in order to deal with divergence. Both works provide an interleaving semantics. In the non-interleaving setting, related work has been done by

Murphy [13] on interval event structures in which events have a duration. Murphy uses time truncation—in a similar way as we do in the metric semantics—as a basis for obtaining limiting infinite objects using ideal completions.

References

1. A.F. Ates, M. Bilgic, S. Saito and B. Sarikaya. Using timed CSP for specification verification and analysis of multi-media synchronization. *IEEE J. on Sel. Areas in Comm.*, **14**(1):126–137, 1996.
2. C. Baier and M.E. Majster-Cederbaum. Denotational semantics in the cpo and metric approach. *Th. Comp. Sci.*, **135**:171–220, 1994.
3. C. Baier and M.E. Majster-Cederbaum. How to interpret consistency and establish consistency results for semantics of concurrent programming languages. *Fund. Inf.*, **29**:225–256, 1997.
4. C. Baier and M.E. Majster-Cederbaum. Metric semantics from partial order semantics. *Acta Inf.*, **34**:701–735, 1997.
5. J.W. de Bakker and J.I. Zucker. Processes and the denotational semantics of concurrency. *Inf. and Contr.*, **54**(1/2):70–120, 1982.
6. J.W. de Bakker and E.P. de Vink. *Control Flow Semantics*. MIT Press, 1996.
7. T. Bolognesi and E. Brinksma. Introduction to the ISO specification language LOTOS. *Comp. Netw. & ISDN Syst.*, **14**:25–59, 1987.
8. J. Davies, J.W. Bryans and S.A. Schneider. Real-time LOTOS and timed observations. In *Formal Description Techniques VIII*. Chapmann & Hall, 1995.
9. J-P. Katoen. *Quantitative and Qualitative Extensions of Event Structures*. PhD thesis, University of Twente, 1996.
10. J-P. Katoen, D. Latella, R. Langerak and E. Brinksma. On specifying real-time systems in a causality-based setting. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*, LNCS 1135, pages 385–405. Springer-Verlag, 1996.
11. R. Langerak. Bundle event structures: a non-interleaving semantics for LOTOS. In *Formal Description Techniques V*, pages 331–346. North-Holland, 1993.
12. R. Loogen and U. Goltz. Modelling nondeterministic concurrent processes with event structures. *Fund. Inf.*, **14**(1):39–74, 1991.
13. D. Murphy. Time and duration in noninterleaving concurrency. *Fund. Inf.*, **19**:403–416, 1993.
14. X. Nicollin and J. Sifakis. An overview and synthesis on timed process algebras. In *Real-Time: Theory in Practice*, LNCS 600, pages 526–548. Springer-Verlag, 1992.
15. M. Nivat. Infinite words, infinite trees, infinite computations. In *Foundations of Computer Science III*, Mathematical Centre Tracts **109**, pages 3-52, 1979.
16. M. Nielsen, G.D. Plotkin and G. Winskel. Petri nets, event structures and domains, part 1. *Th. Comp. Sc.*, **13**(1):85–108, 1981.
17. G.M. Reed and A.W. Roscoe. A timed model for Communicating Sequential Processes. *Th. Comp. Sc.*, **58**:249–261, 1988.
18. J.J. Žic. Time-constrained buffer specifications in CSP+T and timed CSP. *ACM Transactions on Programming Languages and Systems*, 16(6):1661–1674, 1994.

This article was processed using the LaTeX macro package with LLNCS style