Metric Semantics for True Concurrent Real Time

Joost-Pieter Katoen [a,1] Christel Baier [b] Diego Latella [c]

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

[b] *Fakultät für Mathematik und Informatik Universität Mannheim, 68131 Mannheim, Germany*

[c] *CNUCE Istituto del CNR, Via Santa Maria 36, 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 of the metric 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 identities and non-executable events (events that can never occur) is shown to be a complete ultra-metric space. We present an operational semantics for the considered language and show that the metric semantics is an abstraction of it. The operational semantics is characterised by the absence of synchronisation on the advance of time as opposed to the operational semantics of most real-time calculi. The consistency between our metric and an existing cpo-based denotational semantics is briefly investigated.

Key words: consistency of semantics; denotational semantics; (bundle) event structure; interleaving; metric space; process algebra; real-time; semantics; true concurrency

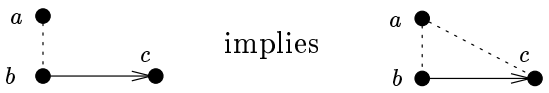
[1] Currently at the Formal Methods and Tools Group, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

1 Introduction

In this paper we consider a metric denotational semantics for an algebraic specification language that besides concurrency, synchronisation, and nondeterminism, encompasses the notion of real-time. This study is carried out in a branching-time non-interleaving context, using the model of event structures. These structures typically consist of a set of labelled events, a causality relation (denoted $\mapsto$) between events, and a conflict relation (denoted $\#$) between events. An event models the occurrence of the action as indicated by its label. The causality relation models a 'happens before' relation in the following sense: $e \mapsto e'$ implies that if event e' happens then event e must have happened before. The conflict relation models a choice: if $e \# e'$ then either event e or event e' can happen, but they cannot occur both. Usually the event identities are not of importance and isomorphism classes of event structures are considered. If no confusion arises, action labels $(a, b, \ldots)$ are used instead of event identities $(e, e', \ldots)$.

1.1 Prime event structures and TCSP

For the untimed case, Loogen and Goltz [29] propose a metric denotational semantics for theoretical CSP using prime event structures, the most elementary form of event structures. In prime event structures [34], the causality relation $\mapsto$ is a partial-order and the conflict relation $\#$ is irreflexive and symmetric. Conflicts are inherited as follows: if $e_1 \# e_2$ and $e_1 \mapsto e_3$ then $e_2 \# e_3$. Pictorially:



where dots represent events, directed arrows model $\mapsto$ and dotted lines model $\#$. The interpretation of prime event structures is defined in terms of sets of configurations, conflict-free sets of events that are downwards closed under $\mapsto$, ordered under set inclusion. For instance, the maximal configurations of the prime event structure above are $\{a\}$ and $\{b, c\}$. To assign a meaning to recursive TCSP specifications, Loogen and Goltz apply a metric approach to (isomorphism classes of) so-called finitely approximable prime event structures. In a nutshell, in such structures the depth of each event — the length of the longest causal chain pointing to that event — is finite, and for each finite depth, there is only a finite number of events of that depth. The notion of distance between prime event structures $\mathcal{E}_1$ and $\mathcal{E}_2$ is based on truncation:

$$d(\mathcal{E}_1, \mathcal{E}_2) =_{df} inf\{\, 2^{-n} \mid \mathcal{E}_1 \upharpoonright n = \mathcal{E}_2 \upharpoonright n \,\}$$

where $\mathcal{E} \upharpoonright n$ denotes the restriction of $\mathcal{E}$ to all events with depth at most n. The set of finitely approximable prime event structures with distance d constitutes a complete ultra-metric space, and the operators of TCSP are non-expansive with respect to d. For example, for prefixing and parallel composition this is guaranteed by the following inequalities:

$$d(a\,.\,\mathcal{E}, a\,.\,\mathcal{E}') \leqslant 2^{-1} \cdot d(\mathcal{E}, \mathcal{E}')$$
$$d(\mathcal{E} \,||_A\, \mathcal{F}, \mathcal{E}' \,||_A\, \mathcal{F}') \leqslant max\{\, d(\mathcal{E}, \mathcal{E}'), d(\mathcal{F}, \mathcal{F}') \,\}.$$

The semantics for TCSP-expression P and any fixed declaration *decl* of processes can then be considered as the unique fixed point of a higher-order function F_{decl} over the domain of functions from TCSP-expressions (Expr) to (isomorphism classes of) finitely approximable prime event structures ($\mathsf{PES}_{fin}/\simeq_{iso}$). The distance d is lifted to this function domain in the following standard way [12]:

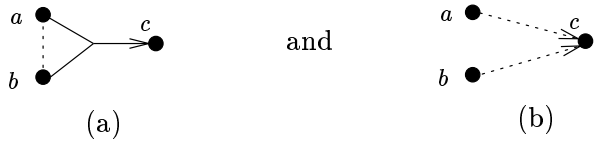
$$\tilde{d}(\phi_1, \phi_2) =_{df} sup\{\, d(\phi_1(P), \phi_2(P)) \mid P \in \mathsf{Expr} \,\}$$

for $\phi_1, \phi_2 : \mathsf{Expr} \longrightarrow \mathsf{PES}_{fin}/\simeq_{iso}$. For each guarded declaration *decl*, function F_{decl} is contracting with respect to distance $\tilde{d}$. Due to Banach's theorem, the contractiveness of F_{decl} guarantees that a fixed point of F_{decl} exists and that it is unique. Declaration *decl* is guarded if any process instantiation is preceded by a prefix for each process definition in *decl*. As a final result, Loogen and Goltz show that for finite processes the metric semantics is weakly bisimilar to the interleaving semantics of TCSP; a result that later has been extended to recursive processes [7].

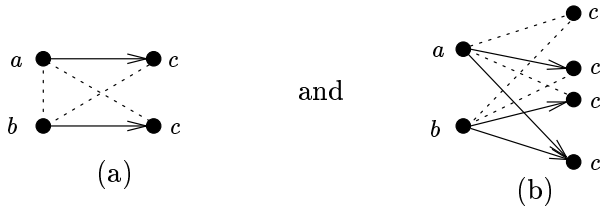
1.2 Bundle event structures and LOTOS

In this paper we consider a real-time extension of a process algebra based on the internationally standardised specification language LOTOS [13] (Language of Temporal Ordering Specification). As semantic domain we take a timed extension of Langerak's *bundle event structures* [26,27], 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. Bundle event structures are strictly more expressive than prime event structures, i.e. there do exist bundle event structures for which there does not exist a prime event structure with the same set of configurations (and not the reverse). A comparison of the expressive power of bundle event structures compared to Winskel's stable [40] and Boudol and Castellani's flow event structures [14] is given in [26].

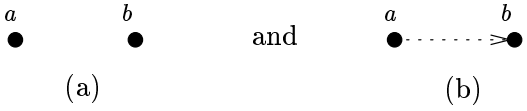
In bundle event structures, $\mapsto$ is a relation between a set of events that are in mutual conflict and an event. The conflict relation is irreflexive, but not required to be symmetric. It is denoted by $\rightsquigarrow$ and depicted by a dotted arrow. In case $e \rightsquigarrow e'$ and $e' \rightsquigarrow e$ we use a dotted line. Intuitively,



denote that (a) event c can happen if either a or b has happened before, respectively that (b) event c disables the occurrence of a and b, i.e. neither a nor b can happen after c happened (notice that c can happen after a, or b, or both a and b instead). Due to the inheritance of conflicts, corresponding prime event structures would lead to copying of events:

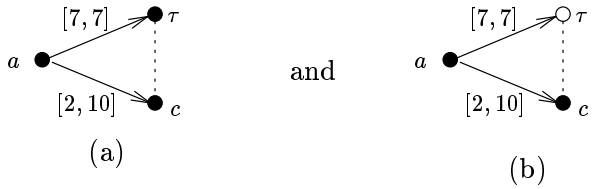


This property makes prime event structures less attractive as a semantical model for a process algebra like LOTOS. Due to the increased expressive power of bundle event structures, an interpretation in terms of configurations ordered under set inclusion is insufficient. For instance, the bundle event structures



have both as maximal configuration $\{a, b\}$, whereas b can happen after the occurrence of a in the left (a), but not in the right (b) structure. Instead, the interpretation is defined in terms of labelled partial-orders ordered under prefixing [38], or equivalently, in terms of event traces. The maximal event traces of the structures above are (a) $a\,b$ and $b\,a$, and (b) $a\,b$ and b. Langerak uses bundle event structures to give a non-interleaving semantics to LOTOS [26,27] and although he provides a meaning to recursive processes using a partial-order approach, it seems that (a slight modification of) the more abstract metric approach of Loogen and Goltz can be used equally well.

In the timed extension of bundle event structures of Katoen et al. [23] the basic idea is to associate relative delays to causality relations (the bundles) and to impose urgency on certain events (open dots). From now on, we refer to this extension as timed event structures. The suitability of this timed truly concurrent model for modelling time-critical systems is addressed in [23] and is not further discussed here. The timed event structures



both denote that after the occurrence of event a, either event τ happens after 7 time units, or that c happens after time t with $2 \leqslant t \leqslant 10$. In structure (b) event τ is urgent, i.e. it must happen 7 time units since the occurrence of a if c did not yet occur, so preventing c from happening thereafter. The interpretation of timed event structures is defined in terms of timed event traces. Example maximal traces of the timed structures above are (a) $(a,t_a)\,(\tau,t_\tau)$ with $t_\tau = t_a{+}7$ and $(a,t_a)\,(c,t_c)$ with $2 \leqslant t_c{-}t_a \leqslant 10$ and (b) $(a,t_a)\,(\tau,t_\tau)$ with $t_\tau = t_a{+}7$ and $(a,t_a)\,(c,t_c)$ with $2 \leqslant t_c{-}t_a \leqslant 7$.

Timed event structures are used as a non-interleaving semantical model for a real-time process algebra where prefixing $a\,.\,P$ is replaced by timed prefixing $a_I\,.\,P$ where I denotes a set of time instants. Moreover, a timeout operator $P \rhd_t Q$ is included that behaves initially like P, but in which control is passed to Q if P does not perform an action [2] before time t.

In order to assign a meaning to recursive specifications we follow a similar approach as Loogen and Goltz. The basic idea of our metric semantics is to consider behaviours of timed event structures up to a certain time. That is, the distance function is based on the amount of time to which timed event structures do 'agree':

$$d(\mathcal{E}_1,\mathcal{E}_2) =_{df} inf\{\, 2^{-t} \mid \mathcal{E}_1 \restriction t = \mathcal{E}_2 \restriction t \,\}$$

where $\mathcal{E} \restriction t$ denotes the restriction of $\mathcal{E}$ to all events that can occur before time t. 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

[2] Opposed to timed CSP [37] we do not distinguish between the occurrence of internal and external actions in P.

an isomorphism $\simeq_{iso}$ that abstracts from event identities (as usual) and from non-executable events, events that can never appear. [3] Secondly, we refine this notion towards finitely approximable timed event structures modulo $\simeq_{iso}$ and show that this quotient model is a complete ultra-metric space. A timed event structure is called finitely approximable if the number of events that can occur before time t is finite, for any t. We show that the operators of our real-time process calculus are non-expansive with respect to our notion of distance, for instance, timed prefixing is contractive and timeout is non-expansive:

$$\begin{aligned} d(a_I \,.\, \mathcal{E}, a_I \,.\, \mathcal{E}') &\leqslant 2^{-inf(I)} \cdot d(\mathcal{E}, \mathcal{E}') \\ d(\mathcal{E} \rhd_t \mathcal{F}, \mathcal{E}' \rhd_t \mathcal{F}') &\leqslant max\{\, d(\mathcal{E}, \mathcal{E}'), 2^{-t} \cdot d(\mathcal{F}, \mathcal{F}') \,\}. \end{aligned}$$

Similarly as we have discussed for the case for prime event structures, the semantics is now defined as the unique fixed point of a higher-order function F_{decl}. As a main result we obtain for any expression with fixed declaration *decl* that

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

where $tg(decl)$, the time-guard of *decl*, is the minimal time between successive process instantiations in any process definition in *decl* and $\phi_1, \phi_2 : \mathsf{Expr} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$. Thus, for time-guarded processes — processes that cannot generate instantaneous recursive process instantiations — the function F_{decl} has a unique fixed point.

Finally, we present a structured operational semantics for the considered language (recalled from Katoen et al. [23]) and show that this semantics is strongly timed bisimilar to an interleaving perspective of our metric true concurrent semantics. The operational semantics is characterised by the absence of synchronisation on the advance of time as opposed to the operational semantics of most real-time process calculi [33]. The traces generated from our operational semantics can be considered as equivalence classes (under reordering of causally independent events) whereas more standard operational semantics for real-time calculi lead to the time-consistent representatives of each equivalence class, and this is less abstract. We also briefly show that the metric semantics presented in this paper is an abstraction of the cpo-based semantics of Katoen et al. [23].

[3] Non-executable events do not appear in the untimed setting with prime event structures.

Several real-time extensions of process algebras have been proposed in the literature; for an overview see [33]. Usually, timed process algebras are provided with an operational interleaving semantics in the style of Plotkin that is based on some notion of timed transition system. Notable exceptions are the works on timed CSP by Reed & Roscoe [37] who define a metric denotational semantics for time-guarded processes based on timed refusals, and, more recently, on real-time LOTOS by Bryans, Davies & Schneider [17] who use a (non-standard) fixed point semantics based on an advanced form of timed refusals in order to deal with divergence. Both works consider an interleaving semantics.

Timed extensions of partial-order models have received scant attention in the literature. For example, extensions of configurations [30], prime event structures [32], posets [21], and higher-dimensional automata [20] do exist, but these models have not been used as a semantic model for a timed process algebra and are merely of theoretical interest. Murphy [32] uses time truncation — in a similar way as we do — as a basis for obtaining limiting infinite objects using ideal completions. Our approach resembles that of Fidge [18]. Fidge proposes a real-time extension of causal trees, equivalence classes of event structures under history-preserving bisimulation, and uses this model to provide a semantics to a timed variant of CCS. This approach has later been extended to include time markers that facilitate the specification of relative time delays between arbitrary actions [19]. Katoen et al. [24] consider a timed variant of bundle event structures (like in this paper), to provide a semantics for a real-time variant of LOTOS, in which a powerful urgency-operator is incorporated. This approach requires a time-consistent setting (unlike this paper), and uses a partial-order approach towards recursive behaviours,

To the best of our knowledge, there are no other approaches that consider real-time true concurrency in a metric setting.

1.5 Organisation of the paper

The organisation of the paper is as follows. Section 2 introduces the real-time process algebra. Section 3 describes timed event structures and Section 4 presents the semantical operators on these structures. The metric semantics is developed in Section 5 which is the core part of the paper. Section 6 presents the operational interleaving semantics and investigates its consistency with the metric semantics. Concluding remarks are provided in Section 7.

A preliminary short version of this paper has been published as [5]; some other

parts were contained in the dissertation [22].

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 $\surd \neq \tau$. Let $\mathbb{R}^+$ denote the set of non-negative reals. 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 $\lambda(\tau) = \tau$, $\lambda(\surd) = \surd$ and $\lambda(a) \neq \surd$ for $a \in \mathsf{Obs}$, and Var a set of process variables with $x \in \mathsf{Var}$. The set of expressions Expr is defined as follows:

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

The operators $+$, $\setminus A$, 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 any time $t \in I$. For $I = [0, \infty)$ the usual untimed prefix is obtained.
- $P \,;\, Q$ 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 the successful termination action) 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 in the literature have proven that the timed operators like $a_I \,.\, P$ and $P \rhd_t Q$ are convenient to specify practical real-time systems [4,41]. This shows the adequacy of the considered timed process algebra.

Process variables are considered in the context of a set of process definitions of the form $x := P$. Note that 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 $P \in \mathsf{Expr}$. Let PA denote the set of all processes.

In order to avoid brackets we introduce the following precedence order of the composition operators, listed in decreasing binding order: a_I . , $+$, $||_A$, $[>$, ; , $\rhd_t$, $\backslash A$ and $[\lambda]$.

3 Timed event structures

3.1 The model

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 [26,27] and equip them 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. These two ingredients make bundle event structures suitable for providing a non-interleaving semantics to LOTOS [26,27].

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. As explained before, 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. Empty bundles are allowed; $\varnothing \mapsto e$ models that e can never happen [4] . The reason for not having a binary causality relation between events (as in prime event structures [34]) is to model parallel composition $||_A$ in a less complex way.

Time is added to event structures in the following way [23]. Relative delays

[4] Events that are pointed to by empty bundles are comparable to self-conflicting events in flow event structures [14], but — as opposed to self-conflicting events — they have the pleasant property that they can always be eliminated using transformations [26,27]. The same applies to bundles like $X \mapsto e$ with $e \in X$.

between events are attached to bundles, and delays relative to the start of the system are attached to events. The latter delays can be considered as absolute delays. 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 *(Timed event structure). 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*
- $\mathcal{U} \subseteq \{ e \in E \mid l(e) = \tau \}$, *the set of* urgent *events,*

such that $l, \mathcal{A}$ *and* $\mathcal{R}$ *are total functions and for any bundle-set* X*:*

(P1) $(X \times X) \setminus Id_E \subseteq \rightsquigarrow$

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

(P2) *for all* $e' \in E$ *and bundle-set* X

$$((e' \rightsquigarrow e \ \vee \ e \rightsquigarrow e') \ \wedge \ X \mapsto e) \ \Rightarrow \ (X \mapsto e' \ \vee \ X \rightsquigarrow e')$$

(P3) *there exists a time point* $t \in \mathbb{R}^+$ *such that*

$$(\mathcal{A}(e) \in \{\varnothing, \{t\}\}) \ \vee \ (\exists X : X \mapsto e \ \wedge \ \mathcal{R}(X, e) \in \{\varnothing, \{t\}\}).$$

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'.

If no confusion arises, timed event structures will be called simply event structures throughout this paper. Event structures are depicted as follows. Events are denoted as dots; near the dot the action label is given. $e \rightsquigarrow e'$ 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 $X \stackrel{I}{\mapsto} e$. Delays $[t, \infty)$ are simply denoted by t; delays $[0, \infty)$ are usually omitted.

Example 2 *Figure 1(a) shows an example event structure with e.g. timed bundles* $\{a\} \stackrel{[0,7]}{\mapsto} b$ *and* $\{a\} \stackrel{[0,5]}{\mapsto} c$*, and conflicts* $b \rightsquigarrow \tau$ *and* $\tau \rightsquigarrow b$*. The set of urgent events* $\mathcal{U} = \{\tau\}$ *and the event delay* $\mathcal{A}$ *is 0 for all events.*

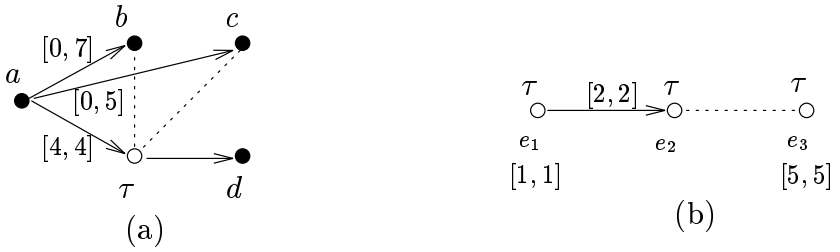
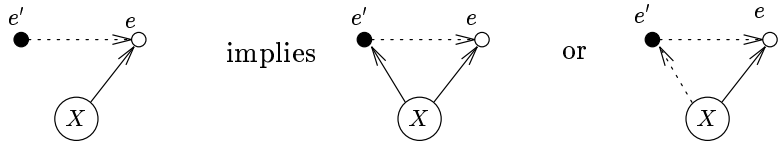


Fig. 1. (a) An event structure and (b) a structure that violates (P2)

The constraints (P1) through (P3) are justified in the following.

- Constraint (P1) requires all events in bundle set X to be in mutual conflict. This enables us to uniquely define a causal ordering between the events in a system run: if some event, e say, occurs in a system run, then it is for each bundle $X \mapsto e$ uniquely determined which event in X has caused e. If constraint (P1) is omitted, several interpretations turn out to be plausible with different characteristics [28]. The constraint is similar to the stability constraint in stable event structures [40].
- Constraint (P2) enforces that as soon as e is enabled either e' is also enabled (provided e' is not disabled in some way), or as soon as e' occurs e will be permanently disabled, since some bundle pointing to e is disabled by e'. Pictorially for the case $e' \rightsquigarrow e$:



  The justification for this constraint is to be able to "locally" decide whether an event can occur by only considering its direct causal predecessors and conflicts. This enables a more straightforward notion of timed event trace (see further on) and does not impose any restriction on the usage of the model as semantics for our language. It forbids structures like Figure 1(b), where event e_3 cannot occur, since the urgent event e_1 — which is neither in a direct causal nor conflict relation with e_3 — is forced to occur at time 1 and subsequently the urgent event e_2 must occur at time 3. That is to say, in order to decide whether event e_3 can occur initially, we have to consider the event e_1 which is not in a direct relation to e_3. For the sake of convenience we like to avoid these situations. As we will see, such structures cannot be described by the real-time process algebra of Section 2.
- Constraint (P3) ensures that urgent events are enabled at a single time instant only, if ever. The motivation for this constraint is that urgent events are used for the sole purpose of modelling timeouts which are internal actions of a process and typically can appear at a single time instant only.

The concept of a system run for tes's is captured by the notion of a *timed event trace.*

Definition 3 *(Enabled events after σ). For σ a sequence of distinct events let the set of events enabled in $\mathcal{E}$ after σ be defined as*[5]

$$\mathsf{en}^{\mathcal{E}}(\sigma) =_{df} \{\, e \in E \setminus \sigma \mid (\forall\, e_i \in \sigma : e \not\leadsto e_i) \ \wedge\ (\forall\, X \mapsto e : X \cap \sigma \neq \varnothing) \,\}.$$

Stated in words, an event is enabled after σ if it is not disabled by one of the events in σ, and if for any bundle pointing to it some event appears in σ.

For events that have more than one bundle pointing to them we take the following interpretation. Consider $\{\, a \,\} \stackrel{I}{\mapsto} c$ and $\{\, b \,\} \stackrel{J}{\mapsto} c$. If a happens at time t_a and b at time t_b, then c is enabled at any $t \in (t_a{+}I) \cap (t_b{+}J)$ where for $t \in \mathbb{R}$ and $I \subseteq \mathbb{R}$, $t{+}I$ denotes $\{\, t{+}t' \mid t' \in I \,\}$. When the intersection of two (or more) sets of time instants is empty this means that (due to incompatible time constraints) the event at hand cannot occur at any time and will be permanently disabled.

Let $\mathsf{tm}^{\mathcal{E}}_{\sigma}(e)$ denote the set of time instants at which an enabled event e after σ could happen, given that each event e_i in σ occurred at time t_i. Event e can occur if (i) its absolute delay $\mathcal{A}(e)$ is respected, (ii) for each event e_i with $e_i \rightsquigarrow e$ we have that e occurs at at least t_i, and (iii) the time relative to all its immediate causal predecessors is respected. Cases (ii) and (iii) take care of the fact that events cannot occur before their causes, entailing that causal ordering implies temporal ordering. So, we obtain

Definition 4 *(Potential time of occurrence). For $\sigma = (e_1, t_1) \ldots (e_n, t_n)$ a timed sequence of distinct events and event $e \in \mathsf{en}^{\mathcal{E}}(\sigma)$ let*

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

It is easy to check that for any urgent event e we have $\mathsf{tm}^{\mathcal{E}}_{\sigma}(e) = \varnothing$ or $\mathsf{tm}^{\mathcal{E}}_{\sigma}(e) = \{\, t \,\}$ for some $t \in \mathbb{R}^+$, due to constraint (P3). In the latter case we often identify $\mathsf{tm}^{\mathcal{E}}_{\sigma}(e)$ with t. Let σ_i denote the i-th prefix of σ, that is, $\sigma_i = (e_1, t_1) \ldots (e_i, t_i)$.

Definition 5 *(Timed event trace). Sequence $\sigma = (e_1, t_1) \ldots (e_n, t_n)$ with $e_i \in E$ (all events being pairwise distinct) and $t_i \in \mathbb{R}^+$, is a* timed event trace *of*

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

$\mathcal{E} \in \mathsf{TES}$ *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) $(e_i \rightsquigarrow e \;\vee\; e \rightsquigarrow e_i) \;\Rightarrow\; t_i \leqslant \mathsf{tm}^{\mathcal{E}}_{\sigma_{i-1}}(e)$ *for all* $e \in \mathcal{U} \cap \mathsf{en}^{\mathcal{E}}(\sigma_{i-1})$*.* [6]

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

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.

Example 6 *For the following timed sequences of events the conditions are given under which they are timed event traces of Figure 1(a):*

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

Note that Figure 1(a) models a typical timeout scenario: if after the occurrence of event a neither b nor c happen within 4 time units, then a timeout (event τ*) is forced to occur. It* τ *would not be urgent, the upper bound 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 would not be forced to occur and time does not resolve the choice.*

Timed event traces do respect causality, but not necessarily the advance of time. That is, two (or more) independent events can occur in a trace in either order regardless of their timing. For example, $(a, 1)(b, 3)(c, 4)$ and $(a, 1)(c, 4)(b, 3)$ are timed event traces of Figure 1(a). The choices correspond to the possible interleavings of the causally independent events. This situation is similar to the untimed case, where in a true concurrent setting, causally independent events can occur in either order when considering event traces, linearisations of partial orders. Since the causal ordering between events implies their temporal ordering, the causal ordering can never contradict the temporal order. Such traces are being referred to as "ill-timed but well-caused" [2].

The following result implies that for any ill-timed event trace σ there exists a corresponding time-consistent event trace σ', that can be obtained from σ by swapping ill-timed pairs of timed events repeatedly.

[6] Here we use $\leqslant$ on sets (singletons or empty sets). By convention we use $t \leqslant \varnothing$.

Theorem 7 *For all $t' < t$ and timed sequences of distinct events σ, σ':*

$$\sigma\,(e,t)(e',t')\,\sigma' \in \mathit{Traces}(\mathcal{E}) \;\Rightarrow\; \sigma\,(e',t')(e,t)\,\sigma' \in \mathit{Traces}(\mathcal{E})$$

PROOF. Let $\sigma_1 = \sigma\,(e,t)(e',t')\,\sigma' \in \mathit{Traces}(\mathcal{E})$ and assume $t' < t$. We prove the theorem by contradiction. Suppose $\sigma_2 = \sigma\,(e',t')(e,t)\,\sigma' \notin \mathit{Traces}(\mathcal{E})$. This can only be because one of the following reasons:

(1) $e_j \rightsquigarrow e_i$ and (i) $j \geqslant i$ or (ii) $t_j > t_i$. The interesting case is $e \rightsquigarrow e'$. The case $e' \rightsquigarrow e$ would contradict $\sigma_1 \in \mathit{Traces}(\mathcal{E})$ since e occurs before e' and in all other cases the order and timing of events is unchanged. Consider $e \rightsquigarrow e'$. Since $\sigma_1 \in \mathit{Traces}(\mathcal{E})$ then $t \leqslant t'$ which contradicts $t' < t$.

(2) $X \overset{I}{\mapsto} e_i$ and (i) $X \cap \{\, e_1, \ldots, e_{i-1} \,\} = \varnothing$ or (ii) $t_i \notin t_j + I$ where $j < i$ and $e_j \in X$. By a similar reasoning as above, we conclude that the interesting case is $X \mapsto e'$ with $e \in X$. Since $\sigma_1 \in \mathit{Traces}(\mathcal{E})$ then $t' \in t + I$, so $t' \geqslant t$, which contradicts $t' < t$.

(3) $t_i \notin \mathcal{A}(e_i)$. This would contradict with $\sigma_1 \in \mathit{Traces}(\mathcal{E})$.

(4) $t_i > \mathsf{tm}_\rho(\hat{e})$ for some urgent event $\hat{e}$ enabled after $\rho = (e_1,t_1)\ldots(e_{i-1},t_{i-1})$, a prefix of σ_2, such that (i) $e_i \rightsquigarrow \hat{e}$ or (ii) $\hat{e} \rightsquigarrow e_i$. The interesting cases are (1) $e_i = e$ and (2) $e_i = e'$; the other cases lead directly to a contradiction with $\sigma_1 \in \mathit{Traces}(\mathcal{E})$.

(i1) $e_i \rightsquigarrow \hat{e}$ and $e_i = e$. So, $\rho = \sigma(e',t')$. For $\hat{e} = e'$ we have $e \rightsquigarrow e'$ which would lead to a contradiction, see case (1) above. Assume $\hat{e} \neq e'$. In case $\hat{e}$ would be enabled after σ, it follows from $\sigma_1 \in \mathit{Traces}(\mathcal{E})$ that $t_i \leqslant \mathsf{tm}_\rho(\hat{e})$, and a contradiction follows. Otherwise, the enabling of $\hat{e}$ necessarily depends on e', i.e. $X \mapsto \hat{e}$ and $e' \in X$. (In case $e' \rightsquigarrow \hat{e}$, $\hat{e}$ would be enabled after σ.) But then, since $e \rightsquigarrow \hat{e}$, it follows from condition (P2) that either $X \mapsto e$ or $X \rightsquigarrow e$. Both cases contradict with $\sigma_1 \in \mathit{Traces}(\mathcal{E})$, since e' occurs after e in σ_1 and this would not be possible if $X \mapsto e$ or $e' \rightsquigarrow e$, given that e' occurs in σ_1.

(i2) $e_i \rightsquigarrow \hat{e}$ and $e_i = e'$. So, $\rho = \sigma$. As for case (i1), assume $\hat{e} \neq e$. From $\sigma_1 \in \mathit{Traces}(\mathcal{E})$ it follows that $t \leqslant \mathsf{tm}_\rho(\hat{e})$. Since $t' < t$, it follows $t' \leqslant \mathsf{tm}_\rho(\hat{e})$. Contradiction.

(ii1) $\hat{e} \rightsquigarrow e_i$ and $e_i = e$. So, $\rho = \sigma(e',t')$. Similar to case (i1).

(ii2) $\hat{e} \rightsquigarrow e_i$ and $e_i = e'$. So, $\rho = \sigma$. Similar to case (i2).

Note that the reverse implication does not hold; for instance, if e causally depends on e' then the order of events $e'\,e$ in a trace cannot be reversed since this would contradict their causal ordering.

This result can be interpreted as follows: the set of timed event traces obtained from a timed event structure can be partitioned in equivalence classes, where each equivalence class consists of traces containing identical elements (i.e. pairs

of events and time points). An equivalence class does not distinguish among total order executions that are equivalent up to the reordering of independent events. This leads to a more abstract representation of concurrency than timed event traces, and is similar to the treatment of traces by Mazurkiewicz [31].

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 [22,23]. We start with some basic notions. Let Events be a set such that for any event $e \in \mathsf{Events}$, $(e,*),(*,e) \in \mathsf{Events}$, and 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 $\mathit{init}(\mathcal{E})$ be the set of initial events of $\mathcal{E}$ and $\mathit{exit}(\mathcal{E})$ its set of successful termination events, i.e. $\mathit{init}(\mathcal{E}) =_{df} \{\, e \in E \mid \neg(\exists X \subseteq E : X \mapsto e) \,\}$ and $\mathit{exit}(\mathcal{E}) =_{df} \{\, e \in E \mid l(e) = \surd \,\}$.

In the rest of this section let $\mathcal{E},\mathcal{E}_1,\mathcal{E}_2 \in \mathsf{TES}$ and $\mathcal{E}_1 = (E_1,\leadsto_1,\mapsto_1,l_1,\mathcal{A}_1,\mathcal{R}_1,\mathcal{U}_1)$, $\mathcal{E}_2 = (E_2,\leadsto_2,\mapsto_2,l_2,\mathcal{A}_2,\mathcal{R}_2,\mathcal{U}_2)$ such that w.l.o.g. $E_1 \cap E_2 = \varnothing$. Let $\hat{\tau}$ denote the urgent variant of τ.

Definition 8 *(Action-prefix). For $a \in \mathsf{Obs} \cup \{\, \tau,\hat{\tau} \,\}$ and $I \subseteq [0,\infty)$ let*

$$a_I \,.\, \mathcal{E}_1 =_{df} (E_1 \cup \{\, e_a \,\},\leadsto_1,\mapsto,l_1 \cup \{\, (e_a,a) \,\},\mathcal{A},\mathcal{R},\mathcal{U}) \text{ where}$$

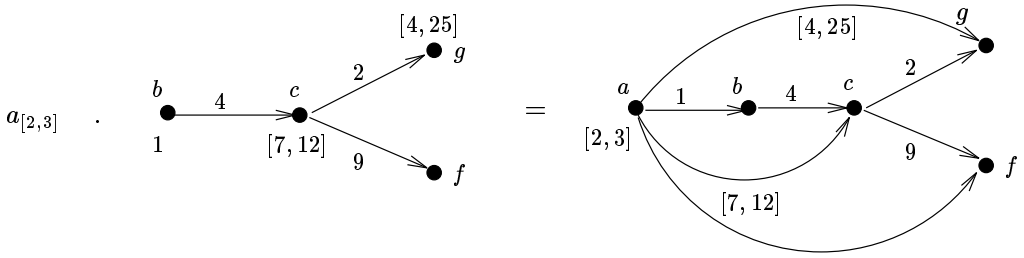
- $\mapsto = \mapsto_1 \;\cup\; (\{\, \{\, e_a \,\} \,\} \times E_1)$
- $\mathcal{A} = \{\, (e_a,I) \,\} \cup (E_1 \times \{\, [0,\infty) \,\})$
- $\mathcal{R} = \mathcal{R}_1 \cup \{\, ((\{\, e_a \,\},e),\mathcal{A}_1(e)) \mid e \in E_1 \,\}$
- $\mathcal{U} = \mathsf{if}\ a = \hat{\tau}\ \mathsf{then}\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

where we assume that $e_a \notin E_1$.

$\hat{\tau}_I \,.\, \mathcal{E}$ 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 $\rhd$ in a concise way. Notice that for $\hat{\tau}_I \,.\, \mathcal{E}$ set I must be either empty or be a point interval in order to guarantee constraint (P3).

In $a_I \,.\, \mathcal{E}$ a bundle is introduced from a new event e_a (labelled a) to all events in $\mathcal{E}$. The delay of each of these events becomes relative to e_a, so for every such event e each bundle $\{\, e_a \,\} \mapsto e$ is associated with a delay $\mathcal{A}(e)$, and $\mathcal{A}(e)$ becomes $[0,\infty)$. $\mathcal{A}(e_a)$ becomes I. In the untimed case it suffices to only introduce bundles from e_a to the initial events of $\mathcal{E}$, cf. [26,27]. The bundles to all events of $\mathcal{E}$ that are introduced in the timed case are used for the sole

purpose of making delays relative to e_a. As an example of prefixing consider [7] :

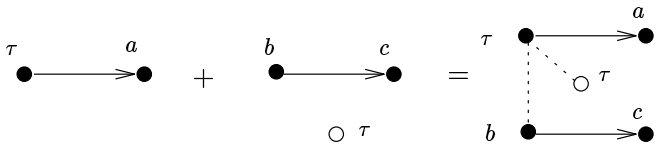


Definition 9 *(Choice).*

$$\mathcal{E}_1 + \mathcal{E}_2 =_{df} (E_1 \cup E_2, \rightsquigarrow, \mapsto_1 \cup \mapsto_2, l_1 \cup l_2, \mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2, \mathcal{U}_1 \cup \mathcal{U}_2)$$

where $\rightsquigarrow = \rightsquigarrow_1 \cup \rightsquigarrow_2 \cup (init(\mathcal{E}_1) \times init(\mathcal{E}_2)) \cup (init(\mathcal{E}_2) \times init(\mathcal{E}_1))$.

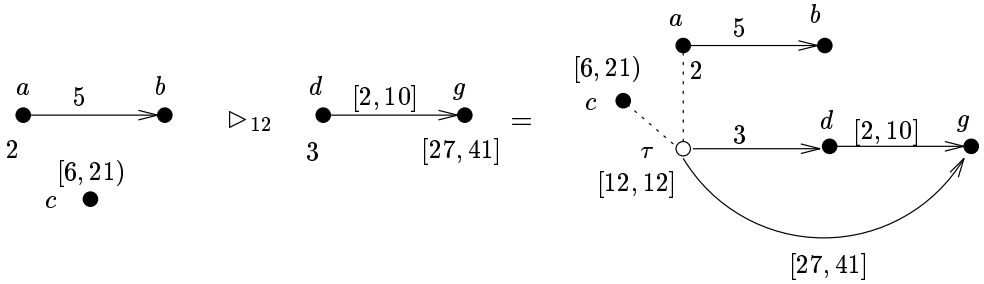
For choice consider the following example. Since the timings of events and bundles are unaffected we omit these for convenience.



For $\mathcal{E}_1 \rhd_t \mathcal{E}_2$ a new internal urgent event e with delay $\{t\}$ 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, event e is put in mutual conflict with all initial events of $\mathcal{E}_1$, like for choice.

Definition 10 *(Timeout). For* $t \in [0, \infty)$ *let* $\mathcal{E}_1 \rhd_t \mathcal{E}_2 =_{df} \mathcal{E}_1 + \hat{\tau}_{\{t\}} . \mathcal{E}_2$.

As an example of the timeout operator consider:



Definition 11 *(Abstraction). For* $A \subseteq \mathsf{Obs}$ *let* $\mathcal{E} \backslash A =_{df} (E, \rightsquigarrow, \mapsto, l', \mathcal{A}, \mathcal{R}, \mathcal{U})$ *where* $(l(e) \in A \Rightarrow l'(e) = \tau) \wedge (l(e) \notin A \Rightarrow l'(e) = l(e))$.

[7] Recall that $[t, \infty)$ is simply denoted by t.

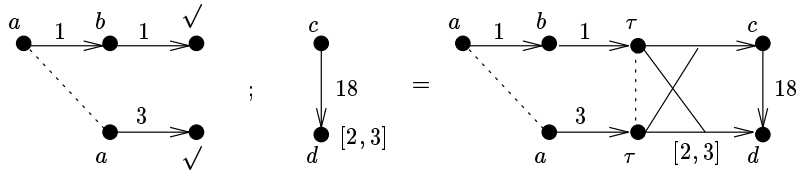
Definition 12 *(Relabelling). For* $\lambda : \mathsf{Act} \longrightarrow \mathsf{Act}$ *with* $\lambda(\tau) = \tau$ *and* $\lambda(\surd) = \surd$ *let* $\mathcal{E}[\lambda] =_{df} (E, \rightsquigarrow, \mapsto, \lambda \circ l, \mathcal{A}, \mathcal{R}, \mathcal{U})$*, where* $\circ$ *denotes function composition.*

Definition 13 *(Sequential composition).*

$$\mathcal{E}_1 \,;\, \mathcal{E}_2 =_{df} (E_1 \cup E_2, \rightsquigarrow, \mapsto, l, \mathcal{A}, \mathcal{R}, \mathcal{U}_1 \cup \mathcal{U}_2) \text{ where}$$

- $\rightsquigarrow = \rightsquigarrow_1 \;\cup\; \rightsquigarrow_2 \;\cup (exit(\mathcal{E}_1) \times exit(\mathcal{E}_1)) \setminus Id_{E_1}$
- $\mapsto = \mapsto_1 \;\cup\; \mapsto_2 \;\cup (\{\, exit(\mathcal{E}_1) \,\} \times E_2))$
- $l = ((l_1 \cup l_2) \setminus (exit(\mathcal{E}_1) \times \{\, \surd \,\})) \cup (exit(\mathcal{E}_1) \times \{\, \tau \,\})$
- $\mathcal{A} = \mathcal{A}_1 \cup (E_2 \times \{\, [0, \infty) \,\})$
- $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \{\, ((exit(\mathcal{E}_1), e), \mathcal{A}_2(e)) \mid e \in E_2) \,\}$.

Bundles are introduced between the successful termination events of $\mathcal{E}_1$ and the events in $\mathcal{E}_2$. In order to create bundles, mutual conflicts are introduced between the successful termination events of $\mathcal{E}_1$. The successful termination events of $\mathcal{E}_1$ are relabelled into internal events. The reason for introducing bundles to all events (and not only the initial ones) of $\mathcal{E}_2$ is to make event delays in $\mathcal{E}_2$ relative to the termination of $\mathcal{E}_1$. This is similar as for action-prefix. As an example of how $\mathcal{E}_1 \,;\, \mathcal{E}_2$ is computed consider:

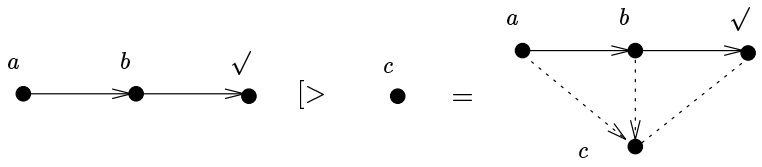


Definition 14 *(Disrupt).*

$$\mathcal{E}_1 [> \mathcal{E}_2 =_{df} (E_1 \cup E_2, \rightsquigarrow, \mapsto_1 \cup \mapsto_2, l_1 \cup l_2, \mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2, \mathcal{U}_1 \cup \mathcal{U}_2)$$

where $\rightsquigarrow = \rightsquigarrow_1 \;\cup\; \rightsquigarrow_2 \;\cup (E_1 \times init(\mathcal{E}_2)) \cup (init(\mathcal{E}_2) \times exit(\mathcal{E}_1))$.

$\mathcal{E}_1 [> \mathcal{E}_2$ is equal to the union of $\mathcal{E}_1$ with $\mathcal{E}_2$ extended with some conflicts. Each event in $\mathcal{E}_1$ may be disabled by an initial event of $\mathcal{E}_2$. This models the fact that $\mathcal{E}_1$ is disrupted once an initial event of $\mathcal{E}_2$ happens. In addition, after the occurrence of a successful termination event in $\mathcal{E}_1$ no initial event of $\mathcal{E}_2$ can happen anymore. As an example of how $\mathcal{E}_1 [> \mathcal{E}_2$ is computed consider the following. Like for the example of choice, the bundle and event delays are omitted since they are unaffected by $[>$.



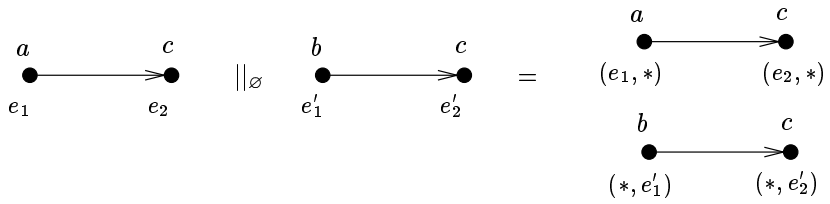
The definition of parallel composition is a bit more involved. The events of $\mathcal{E}_1 \,||_A\, \mathcal{E}_2$ are constructed in the following way: an event e of E_i $(i{=}1,2)$ that does not need to synchronise is paired with the auxiliary symbol $*$, and an event which is labelled with $\surd$ 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). For each event (e_1, e_2) in the parallel composition, the bundles $X \mapsto (e_1, e_2)$ are obtained by the "lifting" of the bundles $X_i \mapsto_i e_i$ of the components $\mathcal{E}_i$. Let for $A \subseteq \mathsf{Obs}$, $E_i^s =_{df} \{\, e \in E_i \mid l_i(e) \in A \cup \{\surd\}\,\}$ be the set of synchronising events and $E_i^f =_{df} E_i \setminus E_i^s$ the set of 'free' events.

Definition 15 *(Parallel composition). For $A \subseteq \mathsf{Obs}$ let*

$$\mathcal{E}_1 \,||_A\, \mathcal{E}_2 =_{df} (E, \rightsquigarrow, \mapsto, l, \mathcal{A}, \mathcal{R}, \mathcal{U}) \text{ where}$$

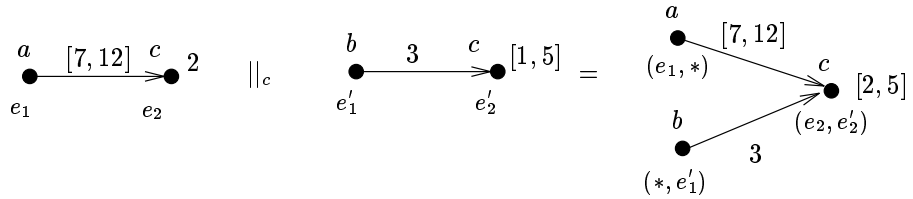
- $E = (E_1^f \times \{*\}) \cup (\{*\} \times E_2^f) \cup \{\, (e_1, e_2) \in E_1^s \times E_2^s \mid l_1(e_1) = l_2(e_2) \,\}$
- $(e_1, e_2) \rightsquigarrow (e_1', e_2')$ *iff*
  - $(e_1 \rightsquigarrow_1 e_1') \;\vee\; (e_2 \rightsquigarrow_2 e_2')$ *or*
  - $(e_1 = e_1' \neq * \;\wedge\; e_2 \neq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

Example 16 *In the first example of parallel composition the timings of events and bundles are unaffected and are omitted for convenience.*

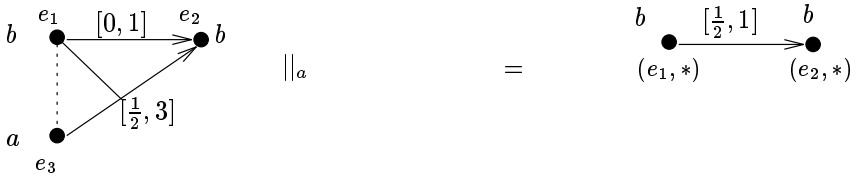


Synchronisation leads to pairing of events, and intersection of the event delays

of its components, cf.



Intersection of bundle delays is illustrated by the following example where the left-hand tes is composed with the empty event structure:



In Section 3 we motivated the use of bundles for modelling parallel composition in a rather intuitive way. Due to the impossibility to have different (conflicting) causes for a single event, the definition of parallel composition on prime event structures is much more involved [29,39]. For flow event structures, the definition of parallel composition poses some technical problems that can be solved by imposing additional structural constraints on the event structures [15].

We can now establish the following closure result:

Theorem 17 TES *is closed under* $a_I\,.\,, +, \backslash A$, $[\lambda]$, ; , $[>$, $||_A$, *and* $\rhd_t$.

PROOF. Let $\mathcal{E}_1, \mathcal{E}_2 \in$ TES. We provide the proofs for $[>$ and $||_A$; the proofs for the other constructs are similar (and simpler). We concentrate our proof on the constraints (P2) and (P3) of Definition 1. The proofs for irreflexivity of $\rightsquigarrow$ and (P1) follow directly from [26] and are omitted here. The fact that urgent events are internal is easy to check and omitted.

(1) $\mathcal{E} = \mathcal{E}_1 [> \mathcal{E}_2$. The proof of (P3) is easy since the event and bundle delays are unaffected by $[>$ and no urgent events are introduced by it. Consider constraint (P2). Let $e \in \mathcal{U}$.

(i) Assume $e' \rightsquigarrow e$ and $X \mapsto e$. If $e' \rightsquigarrow_1 e$ or $e' \rightsquigarrow_2 e$ then the validity of (P2) follows directly from $\mathcal{E}_1, \mathcal{E}_2 \in$ TES. Consider $e' \not\rightsquigarrow_1 e$ and $e' \not\rightsquigarrow_2 e$. From Definition 14 it follows that we have to consider the cases $e' \in E_1$ and $e \in \mathit{init}(\mathcal{E}_2)$, and $e' \in \mathit{init}(\mathcal{E}_2)$ and $e \in \mathit{exit}(\mathcal{E}_1)$. For the latter (P2) follows, since $e \in \mathit{exit}(\mathcal{E}_1)$ contradicts the assumption $e \in \mathcal{U}$ while urgent events are internal. For the former case (P2) also follows, since $e \in \mathit{init}(\mathcal{E}_2)$ contradicts the assumption $X \mapsto e$.

(ii) Assume $e \rightsquigarrow e'$ and $X \mapsto e$. Like for (i) consider $e \not\rightsquigarrow_1 e'$ and $e \not\rightsquigarrow_2 e'$. Consider (the symmetric cases of (i)) $e \in E_1$ and $e' \in \mathit{init}(\mathcal{E}_2)$, and $e \in \mathit{init}(\mathcal{E}_2)$ and $e' \in \mathit{exit}(\mathcal{E}_1)$. The latter case is straightforward since $e \in \mathit{init}(\mathcal{E}_2)$ contradicts $X \mapsto e$. Consider the former case. From the assumption $X \mapsto e$ and Definition 14 it follows $X \mapsto_1 e$. Since $\mathcal{E}_1 \in \mathsf{TES}$ and the fact that e' is initial we have $X \rightsquigarrow_1 e'$ for $X \subseteq E_1$, and consequently $X \rightsquigarrow e'$.

(2) $\mathcal{E} = \mathcal{E}_1 \,||_A\, \mathcal{E}_2$. Let $e = (e_1, e_2) \in \mathcal{U}$. Since urgent events are internal we have $e_1 = *$ and $e_2 \in E_2$, or the reverse. By symmetry it suffices to consider e.g. $e = (*, e_2)$ with $e_2 \in E_2$.

(P2) Let $X \mapsto e$. The cases $e \rightsquigarrow e'$ and $e' \rightsquigarrow e$ are proven in a similar way. We consider $e' \rightsquigarrow e$. Let $e' = (e'_1, e'_2)$. From $(e'_1, e'_2) \rightsquigarrow (*, e_2)$ and Definition 15 it follows that $e'_2 \rightsquigarrow_2 e_2$. In addition, since $e = (*, e_2)$ we have $X = \{\, (e''_1, e''_2) \in E \mid e''_2 \in X_2 \,\}$ where $X_2 \mapsto_2 e_2$. Since $\mathcal{E}_2 \in \mathsf{TES}$ it follows $X_2 \mapsto_2 e'_2$ or $X_2 \rightsquigarrow_2 e'_2$, and by Definition 15, $X \mapsto e'$ or $X \rightsquigarrow e'$.

(P3) Since $\mathcal{E}_2 \in \mathsf{TES}$ we have that $\mathcal{A}_2(e_2) \subseteq [t,t]$ or $X_2 \stackrel{I}{\mapsto}_2 e_2$ with $I \subseteq [t,t]$ for some t. For the former case the validity of (P3) follows since $\mathcal{A}(*, e_2) = [0, \infty) \cap \mathcal{A}_2(e_2) = \mathcal{A}_2(e_2)$. For the second case, it follows from Definition 15 that there is a bundle $X \mapsto e$ with delay $\mathcal{R}(X, e) \subseteq \mathcal{R}_2(X_2, e_2)$ and thus (P3) is satisfied.

5 A metric denotational semantics

In this section we provide a metric denotational semantics for our process algebra. In Section 5.1 we summarise the main ingredients of metric spaces that are needed for the understanding of the rest of this paper. The use of metric spaces for denotational semantics is summarised in Section 5.2. Readers familiar with these topics might want to skip these sections. The basis for an appropriate distance notion is time truncation as described in Section 5.3. Section 5.4 defines a complete ultra-metric space based on time truncation. Time-guardedness is defined in Section 5.5 and a semantics for time-guarded specifications is provided in Section 5.6

5.1 A resumé of metric spaces

A more thorough treatment of metric spaces can be found in, for instance, [16].

Definition 18 *(Metric space). For set A and $d : A \times A \longrightarrow \mathbb{R}$, the pair $\langle A, d \rangle$*

is a metric space *if for all* x *and* $y \in A$*:*

(1) $d(x,y) \geqslant 0$
(2) $d(x,y) = 0 \Leftrightarrow x = y$
(3) $d(x,z) \leqslant d(x,y) + d(y,z)$ *for all* $z \in A$

$\langle A, d\rangle$ *is called an* ultra-metric space *if constraint (3) is replaced by (the stronger) constraint* $d(x,z) \leqslant \max(d(x,y), d(y,z))$*. If constraint (2) is weakened into* $d(x,y) = 0 \Leftarrow x = y$*, then the pair* $\langle A, d\rangle$ *is called a* pseudo-metric *space.*

In this paper we consider one-bounded distance functions, i.e. $d(x,y) \leqslant 1$ for all $x, y \in A$. We will also basically deal with ultra-metric spaces, which is quite natural when the distance function corresponds to the reciprocal of the number of computation steps two processes coincide.

We assume that $\langle A \times A, d'\rangle$ is equipped with the distance

$$d'((x,y),(x',y')) = max\{\, d(x,x'), d(y,y')\,\}$$

for $x, x', y, y' \in A$.

If (x_n) is a sequence in $\langle A, d\rangle$ and $x \in A$ then x is called the *limit* of (x_n) iff

$$\forall\, \varepsilon > 0 : (\exists\, N \in \mathbb{N} : \forall\, n \geqslant N : d(x_n, x) < \varepsilon).$$

$\langle A, d\rangle$ is a *complete metric space* (cms) if each Cauchy sequence has a limit, where a Cauchy sequence is a sequence (x_n), $x_i \in A$, such that

$$\forall\, \varepsilon > 0 : (\exists\, N \in \mathbb{N} : \forall\, m, n \geqslant N : d(x_m, x_n) < \varepsilon).$$

Definition 19 *(Contracting). For* $\langle A, d\rangle$ *a metric space, function* $f : A \longrightarrow A$ *is* contracting *if there exists a real number* $c \in [0,1)$ *such that*

$$\forall\, x, y \in A : d(f(x), f(y)) \leqslant c \cdot d(x,y).$$

In that case, c *is called a* contraction coefficient *of* f*. Function* f *is called* non-distance increasing *or* non-expansive *iff*

$$\forall\, x, y \in A : d(f(x), f(y)) \leqslant d(x,y).$$

Banach's fixed point theorem now says that for each contracting function on a cms there exists a unique fixed point.

Theorem 20 *(Banach's fixed point theorem). For* $\langle A, d\rangle$ *with* $A \neq \varnothing$ *a com-*

plete metric space and $f : A \longrightarrow A$ a contracting function on $\langle A, d\rangle$ we have

(1) f has a unique fixed point, say x, and
(2) any sequence (x_n) such that $x_{i+1} = f(x_i)$ has limit x.

5.2 Denotational semantics

We only give a brief account of our approach; see [35,10,6,11] for more information on the use of metrics for denotational semantics. The semantic domain S — in our case a suitable variant of TES — for PA is equipped with a set Op' of operators that reflect the operators Op of Expr. For any fixed declaration $decl$, the function $P \mapsto \mathcal{M}(decl, P)$ for $P \in$ Expr is a homomorphism from (Expr, Op) to (S, Op') such that the meaning of process variable x is given by $decl(x)$. The requirement of being a homomorphism is an algebraic characterisation of the fact that $\mathcal{M}$ is compositional, that is, the meaning of a composed program $op(P_1, \ldots, P_n)$ with $op \in Op$ can be obtained by applying the corresponding semantic operator $op' \in Op'$ to the meanings $\mathcal{M}(P_i)$ of the modules P_i, shortly

$$\mathcal{M}(decl, op(P_1, \ldots, P_n)) \;=\; op'(\mathcal{M}(decl, P_1), \ldots, \mathcal{M}(decl, P_n)).$$

Function $\mathcal{M}$ satisfies these conditions iff, for any fixed declaration $decl$, the function $P \mapsto \mathcal{M}(decl, P)$ is a fixed point of the higher-order function F_{decl} : $[\mathsf{Expr} \longrightarrow S] \longrightarrow [\mathsf{Expr} \longrightarrow S]$, defined (in our case) by:

$$\begin{aligned}
F_{decl}(\phi)(\mathbf{0}) &=_{df} \mathbf{0}' \\
F_{decl}(\phi)(\mathbf{1}) &=_{df} \mathbf{1}' \\
F_{decl}(\phi)(x) &=_{df} \phi(decl(x)) \\
F_{decl}(\phi)(op\ P) &=_{df} op'\ F_{decl}(\phi)(P) && \text{for unary } op \\
F_{decl}(\phi)(P\ op\ Q) &=_{df} F_{decl}(\phi)(P)\ op'\ F_{decl}(\phi)(Q) && \text{for binary } op.
\end{aligned}$$

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 cms. Distance $\tilde{d}$ is obtained from the cms $\langle S, d\rangle$ where

$$\tilde{d}(\phi_1, \phi_2) =_{df} sup\{\, d(\phi_1(P), \phi_2(P)) \mid P \in \mathsf{Expr}\,\} \qquad (1)$$

for $\phi_1, \phi_2 : \mathsf{Expr} \longrightarrow S$. Function F_{decl} is contracting on $\langle[\mathsf{Expr} \longrightarrow S], \tilde{d}\rangle$ if its constituents $;$, $\rhd_t$, $||_A$ and so on, are non-distance increasing on $\langle S, d\rangle$ and contracting in certain arguments [6,12]. Our first concern is to find an

appropriate function d on the semantical domain S, in our case TES. The semantics of PA is then obtained by $\mathcal{M}(decl, P) =_{df} \phi_{decl}(P)$, where ϕ_{decl} : Expr $\longrightarrow S$ is the unique fixed point of F_{decl}.

5.3 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

$$mintime_{\mathcal{E}}(e) =_{df} inf\{\, t \in \mathbb{R}^+ \mid \exists \sigma \in Traces(\mathcal{E}) : (e,t) \in \sigma \,\}$$

where by convention $inf\, \varnothing =_{df} \infty$. For $t \in \mathbb{R}^+$ and $X \subseteq E$ let $X \restriction t =_{df} \{\, e \in X \mid mintime_{\mathcal{E}}(e) < t \,\}$, the set of events in X that can occur strictly before t. Notice that $X \restriction 0 = \varnothing$ for any X. Let $X \restriction \infty =_{df} \bigcup_{t \geqslant 0} X \restriction t$, i.e. $X \restriction \infty$ is the set of events that can occur. Event e is called *executable* iff $e \in E \restriction \infty$, i.e. if $mintime_{\mathcal{E}}(e) < \infty$.

Definition 21 *(Time truncation). The* truncation *of $\mathcal{E}$ up to $t \in \mathbb{R}^+ \cup \{\infty\}$ is defined by $\mathcal{E} \restriction t =_{df} (E \restriction t, \rightsquigarrow_t, \mapsto_t, l_t, \mathcal{A}_t, \mathcal{R}_t, \mathcal{U}_t)$ where $l_t = l \restriction (E \restriction t)$, $\mathcal{A}_t(e) = \mathcal{A}(e) \cap [0,t)$, $\mathcal{U}_t = \mathcal{U} \restriction t$, and*

- $\rightsquigarrow_t = \rightsquigarrow \cap (E \restriction t \times E \restriction t)$
- $X \mapsto_t e$ *iff there exists* $Y \mapsto e$ *with* $Y \restriction t = X$
- $\mathcal{R}_t(X,e)$ *is the set of all time points* $u \in [0,t[$ *such that there is some timed event trace* $(e_1,t_1)(e_2,t_2)\ldots(e_n,t_n)$ *for which the following conditions hold:*
  - $t_n < t$ *and* $e_n = e$
  - *there is some* $j < n$ *with* $e_j \in X$ *and* $u = t_n - t_j$.

Remark that $\mathcal{E} \restriction 0$ is the empty tes. By straightforward proof one can establish that

Lemma 22 TES *is closed under time truncation.*

Lemma 23 $\forall\, t \geqslant 0 : \mathcal{E} \restriction t = (\mathcal{E} \restriction \infty) \restriction t$.

Example 24 *Time truncation is illustrated by Figure 2. It depicts (a) a tes $\mathcal{E}$ and (b) its truncation $\mathcal{E} \restriction 6$ up to time 6. Events b, f and g are eliminated in $\mathcal{E} \restriction 6$, since the minimal time at which they can occur, time 11, 8 and 6, respectively, is at least 6. Note that $\{\, a \,\} \stackrel{[1,3]}{\mapsto}_6 c$, since the minimal delay between events a and c is 1 ($[1,\infty) \cap [0,9] = [1,9]$ since $\{\, a,b \,\} \stackrel{1}{\mapsto} c$ and $\{\, a \,\} \stackrel{[0,9]}{\mapsto} c$), whereas the maximal delay is at most 3 time units (in the scenario in which a happens at time 3, and c should happen before 6). In a similar way, we obtain $\{\, c \,\} \stackrel{[1,2)}{\mapsto} d$.*

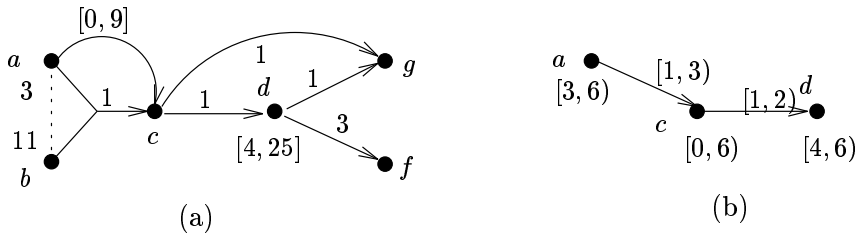


Fig. 2. Time truncation

The idea of time truncation is that by enlarging the time span during which an event structure is considered, we obtain more information about its behaviour. In the limit, that is for an infinite time span, we would expect to capture the entire behaviour of the event structure. The next theorem says that the behaviour of $\mathcal{E}$ can indeed be approximated by its time truncations. In order to pave the way towards its proof we provide the following lemmata.

Lemma 25 *For $\sigma = (e_1, t_1) \ldots (e_n, t_n) \in \mathit{Traces}(\mathcal{E})$ such that $t_i < t$ for all $0 < i \leqslant n$:*

$$\mathsf{en}^{\mathcal{E}}(e_1 \ldots e_n) \restriction t = \mathsf{en}^{\mathcal{E}\restriction t}(e_1 \ldots e_n).$$

PROOF. By checking inclusion in both directions.

(i) '$\subseteq$': let $e \in \mathsf{en}^{\mathcal{E}}(e_1 \ldots e_n) \restriction t$. Then $e \not\rightsquigarrow e_i$, for $0 < i \leqslant n$, and by Definition 21 it follows $e \not\rightsquigarrow_t e_i$. If there is no bundle in $\mathcal{E} \restriction t$ pointing to e, then we yield $e \in \mathsf{en}^{\mathcal{E}\restriction t}(e_1 \ldots e_n)$. Suppose that $X \mapsto_t e$. Then, according to Definition 21, $Y \mapsto e$ for some Y with $Y \restriction t = X$. Since $t_i < t$ for $0 < i \leqslant n$, we have that $(X \cap \{ e_1, \ldots, e_n \} \neq \varnothing) \Leftrightarrow (Y \cap \{ e_1, \ldots, e_n \} \neq \varnothing)$, and $e \in \mathsf{en}^{\mathcal{E}\restriction t}(e_1 \ldots e_n)$.

(ii) '$\supseteq$': let $e \in \mathsf{en}^{\mathcal{E}\restriction t}(e_1 \ldots e_n)$. Then $e \not\rightsquigarrow_t e_i$, for $0 < i \leqslant n$, and by Definition 21 it follows that $e \not\rightsquigarrow e_i$. If there is no bundle in $\mathcal{E}$ pointing to e then the property follows directly. Suppose $X \mapsto e$. Then, by Definition 21, $X \restriction t \mapsto_t e$. But then, $(X \cap \{ e_1, \ldots, e_n \} \neq \varnothing) \Leftrightarrow ((X \restriction t) \cap \{ e_1, \ldots, e_n \} \neq \varnothing)$ since $t_i < t$ for $0 < i \leqslant n$, and $e \in \mathsf{en}^{\mathcal{E}}(e_1 \ldots e_n) \restriction t$.

From the definition of time truncation and the previous lemma, it is not difficult to show that:

Theorem 26 $\mathit{Traces}(\mathcal{E}) = \bigcup_{t \geqslant 0} \mathit{Traces}(\mathcal{E} \restriction t)$.

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:

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

Remark that $\mathcal{E} \upharpoonright 0$ is the empty tes, so each pair of tes's agrees at least up to time 0. Also notice that $d(\mathcal{E}, \mathcal{E} \upharpoonright t) \leqslant 2^{-t}$ for all t. Although this basic notion of distance 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 is that tes's may contain events that can never appear. This is due, for example, to empty bundles, circular bundles, or inconsistent timing constraints. Such events can, for instance, appear in the semantics for expressions like $\mathbf{0} \,||_a\, a \,.\, \mathbf{0}$, $a \,.\, b \,.\, \mathbf{0} \,||_{\{a,b\}}\, b \,.\, a \,.\, \mathbf{0}$, or when timing constraints are specified 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 [27,22] that preserve timed event traces, but it is hard to adapt the definitions of the operators on event structures such that these events are eliminated during construction.

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 examples suggest to abstract from events that can never be executed. This motivates the use of restrictions of $\mathcal{E}$ to its set $\mathcal{E} \upharpoonright \infty$ of executable events. In the above example with $\varnothing \mapsto e$ it would mean that event e is not considered. The idea of those restrictions is that executable events are unaffected. This follows from:

Lemma 27 *Traces*$(\mathcal{E} \upharpoonright \infty) =$ *Traces*$(\mathcal{E})$.

PROOF.

$$\begin{array}{ll} & \textit{Traces}(\mathcal{E} \upharpoonright \infty) \\ = & \quad \{ \text{ Theorem 26 } \} \\ & \bigcup_{t \geqslant 0} \textit{Traces}((\mathcal{E} \upharpoonright \infty) \upharpoonright t) \\ = & \quad \{ \text{ Lemma 23 } \} \end{array}$$

$$
\begin{array}{ll}
 & \bigcup_{t \geqslant 0} \mathit{Traces}(\mathcal{E} \restriction t) \\
= & \quad \{ \text{ Theorem 26 } \} \\
 & \mathit{Traces}(\mathcal{E}).
\end{array}
$$

The equivalence $\simeq$ intended above is now defined by $\mathcal{E}_1 \simeq \mathcal{E}_2$ if and only if $\mathcal{E}_1 \restriction \infty = \mathcal{E}_2 \restriction \infty$.

It is quite standard to abstract from event identities in metric semantics [11,29], i.e. to deal with isomorphism classes of semantic structures. The event identities are only needed for technical reasons but they are meaningless for the semantics of an expression. The following definition is the usual notion of isomorphism with the only exception that the bijection is defined over the executable events.

Definition 28 *(Isomorphism). Tes's* $\mathcal{E}_i = (E_i, \leadsto_i, \mapsto_i, l_i, \mathcal{A}_i, \mathcal{R}_i, \mathcal{U}_i)$ *for* i=1, 2 *are* isomorphic *if there exists a bijection* $f : E_1 \longrightarrow E_2$ *such that* $l_2 \circ f = l_1$, $\mathcal{A}_2 \circ f = \mathcal{A}_1$ *and*

(1) $e_1 \leadsto_1 e_2$ *iff* $f(e_1) \leadsto_2 f(e_2)$ *for all* $e_1, e_2 \in E_1$,
(2) $X \stackrel{I}{\mapsto}_1 e$ *iff* $f(X) \stackrel{I}{\mapsto}_2 f(e)$ *for all* $e \in E_1, X \subseteq E_1$, *and*
(3) $e \in \mathcal{U}_1 \restriction \infty$ *iff* $f(e) \in \mathcal{U}_2$.

$\mathcal{E}_1$ *and* $\mathcal{E}_2$ *are called* timed isomorphic, *denoted* $\mathcal{E}_1 \simeq_{iso} \mathcal{E}_2$, *iff the tes's* $\mathcal{E}_1 \restriction \infty$ *and* $\mathcal{E}_2 \restriction \infty$ *are isomorphic.*

Note that $\mathcal{E} \simeq_{iso} \mathcal{E} \restriction \infty$. We write $f : \mathcal{E}_1 \longrightarrow \mathcal{E}_2$ to denote that f is an isomorphism from $\mathcal{E}_1$ to $\mathcal{E}_2$. 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 $\mathbf{E} \restriction t =_{df} \mathbf{E}_{\mathcal{E} \restriction t}$, where $\mathcal{E}$ is a representative of $\mathbf{E}$. The distance between equivalence classes (under $\simeq_{iso}$) of tes's is given by:

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

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

In order to motivate the next step towards (isomorphism classes of) finite approximable timed event structures consider the following example.

Example 29 *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*

- $E_1 = \{ (k, j) \mid j \geqslant 1 \ \wedge \ 0 < k \leqslant j \}$ *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*

- $\mathcal{R}_i(\{\,(k,j)\,\},(k{+}1,j)) = [1,1]$.

$\mathcal{E}_1$ and $\mathcal{E}_2$ are depicted in Figure 3(a) and (b), respectively. For simplicity, event labels, bundle delays and event identifiers are omitted. Then, $\mathcal{E}_1 \not\simeq_{iso} \mathcal{E}_2$

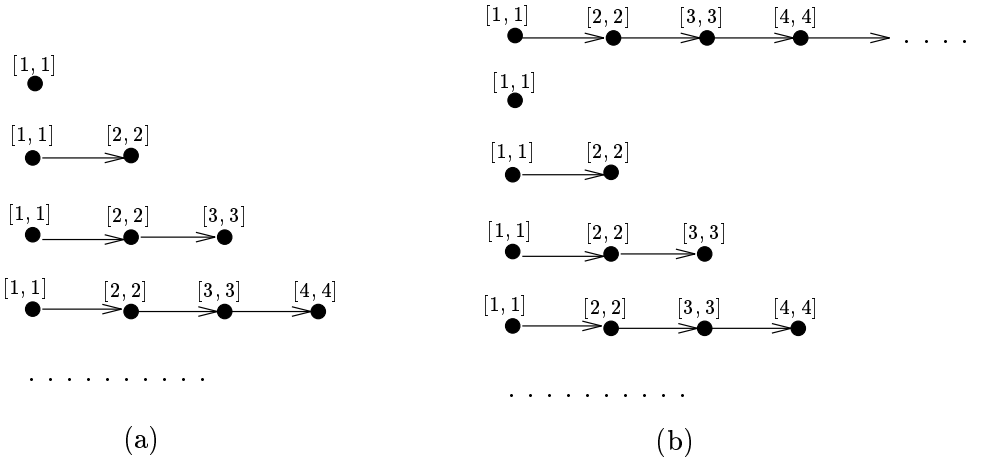


Fig. 3. Two non-isomorphic tes's for which all timed truncations are isomorphic

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 [29]. Note that this is not a real restriction, since for timed systems it is quite natural to avoid the execution of an infinite number of events in a finite time span (so called Zeno behaviours) [3,33].

Definition 30 *(Finite approximable). $\mathcal{E}$ is called* finitely approximable *iff $E \restriction t$ is finite for all $t \in \mathbb{R}^+$.*

Let $\mathsf{TES}_{fin}/\simeq_{iso}$ denote the isomorphism classes of finitely approximable tes's.

Lemma 31 *$\langle \mathsf{TES}_{fin}/\simeq_{iso}, \mathbf{d}\rangle$ is an ultra-metric space.*

PROOF. It is straightforward to check that $\mathbf{d}$ is a pseudo-ultra-metric on $\mathsf{TES}_{fin}/\simeq_{iso}$. We, therefore, concentrate on showing that $\mathbf{d}(\mathbf{E},\mathbf{E}') = 0 \;\Rightarrow\; \mathbf{E} = \mathbf{E}'$. Let $\mathbf{E}$, $\mathbf{E}' \in \mathsf{TES}_{fin}/\simeq_{iso}$ such that $\mathbf{d}(\mathbf{E},\mathbf{E}') = 0$ and let $\mathcal{E} = (E,\rightsquigarrow,\mapsto,l,\mathcal{A},\mathcal{R},\mathcal{U})$, $\mathcal{E}' = (E',\rightsquigarrow',\mapsto',l',\mathcal{A}',\mathcal{R}',\mathcal{U}')$ be representatives of $\mathbf{E}$ and $\mathbf{E}'$, respectively. The proof obligation is $\mathcal{E} \simeq_{iso} \mathcal{E}'$. (Then it follows, $\mathbf{E} = \mathbf{E}'$.) Since $\mathbf{d}(\mathbf{E},\mathbf{E}') = 0$ it follows from the definition of $\mathbf{d}$ that $\mathbf{E} \restriction t$ and $\mathbf{E}' \restriction t$ coincide for all t. The proof technique for showing $\mathbf{E} = \mathbf{E}'$ is to use the thus

existing isomorphisms between $\mathcal{E} \upharpoonright t$ and $\mathcal{E}' \upharpoonright t$ to construct an isomorphism between $\mathcal{E}$ and $\mathcal{E}'$.

Let $(t_n)_{n \geqslant 0}$ be a strictly monotonic sequence of non-negative reals with $t_0 = 0$ and $sup(t_n) = \infty$. Let $f_n : E \upharpoonright t_n \longrightarrow E' \upharpoonright t_n$ be an isomorphism. (From the above it follows that such isomorphism exists.) Clearly, $mintime_{\mathcal{E}}(e) = mintime_{\mathcal{E}'}(f_k(e))$ for all executable events e in $\mathcal{E}$. Moreover, for any event e in $\mathcal{E} \upharpoonright t$ and any time point t, we have $mintime_{\mathcal{E}}(e) = mintime_{\mathcal{E} \upharpoonright t}(e)$. (And the corresponding result for $\mathcal{E}'$.) This yields the following. If $e \in E \upharpoonright t_n$ then $f_k(e) \in E' \upharpoonright t_n$ for all $k \geq n$. We now define for $k \geqslant n$, functions $h_{k,n} : E \upharpoonright t_n \longrightarrow E' \upharpoonright t_n$ by $h_{k,n}(e) = f_k(e)$.

The idea is to define (by induction on $n \geqslant 0$) isomorphisms $g_n : E \upharpoonright t_n \longrightarrow E' \upharpoonright t_n$ and infinite sets I_n of natural numbers such that

(1) $I_0 \supseteq I_1 \supseteq I_2 \supseteq \ldots$ and
(2) $g_n = h_{k,n} (= f_k)$ for all $k \in I_n$.

Base case: let g_0 be the empty function and $I_0 = \{ n \mid n \geqslant 0 \}$.
Induction step: let $n \geqslant 1$ and assume g_k and I_k have been defined for all $0 \leqslant k < n$. Since $E \upharpoonright t_n$ and $E' \upharpoonright t_n$ are finite, the set of all functions from $E \upharpoonright t_n$ to $E' \upharpoonright t_n$ is finite. As I_{n-1} is infinite and the set of all functions from $E \upharpoonright t_n$ to $E' \upharpoonright t_n$ is finite, there exists some function $g_n : E \upharpoonright t_n \longrightarrow E' \upharpoonright t_n$ and some infinite subset I_n of I_{n-1} with $g_n = h_{k,n}$ for all $k \in I_n$.

This completes the definition of g_n and I_n for $n \geqslant 0$. Let function $f : E \upharpoonright \infty \longrightarrow E' \upharpoonright \infty$ be defined as follows: for event $e \in E$ with $mintime_{\mathcal{E}}(e) = t$ and $t_n > t$, let $f(e) = g_n(e)$. (Note that, if $k \geqslant n$ and $t_n > t$ then $g_k(e) = g_n(e)$ for all $e \in E \upharpoonright t$.) We now show that f is an isomorphism $\mathcal{E} \longrightarrow \mathcal{E}'$. From the construction of f in terms of the isomorphisms g_n, it follows that f is a bijection with $l = l' \circ f$, $\mathcal{A} = \mathcal{A}' \circ f$, $e \in \mathcal{U}$ iff $f(e) \in \mathcal{U}'$ and

- $e \rightsquigarrow e'$ iff $f(e) \rightsquigarrow' f(e')$,
- $mintime_{\mathcal{E}}(e) = mintime_{\mathcal{E}'}(f(e))$, and
- $f(X) \upharpoonright t_n \;=\; g_n(X \upharpoonright t_n)$ for all $n \geqslant 0$.

It remains to consider the bundle relations. Let $\mapsto_n$ ($\mathcal{R}_n$) and $\mapsto'_n$ ($\mathcal{R}'_n$) be the bundle relation (the bundle delay functions) of $\mathcal{E} \upharpoonright t_n$ and $\mathcal{E}' \upharpoonright t_n$, respectively. Let $e \in E$, n_0 a natural number with $t_{n_0} > mintime_{\mathcal{E}}(e)$ and assume $X \mapsto e$ is a bundle in $\mathcal{E} \upharpoonright \infty$ with $R_\infty(X, e) = I$. By Definition 21 it follows $X \upharpoonright t_n \mapsto_n e$ for all $n \geqslant n_0$ where $\bigcup \mathcal{R}_n(X \upharpoonright t_n, e) = R_\infty(X, e) = I$. Since $g_n : E \upharpoonright t_n \longrightarrow E' \upharpoonright t_n$ is an isomorphism, it follows

$$f(X) \upharpoonright t_n \;=\; g_n(X \upharpoonright t_n) \;\mapsto'_n\; g_n(e) = f(e)$$

and $\mathcal{R}'_n(f(X) \restriction t_n, f(e)) = \mathcal{R}'_n(g_n(X \restriction t_n), g_n(e)) = \mathcal{R}_n(X \restriction t_n, e)$ for all $n \geqslant n_0$. Thus, $f(X) \stackrel{I}{\mapsto}' f(e)$. In a similar way, we can show that $f(X) \stackrel{I}{\mapsto}' f(e)$ implies $X \stackrel{I}{\mapsto} e$.

This proves that f is an isomorphism from $\mathcal{E}_1$ to $\mathcal{E}_2$, and consequently, that $\mathcal{E}_1 \simeq_{iso} \mathcal{E}_2$. Hence, $\mathbf{E}_1 = \mathbf{E}_2$.

The main result that we need in order to define the metric semantics for PA as the unique fixed point of some higher-order function is completeness of the metric space that is considered.

Theorem 32 *The ultra-metric space* $\langle \mathsf{TES}_{fin}/\simeq_{iso}, \mathbf{d} \rangle$ *is complete.*

PROOF. We show that each Cauchy sequence has a limit in $\mathsf{TES}_{fin}/\simeq_{iso}$ in the following way. Given an arbitrary Cauchy sequence $(\mathbf{E}_n)$: (i) we provide a recipe on how to construct a structure $\mathcal{F}$ that is (ii) a member of TES, is (iii) finitely approximable, and (iv) for which $\mathbf{d}(\mathbf{E}_n, \mathbf{E}_\mathcal{F}) \leqslant 2^{-n}$ for all $n \geqslant 1$.

We start with some preliminaries. Let $(\mathbf{E}_n)$ be a Cauchy sequence in $\mathsf{TES}_{fin}/\simeq_{iso}$. Assume that $\mathbf{d}(\mathbf{E}_n, \mathbf{E}_k) \leqslant 1/2^n$ for all $k \geqslant n \geqslant 1$.[8] Let $\mathcal{E}_n = (E_n, \rightsquigarrow_n, \mapsto_n, l_n, \mathcal{A}_n, \mathcal{R}_n, \mathcal{U}_n)$ be a representative of $\mathbf{E}_n$ and, for $k \geqslant n \geqslant 1$, $f_{n,k} : \mathcal{E}_n \restriction n \longrightarrow \mathcal{E}_k \restriction n$ an isomorphism. Let $\mathcal{E}_n \restriction n = (E_n \restriction n, \rightsquigarrow'_n, \mapsto'_n, l'_n, \mathcal{A}'_n, \mathcal{R}'_n, \mathcal{U}'_n)$. We assume w.l.o.g. $E_n \cap E_k = \varnothing$ if $n \neq k$. Let $E = \bigcup_{n \geqslant 1} E_n \restriction n$, and let $\equiv$ be the smallest equivalence relation on E that identifies e and $f_{n,k}(e)$ for all $e \in E_n \restriction n$ and $k \geqslant n$. Let $F = E/\equiv$ and $g_n : E_n \restriction n \longrightarrow F$ the canonical function that assigns to each $e \in E_n \restriction n$ its equivalence class under $\equiv$, $[e]_\equiv$, that is $\{ e, f_{n,n}(e), f_{n,n+1}(e), f_{n,n+2}(e), \ldots \}$. For $f \in F$ we define

$$rank(f) =_{df} min\{ n \geqslant 1 \mid \exists e \in E_n \restriction n : f = g_n(e) \}.$$

Stated in words, the rank of f is the minimal time instant such that f is the image of some event e under g_n. Let $F_n = \{ f \in F \mid rank(f) \leqslant n \}$, and for $rank(f) \leqslant n$, let $\pi_n(f)$ be the unique element in $E_n \restriction n$ with $g_n(\pi_n(f)) = f$ (the 'generator' of $[e]_\equiv$). Then,

- $\pi_n(g_n(e)) = e$ for all $e \in E_n \restriction n$,
- $f_{n,k}(\pi_n(f)) = \pi_k(f)$ for all $k \geqslant n \geqslant 1$ and $f \in F_n$,
- $l'_k(\pi_k(f)) = l'_k(f_{n,k}(\pi_n(f)) = l'_n(\pi_n(f))$ for all $k \geqslant n \geqslant 1$ and $f \in F_n$,
- $\mathcal{A}'_k(\pi_k(f)) \supseteq \mathcal{A}'_n(\pi_n(f))$ for all $k \geqslant n \geqslant 1$ and $f \in F_n$,

[8] From the theory of metric spaces [16] it is known that for any Cauchy sequence $(\mathbf{E}_n)$ there exists a subsequence $(\mathbf{E}_{i_n})$ with $\mathbf{d}(\mathbf{E}_{i_n}, \mathbf{E}_{i_k}) \leqslant 1/2^n$ for all $k \geqslant n \geqslant 1$. Moreover, the limit of $(\mathbf{E}_n)$ (if any) is identical to the limit of $(\mathbf{E}_{i_n})$.

- $\pi_n(f) \rightsquigarrow'_n \pi_n(f')$ iff $\pi_k(f) \rightsquigarrow'_k \pi_k(f')$ for all $k \geqslant n \geqslant 1$ and $f, f' \in F_n$.

For $Y \subseteq F$ let $\pi_n(Y) = \{ e \in E_n \upharpoonright n \mid g_n(e) \in Y \}$. Clearly, $\pi_n(Y) = \pi_n(Y \cap F_n)$ for all $Y \subseteq F$.

(i) We define $\mathcal{F} = (F, \rightsquigarrow, \mapsto, l, \mathcal{A}, \mathcal{R}, \mathcal{U})$ as follows.
- $f \rightsquigarrow f'$ iff $\pi_n(f) \rightsquigarrow'_n \pi_n(f')$ for all $n \geqslant \max\{rank(f), rank(f')\}$,
- $Y \mapsto f$ iff, for each $n \geqslant rank(f)$, $\pi_n(Y) \mapsto'_n \pi_n(f)$ is a bundle in $\mathcal{E}_n \upharpoonright n$,
- $l(f) = l'_n(\pi_n(f))$ for all $n \geqslant rank(f)$,
- $\mathcal{A}(f) = \bigcup_{n \geqslant rank(f)} \mathcal{A}'_n(\pi_n(f))$,
- $\mathcal{R}(Y, f) = \bigcup_{n \geqslant rank(f)} \mathcal{R}'_n(\pi_n(Y), \pi_n(f))$ for $Y \mapsto f$
- $\mathcal{U} = \bigcup_{n \geqslant 1} \{ g_n(e) \mid e \in \mathcal{U}'_n \}$.

Clearly, if $\pi_k(Y) \mapsto'_k \pi_k(f)$ and $rank(f) \leqslant n < k$ then

$$\pi_n(Y) = f_{n,k}^{-1}(\pi_k(Y) \upharpoonright n) \mapsto'_n f_{n,k}^{-1}(\pi_k(f)) = \pi_n(f)$$

and $\mathcal{R}'_n(\pi_n(Y), \pi_n(f)) \subseteq \mathcal{R}'_k(\pi_k(Y), \pi_k(f))$. Thus, $Y \mapsto f$ iff $\pi_n(Y) \mapsto'_n \pi_n(f)$ for infinitely many $n \geqslant rank(f)$ and

$$\mathcal{R}(Y, f) = \bigcup_{i \geqslant 1} \mathcal{R}'_{n_i}(\pi_{n_i}(Y), \pi_{n_i}(f))$$

for each sequence $n_1 < n_2 < \ldots$.

(ii) We now prove that $\mathcal{F} \in \mathsf{TES}$. It is easy to see that $\mathcal{F}$ satisfies constraint (P1) of Definition 1 and that $\rightsquigarrow$ as defined under (i) is irreflexive.

(P2) Let $Y \mapsto f$ be a bundle in $\mathcal{F}$, $f \in \mathcal{U}$ and either $f \rightsquigarrow f'$ or $f' \rightsquigarrow f$. If $\pi_n(Y) \rightsquigarrow_n \pi_n(f')$ in $\mathcal{E}_n$ for all $n \geqslant rank(f')$ then by the construction in (i), $Y \rightsquigarrow f'$, and (P2) is satisfied. Otherwise there is some $n_0 \geqslant rank(f')$ and some $e \in \pi_{n_0}(Y)$ with $e \not\rightsquigarrow_{n_0} \pi_{n_0}(f')$. For all $n \geqslant n_0$, $f_{n_0,n}(e) \in \pi_n(Y)$ and $f_{n_0,n}(e) \not\rightsquigarrow_n \pi_n(f')$ (†). For each $n \geqslant n_0$, we choose a bundle $X_n \mapsto_n \pi_n(f)$ in $\mathcal{E}_n$ with $X_n \upharpoonright n = \pi_n(Y)$ (which exists as $Y \mapsto f$, thus $\pi_n(Y) \mapsto'_n \pi_n(f)$ in $\mathcal{E}_n \upharpoonright n$). By (†) and (P2), it follows $X_n \mapsto_n \pi_n(f')$ for all $n \geqslant n_0$. (Note that $\pi_n(f) \in \mathcal{U}_n$.) Thus, $\pi_n(Y) \mapsto'_n \pi_n(f')$ is a bundle in $\mathcal{E}_n \upharpoonright n$. By definition of the bundle relation $\mapsto$ in $\mathcal{F}$, $Y \mapsto f'$.

(P3) Let $f \in \mathcal{U}$ such that $\mathcal{A}(f)$ consists of at least two elements. Since $\mathcal{A}'_n(\pi_n(f)) \subseteq \mathcal{A}'_{n+1}(\pi_{n+1}(f))$ there is some $n_0 \geqslant rank(f)$ such that $\mathcal{A}'_n(\pi_n(f))$ contains at least two elements for all $n \geqslant n_0$. By (P3), for each $n \geqslant n_0$, there is some $t_n \in \mathbb{R}^+$ and a bundle $X_n \mapsto_n \pi_n(f)$ in $\mathcal{E}_n$ with $\mathcal{R}_n(X_n, \pi_n(f)) = \{t_n\}$ [9] Thus,

(**) $X_n \upharpoonright n \mapsto'_n \pi_n(f)$ is a bundle in $\mathcal{E}_n \upharpoonright n$ with $\mathcal{R}'_n(X_n \upharpoonright n, \pi_n(f)) = \{t_n\}$.

By induction on n we define subsets Y_n of F_n and infinite sets I_n of natural numbers such that $I_0 \supseteq I_2 \supseteq \ldots$ and $Y_n = g_k(X_k \upharpoonright n)$ for all $k \in I_n$.

[9] The case $\mathcal{R}_n(X_n, \pi_n(f)) = \varnothing$ is not of interest here, since then event $\pi_n(f)$ would not be executable.

Let $I_0 = \{ n \mid n \geqslant n_0 \}$. We suppose that $n \geqslant 1$ and that $Y_1, \ldots, Y_{n-1}$ and $I_0, \ldots, I_{n-1}$ are already defined. As $F_n = g_n(E_n \restriction n)$ is finite (since $\mathcal{E}_n$ is finitely approximable) and $g_k(X_k \restriction n) \subseteq F_n$ for all $k \in I_{n-1}$ there exist $Y_n \subseteq F_n$ and an infinite subset I_n of I_{n-1} with $Y_n = g_k(X_k \restriction n)$ for all $k \in I_n$. Let

$$Y = \bigcup_{n \geqslant 1} Y_n.$$

Clearly, $Y_n = \{ f \in Y \mid \mathit{rank}(f) \leqslant n \} = Y \cap F_n$. Thus, $\pi_n(Y) = \pi_n(Y \cap F_n) = \pi_n(Y_n)$. We show that $Y \mapsto f$ is a bundle in $\mathcal{F}$ with $\mathcal{R}(Y, f) = \{t\}$ for some $t \in \mathbb{R}^+$.

Let $\mapsto_{k,n}$ and $\mathcal{R}_{k,n}$ be the bundle relation and bundle delay function of $\mathcal{E}_k \restriction n$ respectively. (Thus, $\mapsto'_n = \mapsto_{n,n}$ and $\mathcal{R}'_n = \mathcal{R}_{n,n}$.) Let $n \geqslant n_0$. We choose some $k \in I_n$ with $k \geqslant n$. Then,

$$X_k \restriction n = \pi_k(g_k(X_k \restriction n)) = f_{n,k}(\pi_n(Y)).$$

Since $X_k \mapsto_k \pi_k(f)$ we have $X_k \restriction n \mapsto_{k,n} \pi_k(f)$. As $f_{n,k}$ is an isomorphism $\mathcal{E}_n \restriction n \longrightarrow \mathcal{E}_k \restriction n$ and $\pi_k(f) = f_{n,k}(\pi_n(f))$ we obtain $\pi_n(Y) \mapsto'_n \pi_n(f)$. Thus, $Y \mapsto f$. Moreover, for all $n \geqslant n_0$, $\{t_{n_0}\} = \mathcal{R}'_{n_0}(\pi_{n_0}(Y), \pi_{n_0}(f)) = \mathcal{R}_{n,n_0}(X_n \restriction n_0, \pi_n(f)) \subseteq \mathcal{R}_n(X_n, \pi_n(f)) = \{ t_n \}$. Thus, $t_n = t_{n_0}$ for all $n \geqslant n_0$ and $\mathcal{R}(Y, f) = \{ t_{n_0} \}$.

(iii) As a next step we prove that $\mathcal{F}$ is finitely approximable. Let $(f_1, u_1) \ldots (f_k, u_k) \in \mathit{Traces}(\mathcal{F})$. Then, $(\pi_n(f_1), u_1) \ldots (\pi_n(f_k), u_k) \in \mathit{Traces}(\mathcal{E}_n \restriction n)$ for all $n > \mathit{max}\{ u_1, \ldots, u_k \}$. Vice versa, if $(e_1, u_1) \ldots (e_n, u_n) \in \mathit{Traces}(\mathcal{E}_n \restriction n)$ then $(g_n(e_1), u_1) \ldots (g_n(e_k), u_k) \in \mathit{Traces}(\mathcal{F})$. Thus, $\mathit{mintime}_{\mathcal{F}}(f) = \mathit{mintime}_{\mathcal{E}_n \restriction n}(\pi_n(f))$ for all $f \in F$ with $n \geqslant \mathit{rank}(f)$. Hence, $F \restriction n = \{ g_n(e) \mid e \in E_n \restriction n \}$. In particular, as $E_n \restriction n$ is finite, $F \restriction t$ is finite for all $t \geqslant 0$. Hence, $\mathcal{F}$ is finitely approximable.

(iv) Finally we show that $\mathcal{F}$ is a limit, or more precisely, that $\mathbf{E}_{\mathcal{F}}$ is the limit of the Cauchy sequence $(\mathbf{E}_n)$. It is easy to see that $f_n : E_n \restriction n \longrightarrow F \restriction n$, $f_n(e) = g_n(e)$, is an isomorphism $\mathcal{E}_n \restriction n \longrightarrow \mathcal{F} \restriction n$. We obtain $d(\mathcal{E}_n, \mathcal{F}) \leqslant 2^{-n}$ for all $n \geqslant 1$. Thus, $\mathbf{d}(\mathbf{E}_n, \mathbf{E}_{\mathcal{F}}) \leqslant 2^{-n}$ for all $n \geqslant 1$. Therefore, $\lim \mathbf{E}_n = \mathbf{E}_{\mathcal{F}}$.

5.5 Time-guardedness

We now give a metric denotational semantics for (a subset of) PA based on equivalence classes (under $\simeq_{iso}$) of timed event structures. With slight modifications we use the standard procedure (as explained in Section 5.2) to define a denotational semantics on complete metric spaces which is based on non-expansive/contracting semantic operators and Banach's fixed point theorem. 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 [7,29] guardedness ensures that each process instantiation

is preceded by an action-prefix, we use a notion of *time guardedness* (like in timed CSP [37]) which guarantees that a recursive process instantiation can only happen after a positive amount of time. In other words, time guardedness prevents a process instantiation to take place at time 0 like e.g. in $x + a_{[1,2)} \,.\, \mathbf{1}$ or $a_{[0,\infty)} \,.\, x$. Formally, the time guard of expression P is derived from the syntax of P and yields a lower bound for the minimal time instant where a process instantiation is possible. As a subsidiary notion we define the minimal time at which an expression can successfully terminate.

Definition 33 *(Minimal time of termination). Function* $\surd_{min} : \mathsf{Expr} \longrightarrow \mathbb{R}^+ \cup \{\infty\}$ *is defined by:*

$$
\begin{aligned}
\surd_{min}(\mathbf{0}) &=_{df} \infty \\
\surd_{min}(\mathbf{1}) &=_{df} 0 \\
\surd_{min}(x) &=_{df} 0 \\
\surd_{min}(a_I \,.\, P) &=_{df} inf(I) + \surd_{min}(P) \\
\surd_{min}(op\ P) &=_{df} \surd_{min}(P) \quad for\ op \in \{\, \backslash A, [\lambda] \,\} \\
\surd_{min}(P \,;\, Q) &=_{df} \surd_{min}(P) + \surd_{min}(Q) \\
\surd_{min}(P\ op\ Q) &=_{df} min\{\, \surd_{min}(P), \surd_{min}(Q) \,\} \quad for\ op \in \{\, +, [> \,\} \\
\surd_{min}(P \,||_A\, Q) &=_{df} max\{\, \surd_{min}(P), \surd_{min}(Q) \,\} \\
\surd_{min}(P \rhd_t Q) &=_{df} min\{\, \surd_{min}(P), t + \surd_{min}(Q) \,\}.
\end{aligned}
$$

Most of the rules are self-explanatory. For process variable x the minimal time of termination is supposed to be 'unknown' (as it depends on the declaration). Thus, we use 0 as the lower bound of the minimal time of termination for expressions of the form x. If $P \,;\, Q$ terminates successfully at time t, then t is of the form $t = t_P + t_Q$ where t_P is the time at which P has successfully terminated (thus, $t_P \geqslant \surd_{min}(P)$) and t_Q is the time at which Q can perform a successful termination event when started at time point 0 (thus, $t_Q \geqslant \surd_{min}(Q)$).

Example 34 *For instance, for the expression* $P \,;\, Q$ *where*

$$P \;=\; a_{[1,2]} \,.\, \Big(\; b_{[3,\infty)} \,.\, \mathbf{0} \,[>\, \mathbf{1} \;\Big) \;\; and \; Q \;=\; c_{[0,1]} \,.\, \mathbf{1}$$

we have

$$\surd_{min}(P \,;\, Q) = \surd_{min}(P) + \surd_{min}(Q) = (1 + min\{\, 3 + \infty, 0 \,\}) + 0 = 1.$$

The rule for $\surd_{min}(P \,||_A\, Q)$ is based on the fact that $P \,||_A\, Q$ can only perform a successful termination event if both components P and Q are ready to do so. Since $\surd_{min}(P)$ is derived from the syntax of P (rather than the semantics) we cannot expect that $\surd_{min}(P)$ yields the exact minimal termination time. For instance, for the expression $P = \; a_{[1,2]} \,.\, \mathbf{1} \,||_{\{a\}}\, b_{[1,5)} \,.\, \mathbf{1}$, we obtain

$\surd_{min}(P) = 1$ while P cannot terminate as its left component waits forever for the synchronisation on a. (So, the exact minimal termination time of P is ∞.)

By structural induction on terms we define the *time guard* of an expression. Intuitively, the time guard is the minimal time instant at which a process instantiation can take place. For instance, for an expression of the form $P\,;\,Q$ we distinguish between two kinds of process instantiations:

- a process instantiation that is in the scope of P which happens at the earliest at time $tg(P)$,
- a process instantiation that is in the scope of Q which happens at time $t{+}u$ where t is the time instant at which P performs a successful termination event (hence, $t \geqslant \surd_{min}(P)$) and u is the time at which Q (when started at time 0) instantiates the process (hence, $u \geqslant tg(Q)$).

For the expression $P = x \in \mathsf{Var}$ the process instantiation takes place at time 0. Thus, the time guard of x has to be defined as 0.

Definition 35 *(Time guard). Function* $tg : \mathsf{Expr} \longrightarrow \mathbb{R}^+ \cup \{\infty\}$ *is defined by:*

$$
\begin{aligned}
tg(\mathbf{0}) &=_{df} \infty \\
tg(\mathbf{1}) &=_{df} \infty \\
tg(x) &=_{df} 0 \\
tg(a_I\,.\,P) &=_{df} inf(I) + tg(P) \\
tg(op\ P) &=_{df} tg(P) \quad \textit{for } op \in \{\,\backslash A, [\lambda]\,\} \\
tg(P\ op\ Q) &=_{df} min\{\,tg(P), tg(Q)\,\} \quad \textit{for } op \in \{\,+, ||_A\,, [>\,\} \\
tg(P\,;\,Q) &=_{df} min\{\,tg(P), \surd_{min}(P) + tg(Q)\,\} \\
tg(P \rhd_t Q) &=_{df} min\{\,tg(P), t + tg(Q)\,\}.
\end{aligned}
$$

For declaration decl let $tg(decl) =_{df} inf\{\,tg(decl(x)) \mid x \in \mathsf{Var}\,\}$. *decl is called* time-guarded *iff* $tg(decl) > 0$.

Example 36 *For the expressions*

$$
\begin{aligned}
P_1 &= x + a_{[1,\infty)}\,.\,y \\
P_2 &= a_{[0,\infty)}\,.\,x \\
P_3 &= b_{(7,8]}\,.\,\mathbf{0} \rhd_5 (P_2\,||_{\{\,a\,\}}\,y) \\
P_4 &= c_{[2,3]}\,.\,(x\,[>\,b_{[1,\infty)}\,.\,\mathbf{1})
\end{aligned}
$$

we have:

$$\begin{aligned} tg(P_1) &= min\{\, tg(x), tg(a_{[1,\infty)} \,.\, y)\,\} \;=\; min\{\, 0, 1+0\,\} \;=\; 0,\\ tg(P_2) &= 0 + tg(x) \;=\; 0+0 \;=\; 0,\\ tg(P_3) &= min\{7, 5 + tg(P_2)\} \;=\; min\{\, 7, 5+0\,\} \;=\; 5,\\ tg(P_4) &= 2 + tg(x \,[\!>\, b_{[1,\infty)} \,.\, \mathbf{1}) \;=\; 2 \;+\; min\{\, 0, 1+0\,\} \;=\; 2. \end{aligned}$$

Thus, if $\mathsf{Var} = \{\, x, y\,\}$ *and* $decl_1(x) = P_3$, $decl_1(y) = P_4$, $decl_2(x) = P_1$, $decl_2(y) = \mathbf{0}$ *then*

$$tg(decl_1) \;=\; inf\{\, 5, 2\,\} \;=\; 2, \quad tg(decl_2) \;=\; inf\{\, 0, \infty\,\} \;=\; 0.$$

Hence, $decl_1$ *is time-guarded while* $decl_2$ *is not.*

Similarly to the observation we made for $\surd_{min}(\cdot)$, $tg(\cdot)$ is only a lower bound for the minimal time instant at which a process instantiation is possible rather than the exact time. For instance, for $P = a_{[1,2]} \,.\, x \,||_{\{\,a\,\}}\, b_{[1,5)} \,.\, \mathbf{1}$ we have $tg(P) = 1$, while the process instantiation x is never possible.

5.6 A metric semantics for TGPA

We give a metric semantics to TGPA, 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 $\mathcal{M} : \mathsf{TGPA} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$ 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 $\mathbf{E}, \mathbf{F} \in \mathsf{TES}_{fin}/\simeq_{iso}$:

$$\begin{aligned} op\ \mathbf{E} &=_{df} \mathbf{E}_{op\ \mathcal{E}} \quad \text{for } op \in \{\, a_I \,.\,, \backslash A, [\lambda]\,\} \text{ and}\\ \mathbf{E}\ op\ \mathbf{F} &=_{df} \mathbf{E}_{\mathcal{E}\ op\ \mathcal{F}} \text{ for } op \in \{\, +, \,;\,, \,||_A\,, [\!>, \rhd_t\,\} \end{aligned}$$

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

$$\mathcal{E}_\mathbf{1} =_{df} (\{\, e\,\}, \varnothing, \varnothing, \{\, (e, \surd)\,\}, \{\, (e, [0,\infty))\,\}, \varnothing, \varnothing). \tag{4}$$

Together with these semantic operators, $\mathsf{TES}_{fin}/\simeq_{iso}$ constitutes a PA-algebra. The following theorem states the non-expansiveness of the operators in our process algebra with respect to distance $\mathbf{d}$. Moreover, it shows that timed

prefixing is contracting (if $inf(I) \neq 0$) and that timeout is contracting in its second argument (if $t > 0$).

Theorem 37 *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) $\mathbf{d}(\mathbf{E} \rhd_t \mathbf{F}, \mathbf{E}' \rhd_t \mathbf{F}') \leqslant max\{\, \mathbf{d}(\mathbf{E}, \mathbf{E}'), 2^{-t} \cdot \mathbf{d}(\mathbf{F}, \mathbf{F}') \,\}$
(5) $\mathbf{d}(\mathbf{E}\,;\,\mathbf{F}, \mathbf{E}'\,;\,\mathbf{F}') \leqslant max\left\{\, \mathbf{d}(\mathbf{E}, \mathbf{E}'), 2^{-\surd_{min}(\mathbf{E})} \cdot \mathbf{d}(\mathbf{F}, \mathbf{F}') \,\right\}$

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

PROOF. Let $\mathcal{E}, \mathcal{E}', \mathcal{F}$ and $\mathcal{F}'$ be representatives of $\mathbf{E}$, $\mathbf{E}'$, $\mathbf{F}$ and $\mathbf{F}'$, respectively.

(1) It is easy to check that $mintime_{a_I \,.\, \mathcal{E}}(e) = mintime_{\mathcal{E}}(e) + inf(I)$ for $e \in \mathcal{E}$, since these events can only occur if the new event labelled a has occurred before, which causes a delay of at least $inf(I)$. So, if $\mathcal{E}$ and $\mathcal{E}'$ agree up to time u, say, then $a_I \,.\, \mathcal{E}$ and $a_I \,.\, \mathcal{E}'$ agree up to time $inf(I)+u$. That is,

$$d(a_I \,.\, \mathcal{E}, a_I \,.\, \mathcal{E}') = 2^{-inf(I)+u} = 2^{-inf(I)} \cdot 2^{-u} = 2^{-inf(I)} \cdot d(\mathcal{E}, \mathcal{E}').$$

(2) We consider $+$; the proofs for the other cases go along similar lines. Assume that $\mathcal{E}$ and $\mathcal{E}'$ agree up to time u and $\mathcal{F}$ and $\mathcal{F}'$ agree up to time v. From Definition 9 it is not difficult to see that $mintime_{\mathcal{E}+\mathcal{F}}(e) = mintime_{\mathcal{E}}(e)$ if $e \in \mathcal{E}$ and $mintime_{\mathcal{F}}(e)$ if $e \in \mathcal{F}$ [10]. So, $mintime_{\mathcal{E}+\mathcal{F}}(e) \leqslant min\{\, mintime_{\mathcal{E}}(e), mintime_{\mathcal{F}}(e) \,\}$. An analogous reasoning applies to $\mathcal{E}' + \mathcal{F}'$. This means that $\mathcal{E}+\mathcal{F}$ and $\mathcal{E}'+\mathcal{F}'$ agree at least up to time $min\{\, u, v \,\}$. But then we have:

$$d(\mathcal{E} + \mathcal{F}, \mathcal{E}' + \mathcal{F}') \leqslant max\{\, 2^{-u}, 2^{-v} \,\} = max\{\, d(\mathcal{E}, \mathcal{E}'), d(\mathcal{F}, \mathcal{F}') \,\}.$$

(3) Straightforward, since abstraction and relabelling do only change the labels of events and do not affect the timing of events.
(4) Easy from the definition of the timeout operator and the results for choice and prefix in this theorem.
(5) Assume $\mathcal{E}$ and $\mathcal{E}'$ agree up to time u and $\mathcal{F}$ and $\mathcal{F}'$ agree up to time v. Recall that $\surd_{min}(\mathcal{E})$ is the minimal time at which $\mathcal{E}$ can perform an event labelled with $\surd$. Since events in $\mathcal{F}$ can only occur after the occurrence of a $\surd$ in $\mathcal{E}$ we have that $mintime_{\mathcal{E}\,;\,\mathcal{F}}(e) = mintime_{\mathcal{E}}(e)$ if $e \in \mathcal{E}$ and equals $\surd_{min}(\mathcal{E}) + mintime_{\mathcal{F}}(e)$ if $e \in \mathcal{F}$. So, $mintime_{\mathcal{E}\,;\,\mathcal{F}}(e) \leqslant$

[10] Recall that $mintime_{\mathcal{E}}(e) = \infty$ if $e \notin \mathcal{E}$.

$min\{\, mintime_{\mathcal{E}}(e), \surd_{min}(\mathcal{E}) + mintime_{\mathcal{F}}(e)\,\}$. For $\mathcal{E}'\,;\,\mathcal{F}'$ we obtain a similar result. Now distinguish between (a) $\surd_{min}(\mathcal{E}) > u$ and (b) $\surd_{min}(\mathcal{E}) \leqslant u$. For these cases we have:

(a) $\surd_{min}(\mathcal{E}) > u$, or equivalently, $2^{-\surd_{min}(\mathcal{E})} < d(\mathcal{E},\mathcal{E}')$. Since $\mathcal{E}$ and $\mathcal{E}'$ agree up to time u it follows that $\surd_{min}(\mathcal{E}') > u$. An event of $\mathcal{F}$ (resp. $\mathcal{F}'$) can only happen after the successful termination of $\mathcal{E}$ (resp. $\mathcal{E}'$). From $\surd_{min}(\mathcal{E}) > u$ and $\surd_{min}(\mathcal{E}') > u$ it now follows that $\mathcal{E}\,;\,\mathcal{F}$ and $\mathcal{E}'\,;\,\mathcal{F}'$ agree at least up to time u. So, in this case $d(\mathcal{E}\,;\,\mathcal{F},\mathcal{E}'\,;\,\mathcal{F}') \leqslant d(\mathcal{E},\mathcal{E}')$, and hence

$$d(\mathcal{E}\,;\,\mathcal{F},\mathcal{E}'\,;\,\mathcal{F}') \leqslant max\left\{\, d(\mathcal{E},\mathcal{E}'), 2^{-\surd_{min}(\mathcal{E})} \cdot d(\mathcal{F},\mathcal{F}')\,\right\}.$$

(b) $\surd_{min}(\mathcal{E}) \leqslant u$, or equivalently, $2^{-\surd_{min}(\mathcal{E})} \geqslant d(\mathcal{E},\mathcal{E}')$. Since $\mathcal{E}$ and $\mathcal{E}'$ agree up to time u it follows $\surd_{min}(\mathcal{E}) = \surd_{min}(\mathcal{E}')$. Now distinguish between (i) $u \leqslant \surd_{min}(\mathcal{E}) + v$ and (ii) $u > \surd_{min}(\mathcal{E}) + v$. For case (i) we have that $\mathcal{E}\,;\,\mathcal{F}$ and $\mathcal{E}'\,;\,\mathcal{F}'$ agree at least up to time u, whereas for case (ii) they agree at least up to time $\surd_{min}(\mathcal{E}) + v$. So in this case,

$$d(\mathcal{E}\,;\,\mathcal{F},\mathcal{E}'\,;\,\mathcal{F}') \leqslant max\left\{\, d(\mathcal{E},\mathcal{E}'), 2^{-\surd_{min}(\mathcal{E})} \cdot d(\mathcal{F},\mathcal{F}')\,\right\}.$$

As a next step we prove that F_{decl} is contractive with respect to $\tilde{d}$ where $\tilde{d}$ is defined by $\tilde{d}(\phi_1,\phi_2) = sup\{\, \mathbf{d}(\phi_1(P),\phi_2(P)) \mid P \in \mathsf{Expr}\,\}$ for homomorphisms $\phi_1,\ \phi_2 : \mathsf{Expr} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$. In order to prove that F_{decl} is contracting we use the following two lemmata.

Lemma 38 *For homomorphism* $\phi : \mathsf{Expr} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$ *and* $P \in \mathsf{Expr}$*:*

$$\surd_{min}(\phi(P)) \geqslant \surd_{min}(P).$$

PROOF. Straightforward by structural induction on P.

Lemma 39 *For homomorphisms* $\phi_1,\ \phi_2 : \mathsf{Expr} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$ *and* $P \in \mathsf{Expr}$*:*

$$\mathbf{d}(\phi_1(P),\phi_2(P)) \ \leqslant\ 2^{-tg(P)} \cdot \tilde{d}(\phi_1,\phi_2).$$

PROOF. By induction on the structure of P.
Base: the cases $P = \mathbf{0}$ and $P = \mathbf{1}$ are straightforward, e.g. for $\mathbf{0}$ we have

$\mathbf{d}(\phi_1(\mathbf{0}),\phi_2(\mathbf{0}))$
$=$ $\{\ \phi_1$ and ϕ_2 are homomorphisms $\}$
$\mathbf{d}(\mathbf{E_0},\mathbf{E_0})$
$=$ $\{\ \langle \mathsf{TES}_{fin}/\simeq_{iso}, \mathbf{d}\rangle$ is an ultra-metric space $\}$

0.

If $P = x \in \mathsf{Var}$ then $tg(P) = 0$ and — by definition of $\tilde{d}$ — it follows $\mathbf{d}(\phi_1(P), \phi_2(P)) \leqslant \tilde{d}(\phi_1, \phi_2)$.
Induction Step: we illustrate this case for timed action-prefix and sequential composition; the proofs for the other cases are similar and are omitted here.

(1) Consider $P = a_I \,.\, Q$. Then we derive:

$$\begin{array}{ll} & \mathbf{d}(\phi_1(a_I \,.\, Q), \phi_2(a_I \,.\, Q)) \\ \leqslant & \{ \text{ Theorem 37; } \phi_1 \text{ and } \phi_2 \text{ are homomorphisms } \} \\ & 2^{-inf(I)} \cdot \mathbf{d}(\phi_1(Q), \phi_2(Q)) \\ \leqslant & \{ \text{ induction hypothesis } \} \\ & 2^{-inf(I)} \cdot 2^{-tg(Q)} \cdot \tilde{d}(\phi_1, \phi_2) \\ = & \{ \text{ definition of } tg \} \\ & 2^{-tg(P)} \cdot \tilde{d}(\phi_1, \phi_2). \end{array}$$

(2) Let $P = Q \,;\, R$. Then we derive:

$$\begin{array}{ll} & \mathbf{d}(\phi_1(Q \,;\, R), \phi_2(Q \,;\, R)) \\ \leqslant & \{ \text{ Theorem 37; } \phi_1 \text{ and } \phi_2 \text{ are homomorphisms } \} \\ & max\left\{\mathbf{d}(\phi_1(Q), \phi_2(Q)), 2^{-\surd_{min}(\phi_1(Q))} \cdot \mathbf{d}(\phi_1(R), \phi_2(R))\right\} \\ \leqslant & \{ \text{ Lemma 38 } \} \\ & max\left\{\mathbf{d}(\phi_1(Q), \phi_2(Q)), 2^{-\surd_{min}(Q)} \cdot \mathbf{d}(\phi_1(R), \phi_2(R))\right\} \\ \leqslant & \{ \text{ induction hypothesis (twice) } \} \\ & max\left\{2^{-tg(Q)} \cdot \tilde{d}(\phi_1, \phi_2), 2^{-(\surd_{min}(Q) + tg(R))} \cdot \tilde{d}(\phi_1, \phi_2)\right\} \\ = & \{ \text{ definition of } tg \} \\ & 2^{-tg(P)} \cdot d(\phi_1, \phi_2). \end{array}$$

Theorem 40 *For each decl and homomorphisms* ϕ_1, $\phi_2 : \mathsf{Expr} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$*:*

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

PROOF. By structural induction on P we show that

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

Base: for $P \in \{\, \mathbf{0}, \mathbf{1} \,\}$ the result follows directly. For case $P = x$ we derive:

$$\begin{array}{ll} & \mathbf{d}(F_{decl}(\phi_1)(x), F_{decl}(\phi_2)(x)) \\ = & \{ \text{ definition of } F_{decl} \} \end{array}$$

$$
\begin{array}{ll}
& \mathbf{d}(\phi_1(decl(x)), \phi_2(decl(x))) \\
\leqslant & \{ \text{ Lemma 39 } \} \\
& 2^{-tg(decl(x))} \cdot \tilde{d}(\phi_1, \phi_2) \\
\leqslant & \{ \ tg(decl) = inf\{ \ tg(decl(x)) \mid x \in \mathsf{Var} \ \} \ \} \\
& 2^{-tg(decl)} \cdot \tilde{d}(\phi_1, \phi_2)
\end{array}
$$

Induction Step: from Theorem 37 it follows

$$\mathbf{d}(\mathbf{E} \ op \ \mathbf{F}, \mathbf{E}' \ op \ \mathbf{F}') \ \leqslant \ max\{\mathbf{d}(\mathbf{E}, \mathbf{E}'), \mathbf{d}(\mathbf{F}, \mathbf{F}')\} \qquad (5)$$

for $op \in \{ +, \ ; \ , \ [>, \rhd_t, \ ||_A \ \}$. Using this result we derive:

$$
\begin{array}{ll}
& \mathbf{d}(F_{decl}(\phi_1)(P \ op \ Q), F_{decl}(\phi_2)(P \ op \ Q)) \\
\leqslant & \{ \ (5) \ \} \\
& max\{\mathbf{d}(F_{decl}(\phi_1)(P), F_{decl}(\phi_2)(P)), \mathbf{d}(F_{decl}(\phi_1)(Q), F_{decl}(\phi_2)(Q))\} \\
\leqslant & \{ \text{ induction hypothesis (twice) } \} \\
& 2^{-tg(decl)} \cdot \tilde{d}(\phi_1, \phi_2).
\end{array}
$$

A similar reasoning applies to the unary operators $\{ \ a_I \ . \ , \backslash A, [\lambda] \ \}$.

This result says that F_{decl} is contracting with contraction coefficient $2^{-tg(decl)}$ provided that $decl$ is time-guarded, that is, $tg(decl) > 0$. Thus, for time-guarded declaration $decl$, the higher-order function F_{decl} has a unique fixed point, say ϕ_{decl}. The metric semantics $\mathcal{M} : \mathsf{TGPA} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$ is now defined by $\mathcal{M}(decl, P) =_{df} \phi_{decl}(P)$.

6 A consistent operational interleaving semantics

Most timed process algebras are based on an interleaving semantics. In order to facilitate a comparison with these existing approaches and to investigate the 'compatibility' of our proposal with the standard interleaving semantics of LOTOS (in a sense which will be clarified later) we present an operational interleaving semantics for PA and investigate its relation to our metric semantics. We start by introducing the notions of timed transition system and (strong) timed bisimulation. Then we present the operational interleaving semantics of PA, after which we study the consistency between this interleaving and the non-interleaving semantics.

The notions of timed transition system and timed bisimulation, a timed variant of Milner's and Park's strong bisimulation are defined as follows (see also [33,25]).

Definition 41 *(Timed transition system.) A* timed transition system *is a quadruple* $(S, L, \longrightarrow, s_0)$ *with*

- S, *a non-empty set of states*
- $L \subseteq \mathsf{Act} \times \mathbb{R}^+$, *a set of labels*
- $\longrightarrow \subseteq S \times L \times S$, *a transition relation*
- $s_0 \in S$, *the initial state.*

We will write $p \xrightarrow{a,t} q$ *rather than* $(p,(a,t),q) \in \longrightarrow$.

Definition 42 *(Timed bisimulation.) Two equally labelled timed transition systems* $T_i = (S_i, L, \longrightarrow_i, s_{0i})$ *are timed bisimilar, denoted* $T_1 \sim T_2$, *if there exists a bisimulation, i.e. a relation* $\mathcal{R} \subseteq S_1 \times S_2$ *with* $(s_{01}, s_{02}) \in \mathcal{R}$ *and for which for all* $(p,q) \in \mathcal{R}$ *we have:*

(1) whenever $p \xrightarrow{a,t}_1 p'$ *for some* $p' \in S_1$ *then there exists some* $q' \in S_2$ *with* $(p',q') \in \mathcal{R}$ *and* $q \xrightarrow{a,t}_2 q'$, *and*
(2) whenever $q \xrightarrow{a,t}_1 q'$ *for some* $q' \in S_2$ *then there exists some* $p' \in S_1$ *with* $(p',q') \in \mathcal{R}$ *and* $p \xrightarrow{a,t}_2 p'$.

6.2 A timed interleaving semantics

The operational semantics defines a set of transition relations $\xrightarrow{a,t}$. Proposition $P \xrightarrow{a,t} P'$ denotes that P can perform action $a \in \mathsf{Act}$, at time t, and subsequently evolve into P'. Let $\longrightarrow$ be the smallest relation closed under all inference rules of Table 1.

Let $\mathsf{ut}(P)$ denote the set of time instants at which P can initially perform an urgent action. Let PA^+ denote PA including the auxiliary operators ${}^t[\ ]$ and ${}^t\{\ \}$.

Definition 43 *(Time to initial urgent event). Function* $\mathsf{ut} : \mathsf{PA}^+ \longrightarrow \mathcal{P}(\mathbb{R}^+ \cup \{\infty\})$ *is defined by:*

$$\mathsf{ut}({}^t[\,P\,]) =_{df} t + \mathsf{ut}(P)$$
$$\mathsf{ut}(P \ op \ Q) =_{df} \mathsf{ut}(P) \cup \mathsf{ut}(Q) \text{ for } op \in \{+, [>, ||_A\}$$

Table 1
Operational interleaving semantics for PA

$$\begin{array}{rcl}
 & \vdash & \mathbf{1} \xrightarrow{\surd,t} \mathbf{0} \\
t \in I & \vdash & a_I \,.\, P \xrightarrow{a,t} {}^{t}[\, P \,] \\
P \xrightarrow{a,t} P' & \vdash & {}^{t'}[\, P \,] \xrightarrow{a,t+t'} {}^{t'}[\, P' \,] \\
P \xrightarrow{a,t} P' \quad t \leqslant \mathsf{mt}(Q) & \vdash & P + Q \xrightarrow{a,t} P' \\
Q \xrightarrow{a,t} Q' \quad t \leqslant \mathsf{mt}(P) & \vdash & P + Q \xrightarrow{a,t} Q' \\
P \xrightarrow{a,t} P' \quad a \neq \surd & \vdash & P\,;\, Q \xrightarrow{a,t} P'\,;\, Q \\
P \xrightarrow{\surd,t} P' & \vdash & P\,;\, Q \xrightarrow{\tau,t} {}^{t}[\, Q \,] \\
P \xrightarrow{a,t} P' \quad (a \neq \surd \,\wedge\, t \leqslant \mathsf{mt}(Q)) & \vdash & P \,[\!>\, Q \xrightarrow{a,t} P' \,[\!>\, {}^{t}\{\, Q \,\} \\
P \xrightarrow{\surd,t} P' \quad t \leqslant \mathsf{mt}(Q) & \vdash & P \,[\!>\, Q \xrightarrow{\surd,t} P' \\
Q \xrightarrow{a,t} Q' \quad t \leqslant \mathsf{mt}(P) & \vdash & P \,[\!>\, Q \xrightarrow{a,t} Q' \\
P \xrightarrow{a,t} P' \quad t \geqslant t' & \vdash & {}^{t'}\{\, P \,\} \xrightarrow{a,t} {}^{t'}\{\, P' \,\} \\
P \xrightarrow{a,t} P' \quad a \notin A \cup \{\surd\} & \vdash & P \,||_A\, Q \xrightarrow{a,t} P' \,||_A\, Q \\
Q \xrightarrow{a,t} Q' \quad a \notin A \cup \{\surd\} & \vdash & P \,||_A\, Q \xrightarrow{a,t} P \,||_A\, Q' \\
P \xrightarrow{a,t} P' \;\wedge\; Q \xrightarrow{a,t} Q' \;\; a \in A \cup \{\surd\} & \vdash & P \,||_A\, Q \xrightarrow{a,t} P' \,||_A\, Q' \\
P \xrightarrow{a,t} P' \quad a \notin A & \vdash & P \setminus A \xrightarrow{a,t} P' \setminus A \\
P \xrightarrow{a,t} P' \quad a \in A & \vdash & P \setminus A \xrightarrow{\tau,t} P' \setminus A \\
P \xrightarrow{a,t} P' & \vdash & P[\lambda] \xrightarrow{\lambda(a),t} P'[\lambda] \\
P \xrightarrow{a,t'} P' \quad t' \leqslant t & \vdash & P \rhd_t Q \xrightarrow{a,t'} P' \\
t \leqslant \mathsf{mt}(P) & \vdash & P \rhd_t Q \xrightarrow{\tau,t} {}^{t}[\, Q \,] \\
P \xrightarrow{a,t} P' \quad \mathit{decl}(x) = P & \vdash & x \xrightarrow{a,t} P'
\end{array}$$

$$\begin{array}{rl}
\mathsf{ut}({}^{t}\{\, P \,\}) =_{\mathit{df}} & \{\, t' \in \mathsf{ut}(P) \mid t' \geqslant t \,\} \\
\mathsf{ut}(P\,;\, Q) =_{\mathit{df}} & \mathsf{ut}(P) \\
\mathsf{ut}(\mathit{op}\, P) =_{\mathit{df}} & \mathsf{ut}(P) \;\; \textit{for } \mathit{op} \in \{\, \setminus A, [\lambda] \,\} \\
\mathsf{ut}(P \rhd_t Q) =_{\mathit{df}} & \mathsf{ut}(P) \cup \{\, t \,\} \\
\mathsf{ut}(x) =_{\mathit{df}} & \mathsf{ut}(P) \;\; \textit{for } x := P.
\end{array}$$

For all other syntactical constructs let $\mathsf{ut}(P) =_{\mathit{df}} \varnothing$.

Let $\mathsf{mt}(P)$ abbreviate $\mathit{min}(\mathsf{ut}(P))$, where $\mathit{min}\,\varnothing$ equals ∞. In order to let ut be well-defined we require process instantiations to be guarded.

Process $\mathbf{1}$ can perform the successful termination action $\surd$ at any time t. $a_I \,.\, P$ can perform action a at time $t \in I$ while evolving into ${}^{t}[\, P \,]$. Process ${}^{t'}[\, P \,]$ can be considered as process P shifted t' time units in advance. That is, if P can

perform action a, say, at time t, then ${}^{t'}[\, P \,]$ can perform a at time $t{+}t'$. Note that ${}^{t'}[\, P \,]$ is only an auxiliary construct; it has no counterpart at the language level.

The rules for $P + Q$ are somewhat adapted since (initial) urgent events in P or Q can decide the choice. E.g., in $a_4 + (b_3 \rhd_2 Q)$, the time-out will occur at time 2, and resolve the choice in favour of Q. In general, if P performs an action at time t then $P + Q$ can perform the same provided that Q cannot perform a time-out at any time earlier, i.e., if $t \leqslant \mathsf{mt}(Q)$. By symmetry, a similar condition is obtained for Q performing an action. Similar conditions appear for $[>$, and $\rhd$.

The rules for ; are a straightforward extension of the rules for the untimed case except that in case P performs a successful termination action $\surd$ at time t, then $P\,;\, Q$ evolves into ${}^{t}[\, Q \,]$ rather than Q. This represents that t time units have passed before Q can start with its execution.

If P performs an action at t and evolves into P' then $P\,[> Q$ can do the same while evolving into $P'\,[> \; {}^{t}\{\, Q \,\}$. Process ${}^{t}\{\, Q \,\}$ behaves like Q except that it is unable to perform events before t. This ensures that Q cannot disrupt $P'\,[> Q$ by performing an action at time t', say, while P has performed an action at time $t > t'$. The other inference rules for disrupt are straightforward extensions of the rules for the untimed case.

The inference rule for ${}^{t'}\{\, P \,\}$ is that if P can perform an action at time t, then ${}^{t'}\{\, P \,\}$ can do so if $t \geqslant t'$. Note that ${}^{t'}\{\, P \,\}$ is — like ${}^{t'}[\, P \,]$ — an auxiliary operator that cannot be used by the specifier.

The rules for independent parallel composition, hiding, and relabelling are straightforward extensions of the untimed rules. Synchronisation can only take place when both participants can perform an equally labelled action whose label is in A (or equals $\surd$) at time t.

If P performs an action at time t', with $t' \leqslant t$, and evolves into P' then $P \rhd_t Q$ can do the same; in this case the possibility that Q happens is dropped, since P has performed an action before (or at) time t. At time t the time-out can happen and the resulting process is ${}^{t}[\, Q \,]$. This can only be done if $t \leqslant \mathsf{mt}(P)$, which ensures that the time-out is not performed if P can perform another time-out before t.

For expression P and declaration *decl* we denote by $\mathcal{O}(\mathit{decl}, P)$ the timed transition system obtained from the inference rules of Table 1, that is

$$\mathcal{O}(\mathit{decl}, P) =_{\mathit{df}} (\mathsf{TGPA}^{+}, \mathsf{Act} \times \mathbb{R}^{+}, \longrightarrow, P)$$

In order to assess the relationship between our timed event structure and the operationally defined interleaving semantics we first define an "interleaving view" of the true concurrency semantics (like in [6,26,29]) and prove that this perspective is timed bisimilar to the operational semantics.

Definition 44 *(Interleaving view on event structure semantics). The transition relation* $\longrightarrow \subseteq \mathsf{TES}_{fin}/\simeq_{iso} \times(\mathsf{Act} \times \mathbb{R}^+) \times \mathsf{TES}_{fin}/\simeq_{iso}$ *on timed event structures is defined by* $\mathcal{E} \xrightarrow{a,t} \mathcal{E}'$ *iff there exists some event* $e \in init(\mathcal{E})$ *such that*

(1) $l(e) = a$,
(2) $t \in \mathcal{A}(e)$,
(3) $\forall\, e' \in init(\mathcal{E}) \cap \mathcal{U} : (e \rightsquigarrow e' \ \vee \ e' \rightsquigarrow e) \ \Rightarrow \ t \leqslant \mathcal{A}(e')$, *and*
(4) $\mathcal{E}' = (E', \rightsquigarrow', \mapsto', l', \mathcal{A}', \mathcal{R}', \mathcal{U}')$ *with*

- $E' = E - \{\, e \,\}$
- $\rightsquigarrow' = \rightsquigarrow \ \cap (E' \times E')$
- $\mapsto' = (\mapsto - \{\, (X, e') \in \mapsto \mid e \in X \,\}) \ \cup \{\, (\varnothing, e') \mid e' \rightsquigarrow e \,\}$
- $l' = l \upharpoonright E'$
- $\mathcal{A}'(e') = \mathcal{A}(e') \cap \bigcap_{e \rightsquigarrow e'} [t, \infty) \cap \bigcap_{X \overset{I}{\mapsto} e', e \in X} t + I$
- $\mathcal{R}' = (\mathcal{R} \upharpoonright \mapsto') \cup \{\, ((\varnothing, e'), [0, \infty)) \mid \varnothing \mapsto' e' \,\}$
- $\mathcal{U}' = \mathcal{U} \cap E'$.

The interleaving semantics of $\mathcal{E}$*, denoted* $\mathcal{I}(\mathcal{E})$*, is defined as:*

$$\mathcal{I}(\mathcal{E}) =_{df} (\mathsf{TES}_{fin}/\simeq_{iso}, \mathsf{Act} \times \mathbb{R}^+, \longrightarrow, \mathcal{E}).$$

It is not difficult to check that in the above definition, the structure $\mathcal{E}'$ is indeed a timed event structure. We leave the proof of this fact to the interested reader.

Constraints (1) and (2) are straightforward. Constraint (3) checks whether there does not exist an initial urgent event that might prevent event e from happening at time t. This constraint is closely related to a similar condition in the definition of timed event trace, cf. Definition 5. The intuitive interpretation of constraint (4) is as follows. First, the event e labelled with a is removed from the set of events and the conflicts between the remaining events are retained. Each bundle $X \mapsto e'$ with $e \in X$ is removed, because the condition that this bundle poses, namely some event in X must have happened before e' can happen, has now been satisfied. Each event e' that is disabled by e cannot happen anymore, and is made impossible by introducing an empty bundle pointing to it.

In addition, the delay of an event e' which has a bundle pointing to it originating from event e has to be checked: if t plus the required relative time, I say, between e and e' is larger than the delay of e', e' should be postponed to (at least) $t+I$. Because this should hold for all bundles pointing to e' originating from e, the intersection of bundle delays is taken such that all required relative delays are satisfied. Finally, in order to enforce that the causal relation between e and e' induces a temporal precedence, the delay of e' becomes at least t in case $e \rightsquigarrow e'$.

Some example transitions of a timed event structure are depicted in Figure 4.

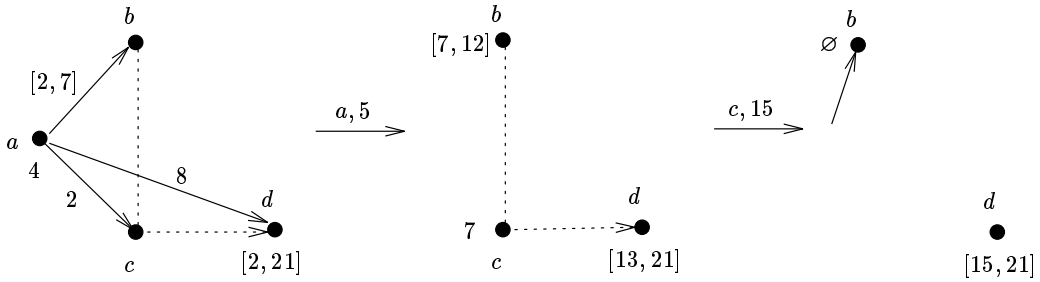


Fig. 4. Some example transitions for a timed event structure

Theorem 45 *(Consistency theorem). For any* $\langle decl, P\rangle \in \mathsf{TGPA}$:

$$\mathcal{I}(\mathcal{M}(decl, P)) \sim \mathcal{O}(decl, P).$$

PROOF. We provide the proof here for finite behaviours only; the proof for recursive behaviours can be provided in a similar way as the consistency proof provided in [7] for the untimed case. For finite behaviours we can consider $\mathcal{M}(P)$ and $\mathcal{O}(P)$, i.e. the declarations *decl* can be omitted, and prove that for $P \sim \mathcal{M}(P)$:

(1) if $P \xrightarrow{a,t} P'$ then $\exists\, \mathcal{E}' : \mathcal{M}(P) \xrightarrow{a,t} \mathcal{E}'$ and $P' \sim \mathcal{E}'$, and
(2) if $\mathcal{M}(P) \xrightarrow{a,t} \mathcal{E}'$ then $\exists\, P' : P \xrightarrow{a,t} P'$ and $\mathcal{E}' \sim P'$.

The proofs of both facts are by induction on the structure of P.

(1) *Base case:* for $P = \mathbf{0}$ the proposition follows easily, since $\mathbf{0}$ has no derivations. For $P = \mathbf{1}$ the only possible transition is labelled with $\surd, t$ for any t, while evolving into $\mathbf{0}$. It is easy to see from (4) and Definition 44 that $\mathcal{M}(P) = \mathcal{E}_{\mathbf{1}} \xrightarrow{\surd,t} \mathcal{E}_{\mathbf{0}}$ and that $\mathbf{0} \sim \mathcal{E}_{\mathbf{0}}$.
Induction step: consider the Q and R with $Q \sim \mathcal{M}(Q)$ and $R \sim \mathcal{M}(R)$ and assume the proposition holds for Q and R. We provide the proofs for prefix, timeout and disrupt. The proofs for the other cases are conducted

in a similar way. Let $\mathcal{M}(Q) = \mathcal{E}_Q = (E_Q, \rightsquigarrow_Q, \mapsto_Q, l_Q, \mathcal{A}_Q, \mathcal{R}_Q, \mathcal{U}_Q)$ and define $\mathcal{M}(R)$ and $\mathcal{M}(P)$ in a similar way.

(a) $P = a_I \,.\, Q$. Let $P \xrightarrow{a,t} P'$. Since prefixing has only one possible derivation for any $t \in I$, it follows $P' = {}^t[\, Q \,]$ and $t \in I$. From Definition 8 it follows that $\mathcal{M}(P)$ equals $\mathcal{M}(Q)$ where all events in E_Q are pointed to by a new conflict-free event e with $l_P(e) = a$ and $\mathcal{A}_P(e) = I$. By Definition 44 it follows that $\mathcal{M}(P) \xrightarrow{a,t} \mathcal{E}'$ for any $t \in I$. From the structure of $\mathcal{M}(P) = a_I \,.\, \mathcal{M}(Q)$ and $\mathcal{M}(P) \xrightarrow{a,t} \mathcal{E}'$ it follows that $\mathcal{E}'$ equals $\mathcal{M}(Q)$ where all events $e' \in E_Q$ have an event delay $\mathcal{A}_Q(e')+t$, the bundle delay of $\{\, e \,\} \mapsto e'$ plus the time of occurrence of e. Since $Q \sim \mathcal{M}(Q)$ it now follows $P' \sim \mathcal{E}'$.

(b) $P = Q \rhd_t R$. Let $P \xrightarrow{a,t'} P'$. According to the inference rules of Table 1 we have either

- $Q \xrightarrow{a,t'} Q'$ and $t' \leqslant t$. Then $P' = Q'$. From Definition 10 it follows that $\mathcal{M}(P)$ equals $\mathcal{M}(Q) + \hat{\tau}_{\{\, t \,\}} \,.\, \mathcal{M}(R)$. Let e be the new urgent event labelled with τ and delay t. From the structure of $\mathcal{M}(P)$ and Definition 44 it follows that any event of $\mathcal{M}(Q)$ can be performed with a delay smaller than t, the delay of the conflicting event e. From the induction hypothesis it follows $\mathcal{M}(Q) \xrightarrow{a,t'} \mathcal{E}'$ and $Q \sim \mathcal{E}'$. Since $P' = Q'$ it now follows $P' \sim \mathcal{E}'$.
- $t \leqslant \mathsf{mt}(Q)$. Then $P' = {}^t[\, R \,]$. The structure of $\mathcal{M}(P)$ is as described just above. It follows from Definition 44 that $\mathcal{M}(P)$ can execute the initial event e if there is no conflicting initial urgent event, e' say, with a delay smaller than t. From the structure of $\mathcal{M}(P)$ it follows that such event (if any) is in E_Q. It is straightforward to see that this condition on the execution of e corresponds to $t \leqslant \mathsf{mt}(Q)$. From the case for prefix we infer that $\hat{\tau}_{\{\, t \,\}} \,.\, \mathcal{M}(R) \xrightarrow{\tau,t} \mathcal{E}'$ where $\mathcal{E}'$ equals $\mathcal{M}(R)$ with all events having an event delay $\mathcal{A}_R(e')+t$. Since $R \sim \mathcal{M}(R)$ it now follows $P' \sim \mathcal{E}'$.

(c) $P = Q \,[\!> R$. Let $P \xrightarrow{a,t} P'$. According to the inference rules of Table 1 we have either

- $R \xrightarrow{a,t} R'$ and $t \leqslant \mathsf{mt}(Q)$. Then $P' = R'$. It follows from Definitions 14 and 44 that $\mathcal{M}(P) = \mathcal{M}(Q) \,[\!> \mathcal{M}(R)$ can execute an initial event of $\mathcal{M}(R)$ provided there is no conflicting urgent event in $\mathcal{M}(Q)$ that is forced to occur earlier. This condition corresponds to $t \leqslant \mathsf{mt}(Q)$. The proposition now follows directly from the induction hypothesis.
- $Q \xrightarrow{\surd,t} Q'$ and $t \leqslant \mathsf{mt}(R)$. Similar to the previous case.
- $Q \xrightarrow{a,t} Q'$ with $a \neq \surd$ and $t \leqslant \mathsf{mt}(R)$. Then $P' = Q' \,[\!> {}^t\{\, R \,\}$. From Definition 14 it follows that all initial events in $\mathcal{M}(R)$ are in conflict with any event in $\mathcal{M}(Q)$. $\mathcal{M}(P)$ can execute an initial event of $\mathcal{M}(Q)$ provided there is no conflicting urgent

event in $\mathcal{M}(R)$ that is forced to occur earlier. This condition corresponds to $t \leqslant \mathsf{mt}(R)$. Under this condition $\mathcal{M}(P) \xrightarrow{a,t} \mathcal{E}'$ where $\mathcal{E}'$ equals $\mathcal{M}(Q)\,[\!>\,\mathcal{E}$, where $\mathcal{E}$ is representing $\mathcal{M}({}^t\{\,R\,\})$. Since the event e labelled with a is in conflict with any initial event of $\mathcal{M}(R)$ it follows from Definition 44 that in $\mathcal{E}$ all the initial events of $\mathcal{M}(R)$ are postponed with t. Using this fact, and the fact that $Q \sim \mathcal{M}(Q)$ and $R \sim \mathcal{M}(R)$ it follows $P' \sim \mathcal{E}'$.

(2) By induction on the structure of P; similar to the proof of (1).

6.4 Consistency with a cpo-based semantics

We conclude this section with a brief comparison of our metric semantics and the cpo-based operational semantics $\mathcal{M}_{cpo}$ of Katoen et al. [23]. The formal relationship between our cpo and metric semantics is as follows. Let TES_{fin} be the set of timed event structures that are finitely approximable. For time-guarded $\langle decl, P\rangle$ it follows that $\mathcal{M}_{cpo}(decl, P)$ is finitely approximable. Function $f : \mathsf{TES}_{fin} \longrightarrow \mathsf{TES}_{fin}/\simeq_{iso}$ with $f(\mathcal{E}) =_{df} \mathbf{E}_{\mathcal{E}}$ is a homomorphism between the PA-algebras TES_{fin} and $\mathsf{TES}_{fin}/\simeq_{iso}$. Then, according to the results of [8], we obtain for any time-guarded process $\langle decl, P\rangle$: $f\,(\mathcal{M}_{cpo}(decl, P)) = \mathcal{M}(decl, P)$. This entails that the presented metric semantics is significantly more abstract than the cpo-based semantics of TGPA.

7 Concluding remarks

In this paper we have extensively studied the use of a metric denotational semantics for a real-time process algebra in a branching-time non-interleaving setting. This study can be seen as a continuation of the work of Loogen and Goltz in the setting of prime event structures for TCSP. In this untimed case the notion of distance is based on the number of discrete computation steps to which two prime event structures do agree. In our real-time setting a continuous version of this notion is adopted, and the distance is based on the amount of time to which two timed event structures do agree. Apart from some technical differences — like the restriction to executable events — that appeared due to the use of Langerak's bundle event structures rather than the more primitive prime event structures, we can conclude that the approach of Loogen and Goltz is well adaptable to the real-time case. Finally, we extended the consistency result between the prime event structure semantics and the operational semantics of (guarded) theoretical CSP to a consistency result between our timed event structure semantics and an operational interleaving semantics for our timed version of LOTOS. This consistency is defined in terms of a timed notion of strong bisimilarity.

Acknowledgements

The authors would like to thank Ed Brinksma and Rom Langerak for useful discussions on timed event structures. The anonymous referees are kindly acknowledged for their detailed comments and suggestions for improvement.

References

[1] S. Abramsky and A. Jung. Domain theory. In *Handbook of Logic in Computer Science*, Vol. 3, Clarendon Press, pages 1-168, 1994.

[2] L. Aceto and D. Murphy. Timing and causality in process algebra. *Acta Inf.*, **33**:317–350, 1996.

[3] R. Alur and D. Dill. A theory of timed automata. *Th. Comp. Sc.*, **126**:183–235, 1994.

[4] 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.

[5] C. Baier, J-P. Katoen and D. Latella. Metric semantics for true concurrent real time. In *Automata, Languages, and Programming — ICALP'98*, LNCS 1443, pages 568–580. Springer-Verlag, 1998.

[6] C. Baier and M.E. Majster-Cederbaum. Denotational semantics in the cpo and metric approach. *Th. Comp. Sci.*, **135**:171–220, 1994.

[7] C. Baier and M.E. Majster-Cederbaum. The connection between an event structure semantics and an operational semantics for TCSP. *Acta Inf.*, **31**:81–104, 1994.

[8] 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.

[9] C. Baier and M.E. Majster-Cederbaum. Metric semantics from partial order semantics. *Acta Inf.*, **34**:701–735, 1997.

[10] J.W. de Bakker and J.I. Zucker. Processes and the denotational semantics of concurrency. *Inf. and Contr.*, **54**(1/2):70–120, 1982.

[11] J.W. de Bakker and E.P. de Vink. *Control Flow Semantics.* MIT Press, 1996.

[12] J.W. de Bakker and E.P. de Vink. Denotational models for programming languages: applications of Banach's fixed point theorem. *Topology and its Applications*, **85**:35–52, 1998.

[13] T. Bolognesi and E. Brinksma. Introduction to the ISO specification language LOTOS. *Comp. Netw. & ISDN Syst.*, **14**:25–59, 1987.

[14] G. Boudol and I. Castellani. Flow models of distributed computations: three equivalent semantics for CCS. *Inf. and Comp.*, **114**:247–314, 1994.

[15] I. Castellani and G-Q. Zhang. Parallel product of event structures. *Th. Comp. Sc.*, **179**:203–215, 1997.

[16] E.T. Copson. *Metric Spaces.* Cambridge Tracts in Mathematics **57**, Cambridge University Press, 1992.

[17] J. Davies, J.W. Bryans and S.A. Schneider. Real-time LOTOS and timed observations. In *Formal Description Techniques VIII.* Chapmann & Hall, 1995.

[18] C.J. Fidge. A constraint-oriented real-time process calculus. In *Formal Description Techniques V*, pages 363–378. North-Holland, 1993.

[19] C.J. Fidge and J.J. Žic. A simple, expressive real-time CCS. In *Proc. 2nd Australasian Conf. on Parallel & Real-Time Systems*, pages 365–372, 1995.

[20] E. Goubault. Durations for truly-concurrent transitions. In *Programming Languages and Systems — ESOP'96*, LNCS 1058, pages 173–188. Springer-Verlag, 1996.

[21] W. Janssen, M. Poel, Q. Wu and J. Zwiers. Layering of real-time distributed processes. In *Formal Techniques in Real-Time and Fault-Tolerant Systems*, LNCS 863, pages 393–417. Springer-Verlag, 1994.

[22] J-P. Katoen. *Quantitative and Qualitative Extensions of Event Structures.* PhD thesis, University of Twente, 1996.

[23] 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.

[24] J-P. Katoen, R. Langerak, E. Brinksma, D. Latella and T. Bolognesi. A consistent causality-based view on a timed process algebra including urgent interactions. *Form. Meth. in Sys. Design*, **12**:189–216, 1998.

[25] A.S. Klusener. *Models and axioms for a fragment of real-time process algebra.* PhD thesis, Eindhoven University of Technology, 1993.

[26] R. Langerak. *Transformations and Semantics for LOTOS.* PhD thesis, University of Twente, 1992.

[27] R. Langerak. Bundle event structures: a non-interleaving semantics for LOTOS. In *Formal Description Techniques V*, pages 331–346. North-Holland, 1993.

[28] R. Langerak, E. Brinksma and J-P. Katoen. Causal ambiguity and partial orders in event structures. In *Concur'97: Concurrency Theory*, LNCS 1243, pages 317–332. Springer-Verlag, 1997.

[29] R. Loogen and U. Goltz. Modelling nondeterministic concurrent processes with event structures. *Fund. Inf.*, **14**(1):39–74, 1991.

[30] A. Maggiolo-Schettini and J. Winkowski. Towards an algebra for timed behaviours. *Th. Comp. Sci.*, **103**:335–363, 1992.

[31] A. Mazurkiewicz. Basic notions of trace theory. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, LNCS 354, pages 285–363. Springer-Verlag, 1989.

[32] D. Murphy. Time and duration in noninterleaving concurrency. *Fund. Inf.*, **19**:403–416, 1993.

[33] 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.

[34] M. Nielsen, G.D. Plotkin and G. Winskel. Petri nets, event structures and domains, part 1. *Th. Comp. Sc.*, **13**(1):85–108, 1981.

[35] M. Nivat. Infinite words, infinite trees, infinite computations. In *Foundations of Computer Science III*, Mathematical Centre Tracts **109**, pages 3-52, 1979.

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

[37] G.M. Reed and A.W. Roscoe. A timed model for Communicating Sequential Processes. *Th. Comp. Sc.*, **58**:249–261, 1988.

[38] A. Rensink. Posets for configurations! In *Concur'92: Concurrency Theory*, LNCS 630, pages 269–285. Springer-Verlag, 1992.

[39] F.W. Vaandrager. A simple definition for parallel composition of prime event structures. Report CS-R8903, Centre for Mathematics and Computer Science, 1989.

[40] G. Winskel. Event structure semantics for CCS and related languages. In *Automata, Languages and Programming — ICALP'82*, LNCS 140, pages 561–576. Springer-Verlag, 1982.

[41] J.J. Žic. Time-constrained buffer specifications in CSP+T and timed CSP. *ACM Trans. on Progr. Lang. and Sys.*, **16**(6):1661–1674, 1994.