Metric Semantics for True Concurrent Real Time

Christel Baier\textsuperscript{a}, Joost-Pieter Katoen\textsuperscript{b} and Diego Latella\textsuperscript{c}

\textsuperscript{a}Fakultät für Mathematik und Informatik, Universität Mannheim  
Seminargebäude A5, D-68159 Mannheim, Germany
\textsuperscript{b}Lehrstuhl für Informatik VII, Friedrich-Alexander-Universität Erlangen-Nürnberg  
Martensstrasse 3, D-91058 Erlangen, Germany
\textsuperscript{c}CNUCE Istituto del CNR, Via Santa Maria 36, l-56100 Pisa, Italy

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

1 Introduction

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

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

2 A real-time process algebra

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

$$P ::= 0 \mid 1 \mid a_1 . P \mid P + P \mid P \mid P\hat{>}_P \mid P \mid P A \mid P[\lambda] \mid P \triangleright_t P \mid x.$$  

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

- $1$ represents the successful termination process; it can only perform action
$\checkmark$ and then becomes $0$, the process that cannot perform any action.
- $a_1 . P$ denotes the prefix of $a$ and $P$ where $a$ is allowed (but not forced) to
occur at $t \in I$.
- $P ; Q$ denotes the sequential composition of $P$ and $Q$; the control is passed
to $Q$ by the termination of $P$ as indicated by the occurrence of $\checkmark$.
- $P \hat{>}_P Q$ denotes the disruption of $P$ by $Q$; i.e. $P$ may at any point of its
execution be disrupted by $Q$, unless $P$ has terminated.
- $P A Q$ denotes the parallel composition of $P$ and $Q$; $P$ and $Q$ execute actions
not in $A$ independently from each other, while actions in $A$ (and successful
termination actions) must be performed by both processes simultaneously.
- $P \triangleright_t Q$ 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 $P \hat{>}_P (0 \triangleright_t Q)$ specifies that $P$ is disrupted by $Q$ at time $t$, unless $P$ has
terminated before. Various case studies have proven that the timed operators like
$a_1 . P$ and $P \triangleright_t Q$ are convenient to specify practical real-time systems [1, 18].

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

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

The meaning of e ~ 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 ~ e' and e' ~ e is equivalent with e # e', the usual symmetric conflict in event structures. The reason for adopting ~ 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 → 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 → to denote the set of bundles of an event structure. The reason for not having a binary causality relation between events (as in prime event structures [10]) is to model parallel composition | Sentry in a less complex way.

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

**Definition 1.** A timed event structure (tes) E is a tuple (E, ~, →, l, A, R, U) with E, a set of events, ~ ⊆ E × E, the (irreflexive) conflict relation, → ⊆ P(E) × E, the bundle relation, l : E → Act, the labelling function, A : E → P(IR^+ ∪ {∞}), the event delay function, R : → → P(IR^+ ∪ {∞}), the bundle delay function, and U ⊆ E, the set of urgent events, such that:

(P1) \((X × X) \setminus \text{Id}_E \subseteq ~\) for any bundle set X

and for all e ∈ U:

(P2) ∀ e' ∈ E, X ⊆ E : ((e' ~ e ∨ e ~ e') ∧ X → e) ⇒ (X → e' ∨ X ~ e')

(P3) ∃ t ∈ IR^+ : (\(\{(X, e) ∈ X \mid X → e ∧ R(X, e) ∈ \{∅, \{t\}\}\}\) ∨ (3 X : X → e ∧ R(X, e) ∈ \{∅, \{t\}\}))

Here, P(·) denotes the power-set function, X ~ e' denotes (∀ e'' ∈ X : e'' ~ e') and \(\text{Id}_E\) denotes the identity relation on set E. Note that ∅ ~ e' for all e'.

Event structures are depicted as follows. Events are denoted as dots; near the dot the action label is given. If no confusion arises we often use action labels rather than event identities to denote events, e ~ e' is indicated by a dotted arrow from e to e'; if also e' ~ e, then a dotted line is drawn instead. A bundle X → 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 \epsilon \text{ with } \mathcal{R}(X, \epsilon) = 1 \text{ is denoted by } X \overset{I}{\mapsto} \epsilon. \text{ Delays } [t, \infty) \text{ are simply denoted by } t; \text{ delays } [0, \infty) \text{ are usually omitted. Figure 1(a) shows an example tes with e.g. } \{a\}^{[0,7]} \text{ and } \{a\}^{[0,5]} \text{.}

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

\text{en}^\mathcal{E}(\sigma) = \left\{ \epsilon \in E \mid \epsilon \not\in \sigma \land (\forall \epsilon_i \in \sigma : \epsilon \neq \epsilon_i) \land (\forall X \mapsto \epsilon : X \cap \sigma \neq \emptyset) \right\}.

The time instants at which an enabled event in \mathcal{E} after \sigma = (\epsilon_1, t_1) \ldots (\epsilon_n, t_n) could potentially happen is determined as

\text{time}^\mathcal{E}(\epsilon) \triangleq \mathcal{A}(\epsilon) \cap \bigcap_{\epsilon_i \mapsto \epsilon} [t_i, \infty) \cap \bigcap_{X \mapsto \epsilon, \epsilon_i \in X} t_i + 1.

**Definition 2.** \(\sigma = (\epsilon_1, t_1) \ldots (\epsilon_n, t_n)\) with \(\epsilon_i \in E\) (all events being pairwise distinct) and \(t_i \in \mathbb{R}^+\), is a timed event trace of \(\mathcal{E} \in \text{TES}\) iff for all \(0 < i \leq n:\)

1. \(\epsilon_j \sim \epsilon_i \Rightarrow (j < i \land t_j \leq t_i)\) for all \(0 < j \leq n\)
2. \(X \mapsto \epsilon_i \Rightarrow (\exists j : X \cap \{\epsilon_1, \ldots, \epsilon_{i-1}\} = \{\epsilon_j\} \land t_i \in t_j + 1)\) for all \(X \subseteq E\)
3. \(t_i \in \mathcal{A}(\epsilon_i)\)
4. \((\epsilon_i \sim \epsilon \lor \epsilon \sim \epsilon_i) \Rightarrow t_i \leq \min(\text{time}^\mathcal{E}_1(\epsilon_1 \ldots \epsilon_{i-1}), \epsilon)\) for \(\epsilon \in \mathcal{U} \cap \text{en}^\mathcal{E}(\epsilon_1 \ldots \epsilon_{i-1})\).

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

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

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

\[(a, t_a) (c, c) (b, t_b) \text{ if } 0 \leq t_a \leq t_c \leq t_b \land t_b \leq t_a + 4 \land t_c \leq t_a + 4\]
\[(a, t_a) (\tau, \tau) (d, t_d) \text{ if } 0 \leq t_a \leq \tau \leq t_d \land \tau = t_a + 4.\]

\(^1\) Often the set of events of a sequence is identified with the sequence itself.
Note that Figure 1(a) models a typical timeout scenario: if after the occurrence of a neither b nor c happen within 4 time units, then a timeout (τ) is forced to occur. It τ would not be urgent, the conditions for τa and τb in the first case would be τb ≤ τa+7 and τc ≤ τa+5, since τ is not forced to occur and time does not resolve the choice.

4 Operators for timed event structures

In this section we present some operators on timed event structures that are needed to define a compositional semantics for PA. They are basically adopted from [9, 10]. We start with some basic notions. Let Events be a set such that for all actions a ∈ Act there is an event e ∈ Events, and (ii) if e ∈ Events then (e, e) ∈ Events, and (ii) if e, e′ ∈ Events then (e, e′) ∈ Events. Let TES denote the set of tes’s E with E ⊆ Events. Let init(E) be the set of initial events of E and exit(E) its set of successful termination events, i.e. init(E) = {e ∈ E | ¬(E ⊆ E : X 7→ e)} and exit(E) = {e ∈ E | l(e) = √}.

In the rest of this section let E ∈ TES and E1 = (E1, ∼1, →1, l1, A1, R1, U1), E2 = (E2, ∼2, →2, l2, A2, R2, U2) such that w.l.o.g. E1 ∩ E2 = ∅. Let τ denote the urgent variant of τ.

Definition 3. For a ∈ Obs \ {τ, τ} and l ⊆ [0, ∞), aI : E1, E2, l. E1 ∪ {e_a}, ∼_1 , →_1 , l. E1 where for e ∉ E1

- → = →_1 \ ∪ \ {e} \ × E1
- A = (e, l) \ × (E1 \ × \ {[0, ∞)})
- R = R1 \ ∪ \ {((e), e'), A(e)} | e' ∈ E1}
- U = if a = τ then U1 \ ∪ \ {e} else U1.

τ . E denotes the prefixing of τ and E where e is declared to be urgent. The possibility τ . E set l must be either empty or equal to [t, t] for some t in order to guarantee axiom (P3).

Definition 4. E1 + E2 = (E1 ∪ E2, ∼1 ∪ ∼2, →1 ∪ →2, l1 ∪ l2, A1 ∪ A2, R1 ∪ R2, U1 ∪ U2) where ∼ = ∼_1 ∪ ∼_2 ∪ (init(E1) × init(E2)) ∪ (init(E2) × init(E1)).

Definition 5. Let A ⊆ Obs. Then E ∖ A = (E, ∼, →, l, A, R, U) where (l(e) ∈ A ⇒ l'(e) = τ) ∧ (l(e) /∈ A ⇒ l'(e) = l(e)).

Definition 6. For λ : Act → Act with λ(τ) = τ and λ(√) = √ let E[λ] = (E, ∼, →, λ ∘ l, A, R, U).

Definition 7. E1 ↓ E2 = (E1 ∪ E2, ∼1 ∪ ∼2, →1 ∪ →2, l, A, R, U1 ∪ U2) where

- → = →_1 \ ∪ \ ∼_2 \ ∪ \ (exit(E1) \ × \ exit(E1)) \ ∩ \ Id_{E1}
- → = →_1 \ ∪ \ →_2 \ ∪ \ (exit(E1) \ × \ E2)
- l = (l1 \ ∪ \ l2) \ ∩ \ (exit(E1) \ × \ \{√\}) \ ∪ \ (exit(E1) \ × \ \{τ\})
- A = A1 \ ∪ \ (E2 \ × \ {[0, ∞)})
As an example of how $\mathcal{E}_1; \mathcal{E}_2$ is computed consider:

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

**Definition 8.** $\mathcal{E}_1 \mathbin{||}_4 \mathcal{E}_2 \triangleq (E_1 \cup \mathcal{E}_2, 0, \rightarrow, 0, l_1 \cup l_2, A_1 \cup A_2, R_1 \cup R_2, U_1 \cup U_2)$ where $\rightarrow = \rightarrow_1 \cup \mathcal{E}_2$ and if any of their components are in conflict, or if different events have a common component different from $*$ (such events appear if two or more events in one tes synchronise with the same event in the other tes). A bundle is introduced iff when we take the projection on the component $\mathcal{E}_i$ of the bundle-set we obtain a bundle in $\rightarrow_i$. Let for $A \subseteq \text{Obs}$, $E_i \triangleq \{ e \in E \mid l_i(e) \in A \cup \{ \} \}$ be the set of synchronising events and $E_i^f \triangleq E_i \setminus E_i^b$ the set of ‘free’ events.

**Definition 9.** Let $A \subseteq \text{Obs}$. Then $\mathcal{E}_1 \mathbin{||}_4 \mathcal{E}_2 \triangleq (E, \rightarrow, 0, l, A, R, U)$ where

- $E = (E_1^f \times \{ * \}) \cup (\{ \} \times E_2^f) \cup \{ (e_1, e_2) \in E_1^f \times E_2^f \mid l_1(e_1) = l_2(e_2) \}$
- $(e_1, e_2) \sim (e'_1, e'_2)$ iff
  - $(e_1 \sim_1 e'_1) \lor (e_2 \sim_2 e'_2)$
  - $(e_1 = e'_1 \neq * \land e_2 = e'_2 \neq * \land e_1 \neq e'_1)$
- $X \rightarrow (e_1, e_2)$ iff
  - $(\exists X_1 : X_1 \rightarrow_1 e_1 \land X = \{ (e, e') \in E \mid e \in X_1 \})$ or
  - $(\exists X_2 : X_2 \rightarrow_2 e_2 \land X = \{ (e, e') \in E \mid e' \in X_2 \})$
- $l(e_1, e_2) = e$ then $l_2(e_2) = l_1(e_1)$
- $A(e_1, e_2) = A_1(e_1) \cap A_2(e_2)$ with $A_4(*) = [0, \infty)$.
- $R(X_1, e_1, e_2) = \bigcap_{X_1 \subseteq E_1} R_1(X_1, e_1) \cap \bigcap_{X_2 \subseteq E_2} R_2(X_2, e_2)$ with
  - $S_1 = \{ X_1 \subseteq E_1 \mid X_1 \rightarrow_1 e_1 \} \land X = \{ (e, e') \in E \mid e \in X_1 \}$ and
  - $S_2 = \{ X_2 \subseteq E_2 \mid X_2 \rightarrow_2 e_2 \} \land X = \{ (e, e') \in E \mid e' \in X_2 \}$
- $(e_1, e_2) \in U$ iff $e_1 \in U_1 \lor e_2 \in U_2$ with $* \notin U_i$.

Parallel composition is illustrated by the following example where the left-hand tes is composed with the empty tes:
For \( \mathcal{E}_1 \bowtie_t \mathcal{E}_2 \) a new urgent event \( e \) with delay \([t, t]\) is introduced that models the expiration of the timer. Since either the timer expires or \( \mathcal{E}_1 \) performs an initial event before (or at) \( t \), \( e \) is put in mutual conflict with all initial events of \( \mathcal{E}_1 \).

**Definition 10.** For \( t \in [0, \infty) \) let \( \mathcal{E}_1 \bowtie_t \mathcal{E}_2 \triangleq \mathcal{E}_1 + \tilde{\tau}_{[0, t]} \cdot \mathcal{E}_2 \).

By straightforward proof one can establish that TES is closed under the operators \( a_I, +, \land, [\lambda], \lor, [\geq], \|A\|, \) and \( \bowtie_t \).

## 5 A metric denotational semantics

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

\[
\begin{align*}
F_{\text{decl}}(\phi)(0) & \triangleq \phi' \\
F_{\text{decl}}(\phi)(1) & \triangleq \phi' \\
F_{\text{decl}}(\phi)(x) & \triangleq \phi(\text{decl}(x)) \\
F_{\text{decl}}(\phi)(\text{op } P) & \triangleq \text{op}' F_{\text{decl}}(\phi)(P) \\
F_{\text{decl}}(\phi)(P \text{ op } Q) & \triangleq F_{\text{decl}}(\phi)(P) \text{ op}' F_{\text{decl}}(\phi)(Q)
\end{align*}
\]

for \( \text{op} \in \{ a_I, +, [\lambda], \lor, [\geq], \|A\|, \} \) and so on, are non-distance increasing on \( (S, d) \) and contracting in certain arguments.

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

**Definition 11.** The *truncation of \( \mathcal{E} \) up to \( t \in \mathbb{R}^+ \) is defined by* \( \mathcal{E} \upharpoonright t \triangleq (E \upharpoonright t, \sim_{\mathcal{E}} \upharpoonright t, l_{\mathcal{E}}, \mathcal{A}_{\mathcal{E}}, R_{\mathcal{E}}, Q_{\mathcal{E}} \) where \( l_{\mathcal{E}}(e) \equiv l(e), \mathcal{A}_{\mathcal{E}}(e) \equiv \mathcal{A}(e) \cap [0, t), U_{\mathcal{E}} \equiv U \upharpoonright t \), and \( \sim_{\mathcal{E}} \equiv \sim \upharpoonright (E \\times E \\times t) \).
It is not difficult to check that for $\mathcal{E} \in \mathrm{TES}$ we have $\mathcal{E} \upharpoonright t \in \mathrm{TES}$, for all $t \in \mathbb{IR}^+$. 

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

Events $e_b, e_f$ and $e_g$ are eliminated since the minimal time at which they can occur, time 11, 8 and 6, respectively, is at least 6. Note that $\{e_a\}^{[1,2]}_t = e_c$ for $t=6$, since $\{e_a\}^{[1,2]}_t = e_c$ and $[1, \infty) \cap [0, 9] = [1, 9]$.

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

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

$$d(E_1, E_2) = \inf \{2^{-t} \mid t = E_1 \upharpoonright t = E_2 \upharpoonright t\}.$$  

(2)

Remark that $\mathcal{E} \upharpoonright 0$ is the empty tes, so each pair of tes’s agrees at least up to time 0. Although this basic idea is rather intuitive, it is, unfortunately, too naive. The problem is that some distinct tes’s cannot be distinguished according to $d$. This means that $d$ is a pseudo-metric rather than a metric. For instance, the tes consisting of a single event $\varepsilon$ with an empty bundle pointing to $\varepsilon$ is indistinguishable from the empty tes, since their time truncations are all empty. That is, according to (2) their distance is 0. The problem with this example is that tes’s may contain events that can never appear. Such events can, for instance, appear in the semantics for expressions that contain circular causal dependencies, like in $a \cdot b \cdot 0 \vdash [a,b] \cdot b \cdot a \cdot 0$, or timing constraints that avoid certain actions from happening, like in $a_2 \cdot 0 \geq_1 b \cdot 0$ where $a$ will never happen. (Such events can be removed by applying the transformations exposed in [11, 9] that preserve timed event traces.)

A solution to this problem is to impose an equivalence relation, $\simeq$ say, on TES, while aiming at $d(E_1, E_2) = 0 \Leftrightarrow E_1 \simeq E_2$. Stated in other words, where $d$ is the equivalent of $d$ on TES/$\simeq$ and $E_i$ denotes the equivalence class of $E_i$, under $\simeq$, we aim at $d(E_1, E_2) = 0 \Leftrightarrow E_1 = E_2$. In order to obtain $\simeq$, the example suggests to abstract from events that can never be executed:

**Definition 13.** The normal form of $\mathcal{E}$, denoted $\text{NF} (\mathcal{E})$, is defined as $\text{NF} (\mathcal{E}) \triangleq (E \upharpoonright \omega, \sim_{\omega}, \rightarrow_{\omega}, i_{\omega}, A_{\omega}, R_{\omega}, U_{\omega})$ where $i_{\omega}(\varepsilon) = i(\varepsilon)$, $A_{\omega}(\varepsilon) = A(\varepsilon)$, $U_{\omega} = U \cap (E \upharpoonright \omega)$, $\sim_{\omega} = \sim \cap (E \upharpoonright \omega \times E \upharpoonright \omega)$ and
For \( \mathcal{E} \in \text{TES} \) it follows by straightforward verification that \( \text{NF}(\mathcal{E}) \in \text{TES} \).

**Lemma 14.** \( \text{Traces}(\text{NF}(\mathcal{E})) = \text{Traces}(\mathcal{E}) \). As in the untimed case \([12]\) the metric approach also allows to abstract from the names of the events, i.e., to deal with isomorphism classes of tes’s. The names of the events are only needed for technical reasons but they are meaningless for the semantics of a PA-process. The advantage of abstraction from event names is that the definitions of operators like +, \( \geq \), and so on, become less awkward.

**Definition 15.** \( \mathcal{E}_1 \equiv (E_1, \sim_1, \rightarrow_1, l_1, A_1, \mathcal{R}_1, \mathcal{U}_1) \) for \( i = 1, 2 \) are isomorphic if there exists a bijection \( f : E_1 \ni \omega \to E_2 \ni \omega \) such that \( l_2 \circ f = l_1 \), \( A_2 \circ f = A_1 \) and

1. \( \epsilon_1 \sim_1 \epsilon_2 \) iff \( f(\epsilon_1) \sim_2 f(\epsilon_2) \) for all \( \epsilon_1, \epsilon_2 \in E_1 \ni \omega \\
2. \( X \rightarrow_1 \epsilon \) iff \( f(X) \rightarrow_2 f(\epsilon) \) for all \( \epsilon \in E_1 \ni \omega, X \subseteq E_1 \ni \omega, \) and
3. \( \epsilon \in \mathcal{U}_1 \ni \omega \) iff \( f(\epsilon) \in \mathcal{U}_2 \).

Let \( \mathcal{E}_1 \sim_{\approx_{\omega}} \mathcal{E}_2 \) iff there exists an isomorphism from \( \mathcal{E}_1 \) to \( \mathcal{E}_2 \). Note that \( \mathcal{E} \sim_{\approx_{\omega}} \text{NF}(\mathcal{E}) \).

For \( \mathcal{E} \in \text{TES} \) let \( \text{E}_\mathcal{E} \) denote the equivalence class of \( \mathcal{E} \) under \( \approx_{\omega} \). For \( \text{E} \in \text{TES} / \approx_{\omega} \) let \( \text{E} \ni t \triangleq \text{E}_\mathcal{E} \), where \( \mathcal{E} \) is a representative of \( \text{E} \). The distance between equivalence classes (under \( \approx_{\omega} \)) of tes’s is given by:

\[
d(\text{E}_1, \text{E}_2) \triangleq \inf \{ 2^{-t} \mid \text{E}_1 \ni t = \text{E}_2 \ni t \}. \tag{3}
\]

Remark that \( d(\text{E}, \text{E} \ni t) \leq 2^{-t} \) for all \( t \geq 0 \).

**Example 2.** Let \( \mathcal{E}_i = (E_i, \sim, \rightarrow, E_i \times \{ a \}, A_i, \mathcal{R}_i, \mathcal{U}_i) \), for \( i = 1, 2 \) where

- \( E_1 = \{ (k, j) \mid j \geq 1, 0 < k < j \} \) and \( E_2 = E_1 \cup \{ (k, 0) \mid k \geq 1 \} \)
- \( \{ (k, j) \} \rightarrow_1 (k+1, j) \) for \( 0 < k < j \) and \( \{ (k, 0) \} \rightarrow_2 (k+1, 0) \) for \( k \geq 1 \)
- \( A_i(k, j) = [k, k] \) for all \( (k, j) \in E_i \), and
- \( \mathcal{R}_i \{ (k, j), (k+1, j) \} = [0, 1] \).

Then, \( \mathcal{E}_1 \not\sim_{\approx_{\omega}} \mathcal{E}_2 \) while \( \mathcal{E}_1 \ni t \sim_{\approx_{\omega}} \mathcal{E}_2 \ni t \) for all \( t \geq 0 \). If we now define \( d \) as suggested in (3) on \( \text{TES} / \approx_{\omega} \) then \( d(\text{E}_1, \text{E}_2) = 0 \), although \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) are not isomorphic, thus yielding a pseudo-metric.

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

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

Let \( \text{TES}_{\text{fin}} / \approx_{\omega} \) denote the isomorphism classes of finitely approximable tes’s. The main result that we need in order to define a metric semantics for PA is:
Theorem 17. \( \langle \text{TES}_{\text{fin}}/\simeq_{\alpha}, d \rangle \) is a complete ultra-metric space.

A metric semantics for PA. We now give a metric semantics for (a subset of) PA based on equivalence classes (under \( \simeq_{\alpha} \)) of tes’s. The main difference with the standard (untimed) case is the notion of ‘guardedness’ which ensures the well-definedness of recursive programs. While in the untimed case [12] guardedness requires that each process instantiation is preceded by an action we use a notion of *time guardedness* (like in timed CSP [17]) which requires that a recursive process instantiation can only happen after a positive amount of time.

Let functions \( \sqrt{\min} : \text{Expr} \to [0, \infty) \) and \( t_g : \text{Expr} \to [0, \infty) \) be defined as in Table 1. For declaration \( \text{decl} \) let \( t_g(\text{decl}) = \inf \{ t_g(\text{decl}(x)) \mid x \in \text{Var} \} \). \( \text{decl} \) is called *time-guarded* iff \( t_g(\text{decl}) > 0 \). We give a metric semantics to TGP A,

<table>
<thead>
<tr>
<th>( P )</th>
<th>( \sqrt{\min}(P) )</th>
<th>( t_g(P) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \emptyset )</td>
<td>( \infty )</td>
<td>( \infty )</td>
</tr>
<tr>
<td>( 1 )</td>
<td>0</td>
<td>( \infty )</td>
</tr>
<tr>
<td>( x )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( a_t \cdot P )</td>
<td>( \inf(I) + \sqrt{\min}(P) )</td>
<td>( \inf(I) + t_g(P) )</td>
</tr>
<tr>
<td>( P[l], P \setminus A )</td>
<td>( \sqrt{\min}(P) )</td>
<td>( t_g(P) )</td>
</tr>
<tr>
<td>( P + Q, P \triangleright Q )</td>
<td>( \min{ \sqrt{\min}(P), \sqrt{\min}(Q) } )</td>
<td>( \min{ t_g(P), t_g(Q) } )</td>
</tr>
<tr>
<td>( P \triangleright Q )</td>
<td>( \max{ \sqrt{\min}(P), \sqrt{\min}(Q) } )</td>
<td>( \min{ t_g(P), t_g(Q) } )</td>
</tr>
<tr>
<td>( P \triangleright Q )</td>
<td>( \min{ \sqrt{\min}(P), t + \sqrt{\min}(Q) } )</td>
<td>( \min{ t_g(P), t + t_g(Q) } )</td>
</tr>
</tbody>
</table>

Table 1. Auxiliary functions \( \sqrt{\min} \) and \( t_g \).

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

\[
\text{op } E \triangleq E_{\text{op }} \quad \text{for } \text{op} \in \{ a_t, \cdot, \setminus, A, [I] \} \text{ and } \quad E \text{ op } F \triangleq E_{\text{op }} F \quad \text{for } \text{op} \in \{ +, ;, ||, A, \triangleright, \triangleright_t \}
\]

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

\[
\mathcal{E}_1 \triangleq \{ (e), ;, \triangleright, \{ (e, \sqrt{\cdot}) \} \}, \{ (e, [0, \infty)) \} \}
\]

Together with these semantic operators, \( \text{TES}_{\text{fin}}/\simeq_{\alpha} \) constitutes a PA-algebra.

Lemma 18. For \( E, E', F, F' \in \text{TES}_{\text{fin}}/\simeq_{\alpha} \) we have

1. \( d(a_t \cdot E, a_t \cdot E') = 2^{-\inf(I)} \cdot d(E, E') \)
Concluding remarks

Relation with untimed case. Notably exceptions are the works of timed CSP by Reed & Roscoe [17] who use a metric denotational semantics based on timed refusals, and real-time LOTOS by Bryans, Davies & Schneider [8] who use a (non-standard) fixed point semantics based on an advanced form of timed refusals in order to deal with divergence. Both works provide an interleaving semantics. In the non-interleaving setting, related work has been done by

\[ \frac{d(E \circ F, E' \circ F') \leq \max \{ d(E, E'), d(F, F') \}}{\text{for } \circ \in \{ +, \|_A, \| \}} \]

\[ \frac{d(E \circ F, E' \circ F') \leq d(E, E')}{\text{for } \circ \in \{ \setminus A, \setminus \}} \]

\[ \frac{d(E \circ F, E' \circ F') \leq \max \{ d(E, E'), 2^{-\sqrt{\text{min}(E)}}, d(F, F') \}}{\text{for } \circ \in \{ +, \|_A, \| \}} \]

\[ \frac{d(E \triangleright F, E' \triangleright F') \leq \max \{ d(E, E'), 2^{-t}, d(F, F') \}}{\text{for } \triangleright} \]

where \( \sqrt{\text{min}(E)} \) = \( \inf \{ \text{mtime}(e) \mid e \in E, t(e) = \sqrt{\} \}) \).

Lemma 19. For each \( \text{decl} \) and homomorphisms \( \phi_1, \phi_2 : \text{Expr} \to \text{TES} \uparrow_{\sim_{k_0}} \):

\[ \frac{\hat{d}(F_{\text{decl}}(\phi_1), F_{\text{decl}}(\phi_2)) \leq 2^{-(g(\text{decl})) \cdot \hat{d}(\phi_1, \phi_2)}}{} \]

where \( \hat{d} \) is defined in equation (1).

This result says that \( F_{\text{decl}} \) is contracting with contraction coefficient \( 2^{-(g(\text{decl}))} \) provided that \( \text{decl} \) is time-guarded. Thus, for time-guarded declaration \( \text{decl} \), \( F_{\text{decl}} \) has a unique fixed point, say \( \phi_{\text{decl}} \). The metric semantics \( M_{\text{cms}} : \text{TGP} \to \text{TES} \uparrow_{\sim_{k_0}} \) is now defined by \( M_{\text{cms}}((\text{decl}, P)) = \phi_{\text{decl}}(P) \).

6 Concluding remarks

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

Consistency. [10] defines a \( \text{cpo} \)-based denotational semantics \( M_{\text{cpo}} \) and an (event-based) operational semantics for PA such that \( \text{Traces of } M_{\text{cpo}} \) are precisely the time event traces that are generated operationally. The formal relationship between our \( \text{cpo} \) and metric semanitics is as follows. Let \( \text{TES} \uparrow_{\sim_{k_0}} \) be the set of time event structures that are finitely approximatable. For time-guarded \( \langle \text{decl}, P \rangle \) it follows that \( M_{\text{cpo}}((\langle \text{decl}, P \rangle)) \) is finitely approximatable. Function \( f : \text{TES} \uparrow_{\sim_{k_0}} \to \text{TES} \uparrow_{\sim_{k_0}} \) with \( f(E) \equiv E \) is a homomorphism between the PA-algebras \( \text{TES} \uparrow_{\sim_{k_0}} \) and \( \text{TES} \uparrow_{\sim_{k_0}} \). Then, according to the results of [3], we obtain for any time-guarded process \( \langle \text{decl}, P \rangle : f(M_{\text{cpo}}((\langle \text{decl}, P \rangle))) = M_{\text{cms}}((\langle \text{decl}, P \rangle)). \)

This entails that the presented metric semantics is significantly more abstract than the \( \text{cpo} \)-based, and consequently, the operational semantics of TGPA.

Related work. Several real-time extensions of process algebras have been proposed in the literature; for an overview see [14]. Usually, timed process algebras are provided with an operational semantics in the style of Plotkin that is based on some notion of (timed) transition system. Notably exceptions are the works on timed CSP by Reed & Roscoe [17] who use a metric denotational semantics based on timed refusals, and real-time LOTOS by Bryans, Davies & Schneider [8] who use a (non-standard) fixed point semantics based on an advanced form of timed refusals in order to deal with divergence. Both works provide an interleaving semantics. In the non-interleaving setting, related work has been done by
Murphy [13] on interval event structures in which events have a duration. Murphy uses time truncation—in a similar way as we do in the metric semantics—as a basis for obtaining limiting infinite objects using ideal completions.

References


This article was processed using the E\TeX macro package with LLNCS style