Third AMAST Workshop on Real-Time Systems Proceedings

Papers presented at Third AMAST Workshop on Real-Time System Development
March 6–8, 1996, Salt Lake City, Utah

Contents

Chapter 1

A consistent causality-based view on a timed process algebra

Joost-Pieter Katoen[a]
Diego Latella[b]
Rom Langerak[a]
Ed Brinksma[a]
Tommaso Bolognesi[b]

[a]*Faculty of Computing Science, University of Twente*
P.O. Box 217, 7500 AE Enschede, The Netherlands
{ katoen, langerak, brinksma }@cs.utwente.nl

[b]*CNUCE Istituto del CNR, Via Santa Maria 36, 56100 Pisa, Italy*
{ d.latella, t.bolognesi }@cnuce.cnr.it

This paper discusses a timed variant of a process algebra akin to LOTOS, baptized TPA, in a causality-based setting. Two timed features are incorporated—a delay function which constrains the occurrence time of atomic actions and an urgency operator that forces (local or synchronized) actions to happen urgently. Timeouts are typical urgent phenomena. A novel timed extension of event structures is introduced and used as a vehicle to provide a denotational causality-based semantics for TPA. In addition, an operational interleaving semantics is presented based on time- and action-transitions that is shown to be consistent with an 'interleaving view' of the event structure semantics. By adopting this dual approach the well-developed timed interleaving view is extended with a consistent timed partial order view and a comparison is facilitated of the partial order model and the variety of existing (interleaved) timed process algebras.

1.1 Introduction

We study—in a causality-based setting—a timed extension of a basic process algebraic formalism including multi-way synchronization. The formalism, referred to as TPA, is based on a subset of LOTOS [4]. The approach followed in this paper can, however, be adapted to related process algebras like CCS [24], CSP [15], and ACP [2]. Two timed features are incorporated—a delay function which constrains the occurrence time of atomic actions and an urgency operator that forces (local or synchronized) actions to happen urgently. Urgent actions are important to model timeouts that are forced to occur at a certain time—irrespective of the rest of the system—in case some desired action (like receiving an acknowledgement) has not happened yet.

Various timed process algebras have been developed based on the *interleaving* of independent actions [5, 25, 30]. Although each timed formalism has its own characteristics and operators, one may say that the way in which to construct a timed process algebra in an interleaving setting is well-developed, see for instance the recipe in [27]. Due to their observational nature interleaving models are quite appropriate for the description of a system at a high level of abstraction (i.e. considering the system's behaviour as viewed from the outside), and for conformance testing [1]. The incorporation of time in such models is important to obtain an overall view on how the system's behaviour evolves in (linear) time. In the final stages of the design trajectory, however, the global state assumption hampers us to faithfully model the distribution aspects of a system, each part having its own local state. At this design phase the 'local' causal dependencies between actions and their timing constraints are important, while interleavings with actions of other (irrelevant) system parts burden the design. (Timed) partial order models are considered to be much more appropriate here.

This motivates the need for the support of the design process with a coherent set of complementary semantic models. For our timed formalism TPA we therefore take a dual approach—we provide an event-based operational semantics for this timed process algebra which yields an interleaving semantics when omitting the event identifiers, and extend this view with a novel *causality-based* semantics. The resulting operational and denotational semantics are proven to be consistent in the sense that they generate identical sets of timed event traces. The causality-based model is a timed extension of Langerak's *bundle* event structures [20], an adaptation of Winskel's labeled event structures [32] to fit the specific requirements of parallel composition with multi-way synchronization.

The specification of timing aspects is crucial for performing performance analysis. Preliminary studies indicate that the analysis of performance aspects could benefit from a causality-based setting [8, 9, 18] as the parallelism between system components is explicitly retained in the semantic model. In addition, a causality-based model facilitates the possibility to study only that part of a system in which one is interested for the analysis in a relatively easy way (locality) and does not suffer from the *state* *explosion problem*—parallelism leads to the sum of the components states, rather than to their product (as in interleaving).

1.2 Timed event structures

Bundle event structures consist of *events* labeled with actions (an event modeling the occurrence of its action), together with relations of causality and conflict between events. System runs can be modeled as partial orders of events satisfying certain constraints posed by the causality and conflict relations between the events. *Conflict* is a symmetric binary relation between events and the intended meaning is that when two events are in conflict, they can never both happen in a single system run. *Causality* is represented by a relation between a set of events X, that are pairwise in conflict, and an event e. The interpretation is that if e happens in a system run, exactly one event in X has happened before (and caused e). This enables us to uniquely define a causal ordering between the events in a system run. When there is neither a conflict nor a causal relation between events they are independent. Once enabled, independent events can occur in any order or in parallel.

Definition 1. *A* bundle event structure $\mathcal{E}$ *is a quadruple* $(E, \#, \mapsto, l)$ *with* E*, a set of* events*,* $\# \subseteq E \times E$*, the (irreflexive and symmetric)* conflict *relation,* $\mapsto \subseteq 2^E \times E$*, the* causality *relation, and* $l : E \to \mathsf{Act}$*, the* action-labeling *function, where* Act *is a set of action labels, such that* $\mathcal{E}$ *satisfies*

$$\forall X \subseteq E, e \in E : X \mapsto e \;\Rightarrow\; (\forall e_i, e_j \in X : e_i \neq e_j \;\Rightarrow\; e_i \# e_j) \quad .$$

The constraint specifies that for bundle $X \mapsto e$ all events in X are in mutual conflict. Bundle event structures are graphically represented in the following way. Events are denoted as dots; near the dot the action label is given. Conflicts are indicated by dotted lines between representations of events. A bundle (X, e) is indicated by drawing an arrow from each event in X to e and connecting all arrows by small lines. We often denote an event labeled a by e_a.

In the sequel we adopt the following notations. For sequences $\sigma = x_1 \ldots x_n$, let $\overline{\sigma}$ denote the set of elements in σ, that is, $\overline{\sigma} = \{ x_1, \ldots, x_n \}$, and let σ_i denote the prefix of σ up to the $(i-1)$-th element, that is, $\sigma_i = x_1 \ldots x_{i-1}$, for $0 < i \leq n+1$. For σ a sequence of events $e_1 \ldots e_n$ we define $\mathsf{cfl}(\sigma) = \{ e \in E \mid \exists e_i \in \overline{\sigma} : e_i \# e \}$ and $\mathsf{sat}(\sigma) = \{ e \in E \mid \forall X \subseteq E : X \mapsto e \;\Rightarrow\; X \cap \overline{\sigma} \neq \emptyset \}$. $\mathsf{cfl}(\sigma)$ is the set of events that are in conflict with some event in σ. $\mathsf{sat}(\sigma)$ is the set of events that have a causal predecessor in σ for all bundles pointing to them. That is, for events in $\mathsf{sat}(\sigma)$ all bundles are 'satisfied'. Let $\mathsf{en}(\sigma) = \mathsf{sat}(\sigma) \setminus (\mathsf{cfl}(\sigma) \cup \overline{\sigma})$.

The concept of a sequential observation of a system's behaviour is defined as follows. Event traces consist of distinct events (i.e. $e_i \notin \overline{\sigma_i}$, for all i) and are conflict-free ($e_i \notin \mathsf{cfl}(\sigma_i)$), for obvious reasons. In addition, each event in the event trace is

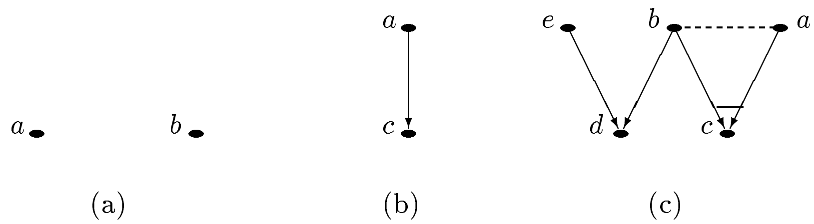

Figure 1.1: Some example bundle event structures.

preceded in the sequence by a causal predecessor for each bundle pointing to it (that is, $e_i \in \mathsf{sat}(\sigma_i)$). That is,

Definition 2. *An* event trace σ *of* $\mathcal{E}$ *is a sequence* $e_1 \ldots e_n$ *with* $\forall\, i : e_i \in \mathsf{en}(\sigma_i)$.

Example 3. Some bundle event structures are depicted in Figure 1.1. Event structure (c) has bundles $\{\, e_a, e_b \,\} \mapsto e_c$, $\{\, e_b \,\} \mapsto e_d$, and $\{\, e_e \,\} \mapsto e_d$, and a conflict between e_a and e_b. Thus, event e_d is enabled once both e_e and e_b have happened, and e_c once either e_a or e_b has occurred before. Example event traces of this structure are $e_a e_e e_c$, $e_b e_c$, and $e_e e_b e_d e_c$.

Time is added to bundle event structures in two ways. To specify the relative delay between causally dependent events time is associated to bundles, and in order to facilitate the specification of timing constraints on events that have no bundle pointing to them (i.e. the initial events), time is also associated to events. Though it seems sufficient to only have time labels for initial events, synchronization of events makes it necessary to allow for equipping all events with time labels, including the non-initial ones. [1]

We assume mappings $\mathcal{T}$ and $\mathcal{D}$ to associate a value of T, the time domain, to bundles and events, respectively. A bundle (X, e) with $\mathcal{T}((X, e)) = t$ is denoted by $X \overset{t}{\mapsto} e$; its interpretation is that if an event in X has happened at a certain time, then e is enabled t time units later. $\mathcal{D}$ associates time to events; the interpretation is that e with $\mathcal{D}(e) = t$ can happen after t time-units from the beginning of the system. Urgency is modeled by a predicate $\mathcal{U}$ on events: $\mathcal{U}(e)$ is true iff e is an urgent event.

Definition 4. *A* timed *event structure is a quadruple* $\langle \mathcal{E}, \mathcal{D}, \mathcal{T}, \mathcal{U} \rangle$ *with* $\mathcal{E}$ *a bundle event structure* $(E, \#, \mapsto, l)$, $\mathcal{T} : \mapsto \to T$, *the* timing *function,* $\mathcal{D} : E \to T$, *the* delay *function, and* $\mathcal{U} : E \to Bool$, *the* urgency *predicate.*

For depicting timed event structures we use the following conventions. The time associated with a bundle and event is a non-negative real and is depicted near to

[1] Alternatively, we could explicitly model the start of the system by some fictitious event, ω say. Then the time associated to event e can be considered as the time associated to the bundle pointing from the fictitious event to e. We do not consider the introduction of such event ω as the definitions become more complex—ω has to be treated differently than 'normal' events—and proof obligations become more severe—e.g. one has to prove that bundles $X \mapsto e$ satisfy $X = \{\, \omega \,\}$, or $\omega \notin X$ and $e \neq \omega$.

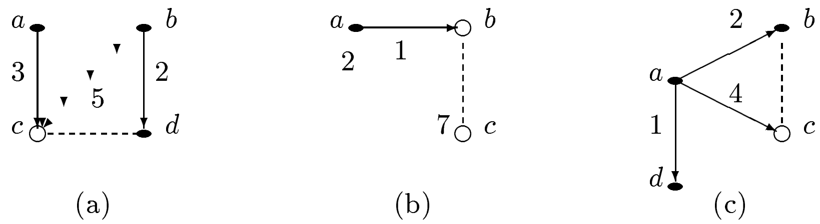

Figure 1.2: Some example timed event structures.

a bundle and event, respectively. For convenience, zero delays are omitted. Urgent events are depicted by open dots, and ordinary events by closed dots.

Example 5. Some example timed event structures are depicted in Figure 1.2. Figure 1.2(a) has bundles $\{ e_a \} \overset{3}{\mapsto} e_c$, $\{ e_b \} \overset{5}{\mapsto} e_c$, $\{ e_b \} \overset{2}{\mapsto} e_d$, and a conflict between urgent event e_c and e_d. For Figure 1.2(b) we have $\mathcal{D}(e_a) = 2$, $\mathcal{D}(e_b) = 0$, $\mathcal{D}(e_c) = 7$ and $\mathcal{T}((\{ e_a \}, e_b)) = 1$.

As a generalization of the notion of event trace we define the notion of timed event trace. A timed event (e, t) denotes that e happened at time t. As a shorthand notation for sequences of timed events $\sigma = (e_1, t_1) \dots (e_n, t_n)$ let $[\sigma]$ denote the sequence of events in σ, i.e. $[\sigma] = e_1 \dots e_n$. Let $\mathsf{time}(\sigma, e)$ denote the moment from which $e \in \mathsf{en}([\sigma])$ could happen, given that each event e_i in timed trace σ occurred at time t_i. e is allowed to occur if at least its delay $\mathcal{D}(e)$ and the time relative to all its immediate causal predecessors is respected. That is,

$$\mathsf{time}(\sigma, e) = max(\mathcal{D}(e), Max\{ t + t_j \mid \exists X \subseteq E : X \overset{t}{\mapsto} e \ \wedge \ X \cap \overline{[\sigma]} = \{ e_j \} \})$$

where Max of the empty set is defined as 0.

Definition 6. *A* timed event trace *of* $\langle \mathcal{E}, \mathcal{D}, \mathcal{T}, \mathcal{U} \rangle$ *is a sequence* σ *of timed events* $(e_1, t_1) \dots (e_n, t_n)$ *with* $e_i \in E$, $t_i \in T$, *satisfying for all* $0 < i \leq n$*:*

1. $e_1 \dots e_n$ *is an event trace of* $\mathcal{E}$
2. $\forall i : (\mathcal{U}(e_i) \ \Rightarrow \ t_i = \mathsf{time}(\sigma_i, e_i)) \ \wedge \ (\neg \mathcal{U}(e_i) \ \Rightarrow \ t_i \geq \mathsf{time}(\sigma_i, e_i))$
3. $\forall i, j : i < j \ \Rightarrow \ t_i \leq t_j$
4. $\forall i, e : e \in \mathsf{en}(\sigma_i) \ \wedge \ \mathcal{U}(e) \ \Rightarrow \ t_i \leq \mathsf{time}(\sigma_i, e)$.

The second constraint requires correct times to be associated to events in σ—ordinary events can happen at any moment from the time they are enabled and urgent events can happen only as soon as they are enabled. This constraint does, however, not take into account the fact that urgent events may prevent other events to occur after a certain time. For instance, according to the first three constraints, the timed event structure depicted in Figure 1.2(a) has timed event trace $(e_a, 0)(e_b, 2)(e_d, 8)$. However, if event e_d has not happened before time $max(0, 0 + 3, 2 + 5) = 7$, then urgent event e_c should have happened at time 7. Thus, $(e_a, 0)(e_b, 2)(e_d, 8)$ should *not* be considered a legal timed event trace. The last constraint takes this matter into account. Some timed event traces of Figure 1.2(a) are (given that $t_a \leq t_b$) $(e_a, t_a)(e_b, t_b)(e_d, t_d)$ with $t_b{+}2 \leq t_d \leq max(t_a{+}3, t_b{+}5)$ and $(e_a, t_a)(e_b, t_b)(e_c, t_c)$ with $t_c = max(t_a{+}3, t_b{+}5)$.

1.3 A temporal process algebra

Let $a \in \mathsf{Act}$, where $\tau \in \mathsf{Act}$ is a special label representing silent actions, $G \subseteq \mathsf{Act}^-$ (G finite and $\mathsf{Act}^- = \mathsf{Act} \setminus \{\tau\}$), $U \subseteq \mathsf{Act}$ and $H : \mathsf{Act} \to \mathsf{Act}$ a relabeling function with $H(\tau) = \tau$ and $H(a) \neq \tau$ for $a \in \mathsf{Act}^-$. We consider the timed process algebra TPA

$$B ::= \mathbf{0} \;\Big|\; (t)\ a\,;\,B \;\Big|\; B + B \;\Big|\; B\,||_G\,B \;\Big|\; B[H] \;\Big|\; B \setminus G \;\Big|\; \mathcal{U}_U(B).$$

We abbreviate $(0)\,a$ by a and denote the time at which action a occurs by t_a. Terminating $\mathbf{0}$s are omitted and $||_\emptyset$ is denoted $|||$. The precedences of the operators are, in decreasing binding order: ;, +, $||_G$, [] and \, $\mathcal{U}_U()$. If G, U are singleton sets, $\{a\}$ say, we simply denote $||_a$, $\backslash a$ and $\mathcal{U}_a()$. Actions are considered to be atomic and to occur instantaneously. $(t)\ a\,;\,B$ denotes a behaviour which may engage in a from t time units on relative to the beginning of the system and after the occurrence of a behaves like B. t specifies the *relative* delay of an action. $\mathcal{U}_U(B)$ behaves like B except that actions in U are *forced* to happen as soon as they are enabled. Notice that U may contain also internal action τ. Actions in U different from τ are visible to the environment but the environment cannot synchronize with them. The other operators have their usual meaning.

Behaviours may synchronize on a common action as soon as all participants are ready to engage in it, i.e. when all individual timing constraints on such action are met. For example, action c is enabled in the composite behaviour $a\,;\,(3)\,c\,||_c\,b\,;\,(7)\,c$ if both a has occurred at least 3 time-units before and b has occurred at least 7 time-units before, that is, $t_c \geq max(t_a{+}3, t_b{+}7)$. In a similar way, in $a\,;\,(t_1)\,b\,||_{\{a,b\}}\,a\,;\,(t_2)\,b$ action b is enabled after $t_a{+}max(t_1, t_2)$.

The notion of urgency here is an extension of the notion of urgency in an earlier paper [9] where urgent actions are assumed to model activities whose occurrence can be controlled completely internally. Here, urgency can involve several participants and is strongly influenced by the notion of urgency in [5, 6] (see also later on). Once made urgent, actions cannot be used for synchronization any further. Without such a restriction, expressions like $B = \mathcal{U}_b((2)\ b)\,||_b\,\mathcal{U}_b((1)\ b)$ would be allowed. Conforming to the principle that an urgent action happens as soon as all participants are ready for it, $(b, 2)$ would be a trace of B. This would cause a delay of action b in the right component, contradicting its local urgency. The fact that we do not allow synchronizations on urgent events is captured by a syntactical constraint on behaviours which can easily be formulated and is omitted here.

Urgent actions are forced to happen as soon as all participants are ready for it. For example, in

$$B = a\,;\,(3)\;c\,||_c\,b\,;\,((2)\;d + (5)\;c)$$

action c can occur at any $t_c \geq max(t_a{+}3, t_b{+}5)$ if d has not yet appeared. If c has not yet occurred, d can occur from $t_b{+}2$ on. In $\mathcal{U}_c(B)$ action c is forced to happen at $t_c = max(t_a{+}3, t_b{+}5)$ in case d has not yet appeared at that time. That is, d is prevented to occur at any time later than t_c, and can only occur in the interval $[t_b{+}2, t_c]$. At time t_c a non-deterministic choice between c and d occurs (so-called weak timeout [27])—urgency does not impose a priority in this case.

The urge operator is inspired by a similar operator, denoted ρ, introduced in [5]. ρ prevents the passage of time as an alternative to the occurrence of an enabled urgent action. [5] allows synchronizations on urged actions. Such synchronizations only succeed if all participants are ready to participate at the same instant of time. In case a synchronization does not succeed, a so-called time deadlock appears, a situation in which passage of time is blocked as a result of which the entire system may halt execution. In our semantic models no notion of time deadlock is possible. [6] generalizes the notion of urgency by introducing the **time** operator. **time** $a(t_1, t_1)$ **in** B denotes B in which a must occur in interval $[t_1, t_2]$ once it is enabled. $\mathcal{U}_a(B)$ is similar to **time** $a(0,0)$ **in** B.

1.4 Causality-based semantics

The model of timed event structures is well-suited to provide a causality-based semantics of TPA in a compositional way. Let $\mathcal{E}[\![\ ]\!]$ associate to each expression B of TPA a timed event structure $\mathcal{E}[\![\,B\,]\!]$. In Appendix A we provide a complete definition of $\mathcal{E}[\![\ ]\!]$, but here we present it just by example. In Figure 1.3 the timed event structures corresponding to the following expressions are depicted:

(a) $((2)\;a\,;\,(3)\;d + (1)\;b\,;\,(2)\;e)\,|||\,(27)\;c$,

(b) $\mathcal{U}_b((2)\;a\,;\,(4)\;b\,||_b\,(7)\;b)$, and

(c) $((2)\;a\,;\,(5)\;c\,||_c\,(7)\;b\,;\,(1)\;c) \setminus b$.

Case (b) illustrates that by parallel composition even events that have a bundle pointing to them can have a non-zero delay $\mathcal{D}$. As a second example the semantics of $B_1\,||_{\{a,b\}}\,B_2$ is given for $B_1 = (1)\;a\,;\,(5)\;b\,||_b\,c\,;\,(3)\;b$ and $B_2 = (4)\;a\,;\,(2)\;b\,||_b\,(b{+}(3)\;d)$. In addition, the corresponding timed bundle event structure of

$$((2)\;a\,;\,(7)\;x + \mathcal{U}_y((4)\;a\,;\,(11)\;y))\,||_a\,((5)\;a\,;\,(2)\;b)$$

is determined. Figures 1.4(a) resp. (b) illustrate the timed event structure semantics of these expressions. In case (a) $\mathcal{D}(e_a) = max(1,4)$ and $\mathcal{T}((\{\,e_a\,\}, e_b)) = max(2,5)$.

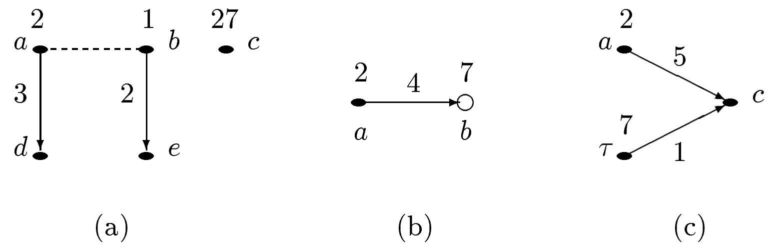

Figure 1.3: Some example timed event structure semantics (I).

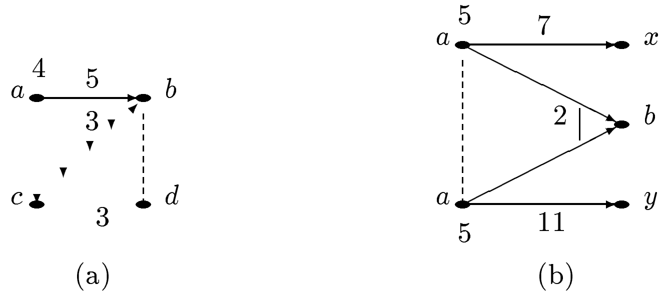

Figure 1.4: Some example timed event structure semantics (II).

A nice result is that $\mathcal{E}[\![\ ]\!]$ is an 'orthogonal' extension of the semantics of LOTOS presented in [21, 20]—that is, removing the parts concerning the timing/urgency of events and timing of bundles in the definition of $\mathcal{E}[\![\ ]\!]$ leads to the causality-based semantics of LOTOS.

1.5 Operational interleaving semantics

Various timed process algebras are known based on an interleaving semantics. In order to compare our non-interleaving approach to these existing approaches and to investigate the 'compatibility' of our proposal with the standard (interleaving) semantics of LOTOS we present an 'interleaving view' on the causality-based semantics. That is, we define an operational (interleaving) semantics for TPA that corresponds to the non-interleaving semantics. The basic idea is to define a transition system (in the sense of [29]) in which we keep track of the occurrence of actions in an expression of TPA rather than the actions themselves. This results in a *timed event transition system*. The approach is adopted from [20] and based on [7].

Each occurrence of an action-prefix is subscripted with an arbitrary but unique event occurrence identifier, denoted by a Greek letter. These occurrence identifiers play the role of event names. For parallel composition new event names can be created. If e is an event name of B and e' an event name in B', then possible new names for events in $B \,||_G\, B'$ are $(e, *)$ and $(*, e')$ for unsynchronized events and (e, e') for synchronized events.

We present two sets of SOS (Structured Operational Semantics [29]) rules that define transition relations $\rightsquigarrow$ and $\rightarrow$, handling the passing of time and the occurrence of events, respectively. These relations transform a pair $\langle B, t\rangle$, where $B \in \mathsf{TPA}$ and $t \in T$. $\langle B, t\rangle$ should be interpreted as behaviour B at time t. Usually one starts with $\langle B, 0\rangle$. $\langle B, t\rangle \rightsquigarrow \langle B', t'\rangle$ means that B evolves into B' as time goes from t to t' $(t' \geq t)$. $\langle B, t\rangle \xrightarrow{(e,a)} \langle B', t\rangle$ means that B at time t performs event e, labeled a, and turns into B' (at t). $\rightsquigarrow$ and $\rightarrow$ are the smallest relations closed under all inference rules defined below.

Inaction: This behaviour cannot perform any internal or communication action, that is, it can perform no $\rightarrow$ transitions. $\mathbf{0}$ permits any amount of time to pass, remaining $\mathbf{0}$.

$$\frac{}{\langle \mathbf{0}, t\rangle \rightsquigarrow \langle \mathbf{0}, t'\rangle} \quad (t' \geq t)$$

Action prefix: The behaviour $(t)\, a_\xi\, ;\, B$ will wait for t time units to become $(0)\, a_\xi\, ;\, B$ after which it either permits any amount of time to pass, remaining the same behaviour, or it may perform event (ξ, a) and behave subsequently like B. Let $x \ominus y$ denote $max(x - y, 0)$ for $x, y \in T$.

$$\frac{}{\langle (t')\, a_\xi\, ;\, B, t\rangle \rightsquigarrow \langle (t' \ominus (t''{-}t))\, a_\xi\, ;\, B, t''\rangle} \quad (t'' \geq t) \qquad \frac{}{\langle (0)\, a_\xi\, ;\, B, t\rangle \xrightarrow{(\xi,a)} \langle B, t\rangle}$$

Choice: If the components B_1 and B_2 permit the passage of time with some amount then so does their choice $B_1 + B_2$. If B_1 (or B_2) performs event (ξ, a) and evolves into B_1' (B_2') then $B_1 + B_2$ can do the same.

$$\frac{\langle B_1, t\rangle \rightsquigarrow \langle B_1', t'\rangle \ \wedge\ \langle B_2, t\rangle \rightsquigarrow \langle B_2', t'\rangle}{\langle B_1 + B_2, t\rangle \rightsquigarrow \langle B_1' + B_2', t'\rangle}$$

$$\frac{\langle B_1, t\rangle \xrightarrow{(\xi,a)} \langle B_1', t\rangle}{\langle B_1 + B_2, t\rangle \xrightarrow{(\xi,a)} \langle B_1', t\rangle} \qquad \frac{\langle B_2, t\rangle \xrightarrow{(\xi,a)} \langle B_2', t\rangle}{\langle B_1 + B_2, t\rangle \xrightarrow{(\xi,a)} \langle B_2', t\rangle}$$

Parallel composition: Like for choice, $B_1 \,||_G\, B_2$ allows the passage of time with some amount if both component behaviours permit this. Components of a parallel composition may perform actions not in the synchronization set G independent of each other, while if both B_1 and B_2 can participate in a synchronization action $a \in G$ then so can their parallel composition.

$$\frac{\langle B_1, t\rangle \rightsquigarrow \langle B_1', t'\rangle \ \wedge\ \langle B_2, t\rangle \rightsquigarrow \langle B_2', t'\rangle}{\langle B_1 \,||_G\, B_2, t\rangle \rightsquigarrow \langle B_1' \,||_G\, B_2', t'\rangle}$$

$$\frac{\langle B_1, t\rangle \xrightarrow{(\xi,a)} \langle B_1', t\rangle}{\langle B_1 \,||_G\, B_2, t\rangle \xrightarrow{((\xi,*),a)} \langle B_1' \,||_G\, B_2, t\rangle} \quad (a \notin G)$$

$$\frac{\langle B_2, t\rangle \xrightarrow{(\xi,a)} \langle B_2', t\rangle}{\langle B_1 \,||_G\, B_2, t\rangle \xrightarrow{((*,\xi),a)} \langle B_1 \,||_G\, B_2', t\rangle} \quad (a \notin G)$$

$$\frac{\langle B_1, t\rangle \xrightarrow{(\xi,a)} \langle B_1', t\rangle \ \wedge\ \langle B_2, t\rangle \xrightarrow{(\psi,a)} \langle B_2', t\rangle}{\langle B_1 \,||_G\, B_2, t\rangle \xrightarrow{((\xi,\psi),a)} \langle B_1' \,||_G\, B_2', t\rangle} \quad (a \in G)$$

Hiding: If B allows the passage of time with a certain amount, then so does $B \setminus G$. Any action that B can perform, can also be performed by $B \setminus G$ whereby actions in set G are turned into silent actions τ.

$$\frac{\langle B, t\rangle \rightsquigarrow \langle B', t'\rangle}{\langle B \setminus G, t\rangle \rightsquigarrow \langle B' \setminus G, t'\rangle}$$

$$\frac{\langle B, t\rangle \xrightarrow{(\xi,a)} \langle B', t\rangle}{\langle B \setminus G, t\rangle \xrightarrow{(\xi,a)} \langle B' \setminus G, t\rangle} \quad (a \notin G) \qquad \frac{\langle B, t\rangle \xrightarrow{(\xi,a)} \langle B', t\rangle}{\langle B \setminus G, t\rangle \xrightarrow{(\xi,\tau)} \langle B' \setminus G, t\rangle} \quad (a \in G)$$

Relabeling: Like for abstraction, if B allows the passage of time with a certain amount, then so does $B[H]$. If B can perform action a and evolve into B', then $B[H]$ can perform $H(a)$ and evolve into $B'[H]$.

$$\frac{\langle B, t\rangle \rightsquigarrow \langle B', t'\rangle}{\langle B[H], t\rangle \rightsquigarrow \langle B'[H], t'\rangle} \qquad \frac{\langle B, t\rangle \xrightarrow{(\xi,a)} \langle B', t\rangle}{\langle B[H], t\rangle \xrightarrow{(\xi,H(a))} \langle B'[H], t\rangle}$$

Urgency: If B permits time to pass with some amount, then $\mathcal{U}_U(B)$ is able to do the same provided that there is no urgent action in U that can be performed by B at any time earlier. Thus, the effect of the urgency operator is to prevent the passage of time as an alternative to the occurrence of an action in the urgent set U. If B can perform (e, a) and evolve into B' then so can $\mathcal{U}_U(B)$, evolving into $\mathcal{U}_U(B')$.

$$\frac{\langle B, t\rangle \rightsquigarrow \langle B', t'\rangle}{\langle \mathcal{U}_U(B), t\rangle \rightsquigarrow \langle \mathcal{U}_U(B'), t'\rangle} \quad (C) \qquad \frac{\langle B, t\rangle \xrightarrow{(\xi,a)} \langle B', t\rangle}{\langle \mathcal{U}_U(B), t\rangle \xrightarrow{(\xi,a)} \langle \mathcal{U}_U(B'), t\rangle}$$

Here, C abbreviates $\forall\, a \in U : t'{-}t \leq d_{min}(a,B)$ where $d_{min}(a,B)$ denotes for initial action a in B the minimal time at which it can appear. The interpretation of $d_{min}(a,B) = \infty$ is that B is not able to perform an a action initially.

Definition 7. For $a \in \mathsf{Act}$ and $B \in \mathsf{TPA}$, function d_{min} is defined as:

$$
\begin{array}{rcl}
d_{min}(a,\mathbf{0}) & = & \infty \\
d_{min}(a,(t)\,b\,;\,B) & = & \left\{ \begin{array}{ll} \infty & \text{if } a \neq b \\ t & \text{if } a = b \end{array} \right. \\
d_{min}(a,B_1+B_2) & = & min(d_{min}(a,B_1), d_{min}(a,B_2)) \\
d_{min}(a,B_1 \,||_G\, B_2) & = & \left\{ \begin{array}{ll} min(d_{min}(a,B_1), d_{min}(a,B_2)) & \text{if } a \notin G \\ max(d_{min}(a,B_1), d_{min}(a,B_2)) & \text{if } a \in G \end{array} \right. \\
d_{min}(a,B \setminus G) & = & \left\{ \begin{array}{ll} Min\{\, d_{min}(b,B) \mid b \in G \cup \{\,\tau\,\}\,\} & \text{if } a = \tau \\ \infty & \text{if } a \in G \\ d_{min}(a,B) & \text{if } a \notin G \cup \{\,\tau\,\} \end{array} \right. \\
d_{min}(a,B[H]) & = & Min\{\, d_{min}(b,B) \mid a = H(b)\,\} \\
d_{min}(a,\mathcal{U}_U(B)) & = & d_{min}(a,B).
\end{array}
$$

where Min of the empty set equals ∞. Here it is assumed that min, max and their generalizations on sets of events are defined on $T \cup \{\,\infty\,\}$ in the obvious way. For instance, $min(t,\infty) = t$ and $max(t,\infty) = \infty$.

From the event transition system defined by $\rightarrow$ we can easily obtain the standard interleaving semantics for LOTOS by omitting time components from tuples $\langle \ldots \rangle$ and the event identifiers from transitions and expressions. When retaining the event identifiers and only omitting the time components we obtain the event transition system obtained in [20]. In this sense the presented transition system can be considered to be an *orthogonal* extension of the untimed one.

1.6 Consistency between causality-based and operational semantics

In this section we investigate the relationship between the causality-based and operational semantics of TPA. For convenience we first introduce a new transition relation $\longrightarrow\!\!\!\!\rightarrow$.

Definition 8. *Let* $\langle B,t\rangle \xrightarrow{(e,a,t')}\!\!\!\!\rightarrow \langle B',t'\rangle$ *iff* $\exists\, B'' : \langle B,t\rangle \rightsquigarrow \langle B'',t'\rangle \;\wedge$ $\langle B'',t'\rangle \xrightarrow{(e,a)} \langle B',t'\rangle$.

Using the relation $\longrightarrow\!\!\!\!\rightarrow$ the notion of timed event trace and a trace derivation relation $\xrightarrow{\sigma}\!\!\!\!\rightarrow$ can be defined in the usual way. We summarize the following results, where for $B \in \mathsf{TPA}$ the set $U(B)$ denotes the set of urgent actions in B. $U(B)$ can easily be defined by induction on B. The proofs are omitted and can be found in [19].

Theorem 9. For all $B, B' \in \mathsf{TPA}$, $t, t' \in T$ and $a \in \mathsf{Act} \cup \{\tau\}$ we have

1. $\langle B, t' \rangle \xrightarrow{(e,a,t)} \;\Rightarrow\; t \geq t' + d_{min}(a, B)$.
2. $\forall\, b \in U(B) : \langle B, t' \rangle \xrightarrow{(e,b,t)} \;\Rightarrow\; t = t' + d_{min}(b, B)$.
3. $d_{min}(a, B) = \infty \;\Rightarrow\; \neg\,(\exists\, B' : \langle B, t \rangle \xrightarrow{(e,a,t')} B')$.
4. $\langle B, t \rangle \xrightarrow{(e,a,t')} \;\Rightarrow\; t' \leq t + Min\{\, d_{min}(b, B) \mid b \in U(B)\,\}$.
5. $\langle B, t \rangle \xrightarrow{(e,a,t_a)} \;\wedge\; \langle B, t \rangle \rightsquigarrow \langle B', t' \rangle \;\Rightarrow\; d_{min}(a, B') = d_{min}(a, B) \ominus (t' - t)$.

1. expresses that the time determined by $d_{min}(a, B)$ corresponds to the earliest moment at which initial action a can be performed by B. 2. says that urgent actions in B can only happen as soon as they are enabled. If $d_{min}(a, B) = \infty$ then B is not able to perform a initially. This is stated in 3. 4. states that actions can only be performed by B provided there is no urgent action in B that could occur earlier. Finally, 5. shows the relation between d_{min} and $\rightsquigarrow$.

We now consider the well-known properties time determinism, action persistency, and time additivity [27] for TPA.

Theorem 10. For all $B, B', B'' \in \mathsf{TPA}$, $t, t', t'' \in T$ we have

1. Time determinism: $(\langle B, t \rangle \rightsquigarrow \langle B', t' \rangle \;\wedge\; \langle B, t \rangle \rightsquigarrow \langle B'', t' \rangle) \;\Rightarrow\; B' = B''$.
2. Action persistency: $(\langle B, t \rangle \xrightarrow{(e,a)} \;\wedge\; \langle B, t \rangle \rightsquigarrow \langle B', t' \rangle) \;\Rightarrow\; \langle B', t' \rangle \xrightarrow{(e,a)}$.
3. Time additivity[2]: $\langle B, t \rangle \rightsquigarrow \langle B', t{+}(t'{+}t'') \rangle$ iff
   $(\exists\, B'' : \langle B, t \rangle \rightsquigarrow \langle B'', t{+}t' \rangle \rightsquigarrow \langle B', t{+}(t'{+}t'') \rangle$.

The proofs of all theorems are by induction on the structure of B. These proofs are rather straightforward and omitted here; see [19].

Since the transition system under $\longrightarrow\!\!\!\!\rightarrow$ is deterministic, this transition system can be represented by its set of timed event traces $\mathcal{T}[\![\, B \,]\!]$. This set can be characterized in a denotational way, and subsequently proven to coincide with the set of timed event traces of the corresponding timed event structure $\mathcal{E}[\![\, B \,]\!]$. We thus have the following

[2]For timed I/O-automata [31] a stronger notion is adopted that says that there must be a *trajectory* of consistent states through the interval $[t, t']$. Since our timed transition system satisfies the image-finiteness condition (that is, for any B and t' there are at most finitely many B' such that $\langle B, t \rangle \rightsquigarrow \langle B', t' \rangle$) it follows from [17] that our model also satisfies this stronger trajectory condition.

consistency result, where $TT(\mathcal{E}[\![B]\!])$ denotes the set of timed event traces of $\mathcal{E}[\![B]\!]$.

Theorem 11. For all $B \in \mathsf{TPA} : TT(\mathcal{E}[\![B]\!]) = \mathcal{T}[\![B]\!]$.

The proof of this theorem is quite involved and omitted here for space reasons; see [19]. The main issue is to characterize correctly the timed traces of + and $\mathcal{U}_U()$ in a denotational way and to prove that this characterization coincides with the timed event traces of the corresponding timed event structure.

1.7 Related Work

To the best of our knowledge this constitutes the first timed causality-based model incorporating urgent and non-urgent actions. A few timed extensions of causality-based models do exist. For the sake of brevity we just briefly mention them here. [13] describes an interesting real-time extension of CCS based on causal trees. [23] considers a theoretical model, called timed configurations, where all events are treated to be urgent. [26] treats a timed extension of event structures in which events have a duration and all are urgent. [16] introduce a real-time process language consisting of simple sequential processes that are composed by means of 'layering' and independent parallelism. An extension of Pratt's pomset model with delays is studied in [10]. The behaviour of timed systems with both disjunctive and conjunctive causality is analyzed in [14].

For the untimed case several approaches exist that relate a causality-based semantics to an interleaving one [3, 7, 22]. These investigations differ from our work in particular in the causality-based model, the language at hand, and the type of consistency relation between the two types of semantics. [3, 22] prove the consistency between an operational semantics for Theoretical CSP (TCSP) and a compositional true concurrency semantics based on labeled prime event structures. They show that the 'interleaved view' of the event structure semantics—obtained by considering remainders of event structures after the execution of a single event—is (weak) bisimilar to the operational semantics of TCSP. [12] proposed an approach to prove the consistency of an operational non-interleaving semantics of CCS (with guarded recursion) and a denotational one based on labeled prime event structures. From the operational semantics an occurrence net is derived which is shown—using the well-known connection between this class of nets and event structures [28]—to be equal to the event structure obtained in the denotational way.

1.8 Conclusions

This paper introduced a temporal process algebra TPA with just two timed features—a simple delay function and an urgency operator. A novel timed enhancement of event structures is used to provide a causality-based semantics in a compositional way. In addition an operational semantics is given inspired by the separation of the passage of time (relation $\rightsquigarrow$) and the occurrence of actions (relation $\longrightarrow$) as introduced by [25] and adopted by several others [5, 30]. It turns out that the transition system for $\longrightarrow$ is identical to Langerak's untimed transition model [20]. Thus, time is added in a completely orthogonal way. The operational semantics of TPA turns out to be very close to the proposal(s) of Bolognesi & Lucidi [5, 6]. The main difference with these proposals is the treatment of synchronization on urgent actions—they allow them at the prize of introducing time deadlocks, whereas our proposal avoids them. Since the operational and denotational semantics of TPA are consistent in the sense that identical sets of timed event traces are generated we consider the aforementioned characteristics to provide evidence for the adequacy of our timed causality-based model.

A problem in defining an operational semantics is that there seems to be no consensus on how to include time into transition systems—besides models that explicitly distinguish between time- and action-transitions, another school advocates timed action transitions. It can be shown that for TPA without urgency an elegant transition model based on timed actions can be provided which is strong bisimulation equivalent to the 'interleaving view' of the causality-based semantics and which allows 'ill-timed' traces to occur. For space reasons this alternative approach is not presented in this extended abstract.

Though TPA does not include recursion, there are no serious problems in incorporating recursive behaviours. For technical reasons we only need to require *guarded* instantiation, which—from a user's perspective—is not a severe restriction. In the denotational model the semantics of a recursive behaviour is defined using standard fixed point theory. The approach in [8] for the stochastic timed case can be carried over to the timed setting of this paper in a straightforward manner. The full details can be found in [19].

Appendix A: Denotational semantics of TPA

In this appendix we provide the full definition of the causality-based semantics of TPA. Let $\mathcal{E}[\![B_i]\!] = \Gamma_i = \langle \mathcal{E}_i, \mathcal{D}_i, \mathcal{T}_i, \mathcal{U}_i \rangle$, for $i = 1, 2$, with $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ and $E_1 \cap E_2 = \emptyset$.

For action-prefix $(t)\ a\,;\ B_1$, a bundle is introduced from a new non-urgent event e_a (labeled a) to all initial events of Γ_1 (as e_a causally precedes them) and all events in Γ_1 that have a non-zero delay. For all these initial and non-zero delay events e the delay is now relative to e_a, so each bundle $\{ e_a \} \mapsto e$ is associated with a time delay $\mathcal{D}_1(e)$, and $\mathcal{D}(e)$ is made zero. $\mathcal{D}(e_a)$ becomes t.

$\mathcal{E}[\![B_1 + B_2]\!]$ is equal to the union of Γ_1 and Γ_2 extended with mutual conflicts between all initial events of Γ_1 and Γ_2 such that either B_1 or B_2 can happen. $\mathcal{E}[\![B_1 \backslash G]\!]$

is identical to Γ_1 except that events labeled with a label in G are now labeled with τ. $\mathcal{E}[\![B_1[H]]\!]$ and $\mathcal{E}[\![\mathcal{U}_U(B_1)]\!]$ are defined similarly where events are relabeled according to H, respectively become urgent when $l_1(e) \in U$.

For parallel composition the events of $\mathcal{E}[\![B_1 \,||_G\, B_2]\!]$ are constructed as follows: an event e of Γ_1 or Γ_2 that does not need to synchronize is paired with the auxiliary symbol $*$, and an event which is labeled with an action in G is paired with all events (if any) in the other process that are equally labeled. Thus events are *pairs* of events of Γ_1 and Γ_2, or with one component equal to $*$. Two events are now put in conflict if any of their corresponding components are in conflict, or if different events have a common component different from $*$ (such events appear if two or more events in one process synchronize with the same event in the other process). A bundle is introduced such that if we take the projection on the i-th component (i=1, 2) of all events in the bundle we obtain a bundle in Γ_i. The bundle delay is equal to the maximum of the delays of the bundles we get by projecting on the i-th components (i=1, 2),if this projection yields a bundle in Γ_i. The event delay is the maximum of the delays of its components that are not equal to $*$. Finally, an event is urgent when one of its components is urgent.

Definition 12. For $\Gamma = \langle \mathcal{E}, \mathcal{D}, \mathcal{T}, \mathcal{U} \rangle$ let $pos(\Gamma)$ denote the set of non-zero delay events in Γ $init(\Gamma)$ the set of initial events in Γ. That is, $pos(\Gamma) = \{ e \in E \mid \mathcal{D}(e) \neq 0 \}$ and $init(\Gamma) = \{ e \in E \mid \neg (\exists X \subseteq E : X \mapsto e) \}$.

As a shorthand notation we use $pin(\Gamma) = pos(\Gamma) \cup init(\Gamma)$. We suppose there is an infinite universe of events E_U. For $G \subseteq \mathsf{Act}^-$ let $E_i^s = \{ e \in E_i \mid l_i(e) \in G \}$ the set of synchronization events and $E_i^f = E_i \setminus E_i^s$ the set of non-synchronizing events (i=1, 2).

Definition 13. $\mathcal{E}[\![\ ]\!]$ is defined as follows:

$$
\begin{array}{rcl}
\mathcal{E}[\![\mathbf{0}]\!] & = & \langle (\emptyset, \emptyset, \emptyset, \emptyset), \emptyset, \emptyset, \emptyset \rangle \\
\mathcal{E}[\![(t)\ a\,;\, B_1]\!] & = & \langle (E, \#_1, \mapsto, l_1 \cup \{ (e_a, a) \}), \mathcal{D}, \mathcal{T}, \mathcal{U}_1 \cup \{ (e_a, \text{false}) \} \rangle \text{ where} \\
E & = & E_1 \cup \{ e_a \} \text{ for } e_a \in E_U \setminus E_1 \\
\mapsto & = & \mapsto_1 \cup (\{ \{ e_a \} \} \times pin(\Gamma_1)) \\
\mathcal{D} & = & \{ (e_a, t) \} \cup (E_1 \times \{ 0 \}) \\
\mathcal{T} & = & \mathcal{T}_1 \cup \{ ((\{ e_a \}, e), \mathcal{D}_1(e)) \mid e \in pin(\Gamma_1) \} \\
\mathcal{E}[\![B_1 + B_2]\!] & = & \langle (E_1 \cup E_2, \#, \mapsto_1 \cup \mapsto_2, l_1 \cup l_2), \mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{T}_1 \cup \mathcal{T}_2, \mathcal{U}_1 \cup \mathcal{U}_2 \rangle \\
\# & = & \#_1 \cup \#_2 \cup (init(\Gamma_1) \times init(\Gamma_2)) \\
\mathcal{E}[\![B_1 \setminus G]\!] & = & \langle (E_1, \#_1, \mapsto_1, l), \mathcal{D}_1, \mathcal{T}_1, \mathcal{U}_1 \rangle \text{ where} \\
 & & (l_1(e) \in G \Rightarrow l(e) = \tau) \wedge (l_1(e) \notin G \Rightarrow l(e) = l_1(e)) \\
\mathcal{E}[\![B_1[H]]\!] & = & \langle (E_1, \#_1, \mapsto_1, H \circ l_1), \mathcal{D}_1, \mathcal{T}_1, \mathcal{U}_1 \rangle \\
\mathcal{E}[\![\mathcal{U}_U(B_1)]\!] & = & \langle (E_1, \#_1, \mapsto_1, l_1), \mathcal{D}_1, \mathcal{T}_1, \mathcal{U} \rangle \text{ where} \\
\mathcal{U}(e) & = & \mathcal{U}_1(e) \vee (l_1(e) \in U)
\end{array}
$$

$$
\begin{array}{rcl}
\mathcal{E}[\![\, B_1 \,||_G\, B_2 \,]\!] & = & \langle (E, \#, \mapsto, l), \mathcal{D}, \mathcal{T}, \mathcal{U} \rangle \text{ where} \\
E & = & (E_1^f \times \{\, * \,\}) \cup (\{\, * \,\} \times E_2^f) \cup \\
 & & \{\, (e_1, e_2) \in E_1^s \times E_2^s \mid l_1(e_1) = l_2(e_2) \,\} \\
(e_1, e_2) \,\#\, (e_1', e_2') & \Leftrightarrow & (e_1 \,\#_1\, e_1') \;\vee\; (e_2 \,\#_2\, e_2') \;\vee\; (e_1 = e_1' \neq * \;\wedge\; e_2 \neq e_2') \;\vee \\
 & & (e_2 = e_2' \neq * \;\wedge\; e_1 \neq e_1') \\
X \mapsto (e_1, e_2) & \Leftrightarrow & \exists\, X_1 : (X_1 \mapsto_1 e_1 \;\wedge\; X = \{\, (e_i, e_j) \in E \mid e_i \in X_1 \,\}) \;\vee \\
 & & \exists\, X_2 : (X_2 \mapsto_2 e_2 \;\wedge\; X = \{\, (e_i, e_j) \in E \mid e_j \in X_2 \,\}) \\
l((e_1, e_2)) & = & \mathsf{if}\ e_1 = * \ \mathsf{then}\ l_2(e_2)\ \mathsf{else}\ l_1(e_1) \\
\mathcal{D}((e_1, e_2)) & = & max(\mathcal{D}_1(e_1), \mathcal{D}_2(e_2)) \text{ with } \mathcal{D}_i(*) = 0. \\
\mathcal{T}((X, (e_1, e_2))) & = & max(h_1, h_2) \text{ with} \\
h_1 & = & \mathsf{if}\ (\exists\, X_1 \subseteq E_1 : X_1 \overset{t_1}{\mapsto}_1 e_1 \;\wedge\; X = \{\, (e_i, e_j) \in E \mid e_i \in X_1 \,\}) \\
 & & \mathsf{then}\ t_1\ \mathsf{else}\ 0 \\
h_2 & = & \mathsf{if}\ (\exists\, X_2 \subseteq E_2 : X_2 \overset{t_2}{\mapsto}_2 e_2 \;\wedge\; X = \{\, (e_i, e_j) \in E \mid e_j \in X_2 \,\}) \\
 & & \mathsf{then}\ t_2\ \mathsf{else}\ 0 \\
\mathcal{U}((e_1, e_2)) & = & \mathcal{U}_1(e_1) \;\vee\; \mathcal{U}_2(e_2) \text{ with } \mathcal{U}_i(*) = \text{false.}
\end{array}
$$

Bibliography

[1] S. Abramsky. Observational equivalence as a testing equivalence. *Theor. Comp. Sci.* **53**:225–241 (1987)

[2] J.C.M. Baeten and W.P. Weijland. *Process Algebra, Cambridge Tracts in Theoret. Comp. Sci.* vol. **18**. Cambridge Univ. Press (1990)

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

[4] T. Bolognesi and E. Brinksma. Introduction to the ISO specification language LOTOS. *Comp. Netw. and ISDN Sys.* **14**:25–59 (1987)

[5] T. Bolognesi and F. Lucidi. Timed process algebras with urgent interactions and a unique powerful binary operator. In de Bakker et al.[11], pages 124–148.

[6] T. Bolognesi, F. Lucidi, and S. Trigila. Converging towards a timed LOTOS standard. *Comp. Standards & Interfaces* **16**:87–118 (1994)

[7] G. Boudol and I. Castellani. Three equivalent semantics for CCS. In I. Guessarian (ed), *Semantics of Systems of Concurrent Processes*, LNCS **469**:96–141. Springer-Verlag (1990)

[8] E. Brinksma, J.-P. Katoen, R. Langerak, and D. Latella. A stochastic causality-based process algebra. *The Computer Journal* **38**(6):552–565 (1995)

[9] E. Brinksma, J.-P. Katoen, R. Langerak, and D. Latella. Performance analysis and true concurrency semantics. In T. Rus and C. Rattray (eds), *Theories and Experiences for Real-Time System Development, AMAST Series in Computing* vol. **2**:309–337. World Scientific (1994)

[10] R. Casley, R.F. Crew, J. Meseguer, and V. Pratt. Temporal structures. *Mathematical Structures in Comp. Sci.* **1**:179–213 (1991)

[11] J.W. de Bakker et al. (eds). *Real-time: Theory in Practice*, LNCS **600**. Springer-Verlag (1992)

[12] P. Degano, R. De Nicola, and U. Montanari. On the consistency of 'truly concurrent' operational and denotational semantics (extended abstract). In *Third Ann. Symp. on Logic in Comp. Sci.*, 133–141. IEEE Computer Society Press (1988)

[13] C. Fidge. A constraint-oriented real-time process calculus. In M. Diaz and R. Groz (eds), *Formal Description Techniques V*, *IFIP Transactions* **C–10**:363–378. North-Holland (1993)

[14] J. Gunawardena. A dynamic approach to timed behaviour. In B. Jonsson and J. Parrow (eds), *Concur' 94*, LNCS **836**:178–193. Springer-Verlag (1994)

[15] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall (1985)

[16] W. Janssen, M. Poel, Q. Wu, and J. Zwiers. Layering of real-time distributed processes. In H. Langmaack et al. (eds), *Formal Techniques in Real-Time and Fault-Tolerant Systems*, LNCS **863**:393–417 (1994)

[17] A. Jeffrey, S. Schneider, and F. Vaandrager. A comparison of additivity axioms in timed transition systems. Technical Report CS-R9366, Centre for Mathematics and Computer Science (1993)

[18] J.-P. Katoen, R. Langerak, and D. Latella. Modelling systems by probabilistic process algebra: An event structures approach. In R.L. Tenney et al. (eds), *Formal Description Techniques VI*, *IFIP Transactions* **C–22**:253–268. North-Holland (1994)

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

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

[21] R. Langerak. Bundle event structures: a non-interleaving semantics for LOTOS. In M. Diaz and R. Groz (eds), *Formal Description Techniques V*, *IFIP Transactions* **C–10**:331–346. North-Holland (1993)

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

[23] A. Maggiolo-Schettini and J. Winkowski. Towards an algebra for timed behaviours. *Theor. Comp. Sci.* **103**:335–363 (1992)

[24] R. Milner. *Communication and Concurrency*. Prentice-Hall (1989)

[25] F. Moller and C. Tofts. A temporal calculus of communicating systems. In J.C.M. Baeten and J.W. Klop (eds), *Concur '90*, LNCS **458**:401–415. Springer-Verlag (1990)

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

[27] X. Nicollin and J. Sifakis. An overview and synthesis on timed process algebras. In de Bakker et al.[11], pages 526–548.

[28] M. Nielsen, G. D. Plotkin, and G. Winskel. Petri nets, event structures and domains I. *Theor. Comp. Sci.* **13**:85–108 (1981)

[29] G.D. Plotkin. A structural approach to operational semantics. Tech. Rep. DAIMI FN-19, Comp. Sci. Dept., Aarhus University (1981)

[30] S. Schneider. An operational semantics for timed CSP. *Inf. and Comp.* **116**:193–213 (1995)

[31] F. Vaandrager and N. Lynch. Action transducers and timed automata. In W.R. Cleaveland (ed), *Concur '92*, LNCS **630**:436–455. Springer-Verlag (1992)

[32] G. Winskel. An introduction to event structures. In J.W. de Bakker et al. (eds), *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, LNCS **354**:364–397. Springer-Verlag (1989)