Joint NCC & IIS Bull., Comp. Science, 18 (2002), 1–14

Concurrent testing for timed event structures*

M.V. Andreeva, I.B. Virbitskaite

The intention of the paper is to extend the testing methodology to true concurrent models with a dense time domain. In particular, we develop three different semantics, based on interleaving, steps, and partial orders of actions, for testing equivalence in the setting of timed event structures. We study the relationship between these three approaches and show their discriminating power. Furthermore, when dealing with particular subclasses of the model under consideration, such as timed sequential and timed deterministic event structures, there is no difference between a more concrete and a more abstract approach.

1. Introduction

For the purpose of correctness analysis of systems, it is necessary to provide a number of equivalence notions in order to be able to choose the most suitable view of system behaviours. In concurrency theory, a variety of equivalences have been promoted, and the relationship between them has been quite well-understood (see, for example, [9, 10]).

Testing [8] is one of the major equivalences of concurrency theory. Testing equivalences and preorders are defined in terms of tests that processes may and must satisfy. Two processes are considered as testing equivalent, if there is no test that can distinguish them. A test is usually itself a process applied to a process by computing both together in parallel. A particular computation is considered to be successful if the test reaches a designated successful state, and the process guarantees the test if every computation is successful.

Recently, the demand for correctness analysis of real time systems, i.e. systems whose descriptions involve a quantitative notion of time, increases rapidly. Timed extensions of interleaving models have been investigated thoroughly in the last ten years. Various recipes on how to incorporate time in transition systems — the most prominent interleaving model — are, for example, described in [2, 14], whereas, the incorporation of real time into equivalence notions is less advanced. There are a few papers (see, for example, [5, 15, 17]), where decidability questions of time-sensitive equivalences are investigated in the setting of timed interleaving models.

*Partially supported by the Russian Foundation for Basic Research under (Grant N 03-01-00648).

In this paper, we seek to develop a framework for testing equivalences in the setting of a timed true concurrent model, to take into account the processes' timing behaviour in addition to their degrees of relative concurrency and nondeterminism. In particular, we develop three different semantics, based on interleaving, steps, and partial orders of actions, for testing equivalence in the setting of event structures with the dense time domain. We also study the relationship between these three approaches and show their discriminating power. Furthermore, when dealing with particular subclasses of the model, such as timed sequential and timed nondeterministic processes, there is no difference between a more concrete and a more abstract approach. This line of research is sometimes referred to as comparative concurrency semantics.

There have been several motivations for this work. One has been the papers [1, 11] which have developed concurrent variants of testing in the setting of event structures. Another origin of this study has been the papers [7] and [15], which have treated timed interleaving testing for discrete time and dense time transition models, respectively. A next origin of this study has been given by the papers (see [5, 15, 17] among others), which have extensively studied time-sensitive equivalence notions for interleaving models. However, to our best knowledge, the literature on timed true concurrent models has hitherto lacked for such equivalences. In this regard, the papers [3, 13] is a welcome exception, where the decidability question of timed interleaving testing has been treated in the framework of timed event structures. Finally, another origin has been the papers where step based equivalences have been investigated in the framework of stochastic Petri nets with discrete time.

The rest of the paper is organized as follows. The basic notions concerning timed event structures are introduced in the next section. The definitions of three different semantics (sequences of actions, sequences of multisets, partial ordering of actions) of timed testing are given in Sections 3, 4, and 5, respectively. In the following section, we establish the interrelations between the equivalence notions in the setting of the model under consideration and some its subclasses. The conclusion can be found in Section 7.

2. Timed event structures

In this section, we introduce some basic notions and notations concerning timed event structures.

We first recall a notion of an event structure [18]. The main idea behind event structures is to view distributed computations as action occurrences, called events, together with a notion of causality dependency between events (which is reasonably characterized via a partial order). Moreover, in order to model nondeterminism, there is a notion of conflicting (mutually incompatible) events. A labelling function records which action an event corresponds to. Let *Act* be a finite set of actions.

Definition 1. A (*labelled*) *event structure* over *Act* is a 4-tuple $S = (E, \leq, \#, l)$, where

- E is a countable set of events;
- $\leq \subseteq E \times E$ is a partial order (the *causality relation*), satisfying *the principle of finite causes*: $\forall e \in E \diamond \{e' \in E \mid e' \leq e\}$ is finite;
- $\# \subseteq E \times E$ is a symmetric and irreflexive relation (the *conflict relation*), satisfying *the principle of conflict heredity*: $\forall e, e', e'' \in E \diamond e \# e' \leq e'' \Rightarrow e \# e''$;
- $l : E \longrightarrow Act$ is a labelling function.

For an event structure $S = (E, \leq, \#, l)$, we define $\smile = (E \times E) \setminus (\leq \cup \leq^{-1} \cup \#)$ (the *concurrency relation*); for $e, f \in E$, we let $e \#^1 f \Leftrightarrow e \# f \wedge (\forall e', f' \in E \diamond e' \leq e \wedge f' \leq f \wedge e' \# f' \Rightarrow e' = e \wedge f' = f)$ (the *immediate conflict*). For $C \subseteq E$, the *restriction* of S to C is defined as $S\lceil C = (C, \leq \cap (C \times C), \# \cap (C \times C), l \mid_C)$. We will use $\mathcal{O}$ to denote the empty event structure $(\emptyset, \emptyset, \emptyset, \emptyset)$.

Let $C \subseteq E$. Then C is *left-closed* iff $\forall e, e' \in E \diamond e \in C \wedge e' \leq e \Rightarrow e' \in C$; C is *conflict-free* iff $\forall e, e' \in C \diamond \neg(e \# e')$; C is a *configuration of* S iff C is left-closed and conflict-free. Let $\mathcal{C}(S)$ denote the set of all finite configurations of S.

Next we present a model of timed event structures which are a timed extension of event structures by associating their events with timing constraints that indicate times of event occurrences with regard to a global clock. An execution of a timed event structure is a *timed configuration* that consists of the configuration and the timing function recording global time moments at which events occur and satisfies some additional requirements.

Before introducing the concept of a timed event structure, we need to define some auxiliary notations. Let $\mathbf{N}$ be the set of natural numbers, and $\mathbf{R}_0^+$ the set of nonnegative real numbers. Define the set of intervals: $Interv = \{[d_1, d_2] \mid d_1, d_1 \in \mathbf{R}_0^+,\ d_1 \leq d_2\}$.

We are now ready to introduce the concept of timed event structures.

Definition 2. A (*labelled*) *timed event structure* over *Act* is a pair $TS = (S, D)$, where

- $S = (E, \leq, \#, l)$ is a (labelled) event structure over *Act*;

- $D : E \longrightarrow Interv$ is a timing function such that $e' \leq_S e \Rightarrow \min D(e') \leq \min D(e)$ and $\max D(e') \leq \max D(e)$.

In a graphic representation of a timed event structure, the corresponding action labels and time intervals are drawn near to events. If no confusion arises, we will often use action labels rather than event identities to denote events. The $<$-relation is depicted by arcs (omitting those derivable by transitivity), and conflicts are also drawn (omitting those derivable by conflict heredity). Following these conventions, a trivial example of a labelled timed event structure is shown in Figure 1.

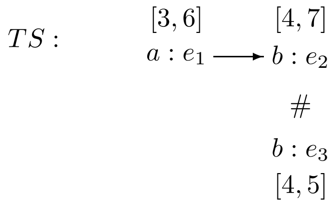

Figure 1

Timed event structures TS and TS' are *isomorphic* (denoted $TS \simeq TS'$), if there exists a bijection $\varphi : E_{TS} \longrightarrow E_{TS'}$ such that $e \leq_{TS} e' \iff \varphi(e) \leq_{TS'} \varphi(e')$, $e \#_{TS} e' \iff \varphi(e) \#_{TS'} \varphi(e')$, $l_{TS}(e) = l_{TS'}(\varphi(e))$, and $D_{TS}(e) = D_{TS'}(\varphi(e))$, for all $e, e' \in E_{TS}$.

Definition 3. Let $TS = (S, D)$ be a timed event structure, $C \in \mathcal{C}(S)$, and $T : C \longrightarrow \mathbf{R}_0^+$. Then $TC = (C, T)$ is a *timed configuration* of TS iff the following conditions hold:

(i) $\forall\, e \in C \;\diamond\; T(e) \in D(e)$;

(ii) $\forall\, e, e' \in C \;\diamond\; e \leq_{TS} e' \;\Rightarrow\; T(e) \leq T(e')$;

(iii) $\forall\, e \in (E \setminus C) \;\diamond\; \max\; D(e) \geq T(e')$ for all $e' \in C$ or
for some $e' \in C$ s.t. $e' \# e$.

Informally speaking, a timed configuration consisting of the configuration and the timing function recording global time moments at which events occur satisfies the following requirements:

(i) an event can occur at a time when its timing constraints are met;

(ii) for all events e and e' occurred if e causally precedes e' then e should temporally precede e';

(iii) occurrences of events should not temporally prevent other events to occur except the events whose conflicting events have occurred before the events had time to occur.

Note, the above definition of a timed configuration ensures that events once ready — i.e., all their causal predecessors have occurred and their timing constraints are satisfied — are forced to occur, provided they are not disabled by others events. Typically such events are timeout mechanisms that guard the occurrence time of other events in the sense that the events are prevented from happening after a particular time instant. The approach looks more suitable to model realistic systems (see [12] for more explanation).

The *initial timed configuration* of TS is $TC_{TS} = (\emptyset, 0)$. We use $\mathcal{TC}(TS)$ to denote the set of finite timed configurations of TS.

To illustrate the concept, consider the set of possible timed configurations of the timed event structure TS shown in Figure 1: $\{(\emptyset,0)$, $(\{e_1\},T_1)$, $(\{e_3\},T_2)$, $(\{e_1,e_3\},T_3)$, $(\{e_1,e_2\},T_4) \mid T_1(e_1) \in [3,5];\ T_2(e_3) \in [4,5];\ T_3(e_1) \in [3,6],\ T_3(e_3) \in [4,5];\ T_4(e_1) \in [3,5],\ T_4(e_2) \in [4,5],\ T_4(e_1) \leq T_4(e_2)\}$.

From now on, for $TC_1 = (C_1,T_1), TC_2 = (C_2,T_2) \in \mathcal{TC}(TS)$ we will write $TC_1 \longrightarrow TC_2$ iff $C_1 \subseteq C_2$, $T_2|_{C_1} = T_1$, and $\forall e \in C_1\ \forall e' \in (C_2 \setminus C_1) \diamond T_1(e) \leq T_2(e')$.

3. Interleaving semantics

In this section, we define timed testing equivalences based on an interleaving observation on timed event structures.

For this purpose we need the following notation. Let $(Act, \mathbf{R}_0^+) = \{(a,d) \mid a \in Act,\ d \in \mathbf{R}_0^+\}$ be the set of *timed actions*.

In the interleaving semantics, a timed event structure progresses through a sequence of timed configurations by occurrences of timed actions. In a timed configuration $TC_1 = (C_1,T_1)$, the *occurrence* of a timed action (a,d) *leads to* a timed configuration $TC_2 = (C_2,T_2)$ (denoted $TC_1 \stackrel{(a,d)}{\longrightarrow} TC_2$), if $TC_1 \longrightarrow TC_2$, $C_2 \setminus C_1 = \{e\}$, $l(e) = a$, and $T_2(e) = d$. The leading relation is extended to a sequence of timed actions from $(Act, \mathbf{R}_0^+)^*$ as follows: $TC \stackrel{(a_1,d_1)}{\longrightarrow} \cdots \stackrel{(a_n,d_n)}{\longrightarrow} TC' \Leftrightarrow TC \stackrel{(a_1,d_1)\ldots(a_n,d_n)}{\longrightarrow} TC'$. The set $L_{ti}(TS) = \{w \in (Act, \mathbf{R}_0^+)^* \mid TC_{TS} \stackrel{w}{\longrightarrow} TC$ for some $TC \in \mathcal{TC}(TS)\}$ is the *ti-language* of TS. As an illustration, consider the *ti*-language of the timed event structure TS shown in Figure 1: $\{\epsilon$, (a,d_1), (b,d_2), $(a,d_3)(b,d_4)$, $(b,d_5)(a,d_6) \mid d_1,d_3 \in [3,5],\ d_2,d_4,d_5 \in [4,5],\ d_6 \in [4,6],\ d_3 \leq d_4,\ d_5 \leq d_6\}$.

Testing equivalences [8] are defined in terms of tests which processes may and must satisfy. Two processes are considered testing equivalent if there is no test that can distinguish them. A test is usually itself a process applied to a process by computing both together in parallel. A particular computation is considered to be successful if the test reaches a designated successful state, and the process guarantees the test if every computation is successful. However, following the papers [1, 6, 11], we use an alternative characterization of the timed testing concept from [3]. In interleaving semantics, a test consists of a timed word and a set of timed actions. A process passes this test if after every execution of the timed word the timed actions are inevitable next.

Definition 4. Let TS and TS' be timed event structures.

- For $w \in (Act, \mathbf{R}_0^+)^*$ and $L \subseteq (Act, \mathbf{R}_0^+)$, TS **after** w **MUST** L iff for all $TC \in \mathcal{TC}$ such that $TC_{TS} \xrightarrow{w} TC$ there exists an $(a, d) \in L$ and $TC' \in \mathcal{TC}$ such that $TC \xrightarrow{(a,d)} TC'$;
- TS and TS' are *timed interleaving test equivalent* (denoted $TS \sim_{tit} TS'$) iff for all $w \in (Act, \mathbf{R}_0^+)^*$ and for all $L \subseteq (Act, \mathbf{R}_0^+)$ the following holds:

$$TS \textbf{ after } w \textbf{ MUST } L \iff TS' \textbf{ after } w \textbf{ MUST } L.$$

As an illustration, consider the timed event structures shown in Figure 2. We have $TS_2 \sim_{tit} TS_3$ but $TS_1 \not\sim_{tit} TS_2$, because TS_1 **after** $(a, 0)(b, 1)$ **MUST** $\{(c, 2)\}$ and $\neg(TS_2$ **after** $(a, 0)(b, 1)$ **MUST** $\{(c, 2)\})$.

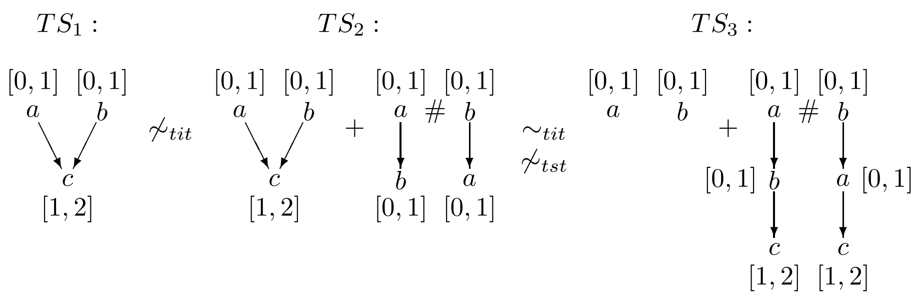

Figure 2

4. Step semantics

In this section, we define a step observation on timed event structures in order to develop timed step testing equivalences. Step semantics generalizes interleaving semantics by allowing timed actions to occur concurrently with themselves. We show that timed step semantics gives a more precise account of concurrency than the timed interleaving one.

Let A be an arbitrary set. A finite multiset M over A is a function $M : A \longrightarrow \mathbf{N}$ such that $\mid \{a \in A \mid M(a) > 0\} \mid < \infty$. Let $\mathcal{M}^{Act}$ denote the set of finite nonempty multisets over Act. We use $(\mathcal{M}^{Act}, \mathbf{R}_0^+) = \{(A,d) \mid A \in \mathcal{M}^{Act},\ d \in \mathbf{R}_0^+\}$ to indicate the set of *timed steps*.

In the step semantics, timed configurations of a timed event structure change, if timed steps from $(\mathcal{M}^{Act}, \mathbf{R}_0^+)$ are executed. In a timed configuration $TC_1 = (C_1, T_1)$, the *execution* of a timed step $(A,d) \in (\mathcal{M}^{Act}, \mathbf{R}_0^+)$ *leads* to a timed configuration $TC_2 = (C_2, T_2)$ (denoted $TC_1 \stackrel{(A,d)}{\longrightarrow} TC_2$), if $TC_1 \longrightarrow TC_2,\ C_2 \setminus C_1 = X,\ \forall\ e, e' \in X \diamond e \smile e',\ l(X) = A,\ \forall e \in X \diamond T_2(e) = d$, where $l(X)(a) = |\{e \in X \mid l(e) = a\}|$. The leading relation is extended to a sequence of timed steps from $(\mathcal{M}^{Act}, \mathbf{R}_0^+)^*$ as follows: $TC \stackrel{(A_1,d_1)}{\longrightarrow} \cdots \stackrel{(A_n,d_n)}{\longrightarrow} TC' \Leftrightarrow TC \stackrel{(A_1,d_1)\ldots(A_n,d_n)}{\longrightarrow} TC'$. The set $L_{ts}(TS) = \{w \in (\mathcal{M}^{Act}, \mathbf{R}_0^+)^* \mid TC_{TS} \stackrel{w}{\longrightarrow} TC$ for some $TC \in \mathcal{TC}(TS)\}$ is the *ts-language* of TS. Considering the timed event structure TS shown in Figure 1, we have $L_{ts}(TS) = \{\epsilon, (\{a\}, d_1), (\{b\}, d_2), (\{a\}, d_3)(\{b\}, d_4), (\{b\}, d_5)(\{a\}, d_6), (\{a,b\}, d_2) \mid d_1, d_3 \in [3,5],\ d_2, d_4, d_5 \in [4,5],\ d_6 \in [4,6],\ d_3 \leq d_4,\ d_5 \leq d_6\}$.

We now come to a definition of timed step testing.

Definition 5. Let TS and TS' be timed event structures.

- For $w \in (\mathcal{M}^{Act}, \mathbf{R}_0^+)^*$ and $L \subseteq (Act, \mathbf{R}_0^+)$, TS **after** w **MUST** L iff for all $TC \in \mathcal{TC}$ such that $TC_{TS} \stackrel{w}{\longrightarrow} TC$ there exists an $(a,d) \in L$ and $TC' \in \mathcal{TC}$ such that $TC \stackrel{(a,d)}{\longrightarrow} TC'$;
- TS and TS' are *timed step test equivalent* (denoted $TS \sim_{tst} TS'$) iff for all $w \in (\mathcal{M}^{Act}, \mathbf{R}_0^+)^*$ and for all $L \subseteq (Act, \mathbf{R}_0^+)$ it holds that

$$TS \textbf{ after } w \textbf{ MUST } L \iff TS' \textbf{ after } w \textbf{ MUST } L.$$

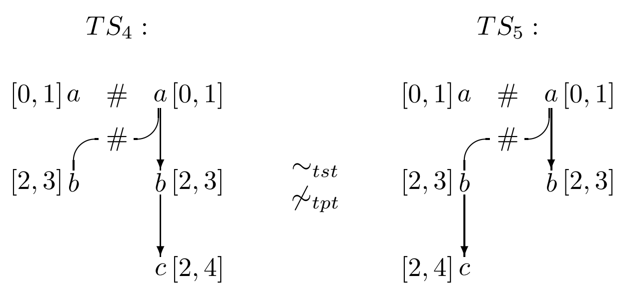

Figure 3

To illustrate the concepts, consider the timed event structures shown in Figures 2 and 3. We have $TS_4 \sim_{tst} TS_5$, but $TS_2 \not\sim_{tst} TS_3$, because

TS_2 **after** $(\{a, b\}, 1)$ **MUST** $\{(c, 2)\}$ and $\neg(TS_3$ **after** $(\{a, b\}, 1)$ **MUST** $\{(c, 2)\})$.

5. Partial order semantics

In this section, we consider some suggestions in order to define a timed testing notion based on partial orders which take into account causality between timed actions.

Define a *timed partial order set* as a timed event structure $TP = (S_{TP} = (E_{TP}, \leq_{TP}, \#_{TP}, l_{TP}), D_{TP})$ such that $\#_{TP} = \emptyset$ and $D_{TP} : E_{TP} \longrightarrow Points$, where $Points = \{[d_1, d_2] \in Interv \mid d_1 = d_2\}$. Isomorphism classes of timed partial order sets are called *timed pomsets*.

We now consider leading relations of the form $\xrightarrow{TP}$, where TP is a timed pomset. For $TC_1 = (C_1, T_1), TC_2 = (C_2, T_2) \in \mathcal{TC}(TS)$, we shall write $TC_1 \xrightarrow{TP} TC_2$, if $TC_1 \longrightarrow TC_2$ and TP is the isomorphism class of $(S_{TS}\lceil(C_2 \setminus C_1)\ , T_2|_{(C_2\setminus C_1)})$. The set $L_{tp}(TS) = \{TP \mid TC_{TS} \xrightarrow{TP} TC$ for some $TC \in \mathcal{TC}(TS)\}$ is the *tp-language* of TS. To illustrate the concept, we consider the *tp*-language of the timed event structure TS shown in Figure 1: $L_{tp}(TS) = \{(\mathcal{O}, 0),\ \overset{[d_1,d_1]}{a},\ \overset{[d_2,d_2]}{b},\ \begin{matrix}\overset{[d_3,d_3]}{a}\\ \overset{[d_2,d_2]}{b}\end{matrix},\ \overset{[d_4,d_4]}{a} \longrightarrow \overset{[d_5,d_5]}{b} \mid d_1, d_4 \in [3, 5],\ d_2, d_5 \in [4, 5],\ d_3 \in [3, 6],\ d_4 \leq d_5\}$.

We are now ready to define partial order testing in the setting of the model under consideration.

Definition 6. Let TS and TS' be timed event structures.

- For a timed pomset TP and $L \subseteq (Act, \mathbf{R}_0^+)$, TS **after** TP **MUST** L iff for all $TC \in \mathcal{TC}$ such that $TC_{TS} \xrightarrow{TP} TC$ there exists an $(a, d) \in L$ and $TC' \in \mathcal{TC}$ such that $TC \xrightarrow{(a,d)} TC'$;
- TS and TS' are *timed pomset test equivalent* (denoted $TS \sim_{tpt} TS'$) iff for all timed pomsets TP and for all $L \subseteq (Act, \mathbf{R}_0^+)$ it holds that

$$TS \textbf{ after } w \textbf{ MUST } L \iff TS' \textbf{ after } w \textbf{ MUST } L.$$

Consider the timed event structures shown in Figures 3 and 4. We have $TS_6 \sim_{tpt} TS_7$, but $TS_4 \not\sim_{tpt} TS_5$, because TS_4 **after** $\overset{[1,1]}{a} \longrightarrow \overset{[2,2]}{b}$ **MUST** $\{(c, 2)\}$ and $\neg(TS_5$ **after** $\overset{[1,1]}{a} \longrightarrow \overset{[2,2]}{b}$ **MUST** $\{(c, 2)\})$.

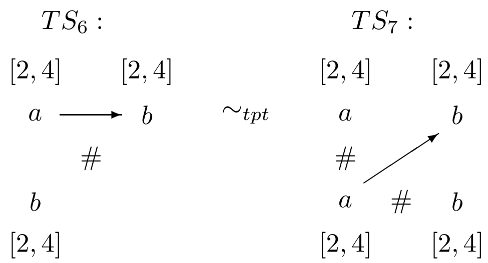

Figure 4

6. Comparison of the equivalences

The common framework used to define different observational equivalences allows us to study the relationships between the three induced semantics. The theorems we state are a step towards a better understanding of the relationships between interleaving, multisets, and partial order semantics. In particular, we will give the hierarchy for the equivalences and will establish that some of them coincide on particular subclasses of timed event structures.

Theorem 1. *Let TS and TS' be timed event structures. Then*

$$TS \sim_{tit} TS' \;\Leftarrow\; TS \sim_{tst} TS' \;\Leftarrow\; TS \sim_{tpt} TS'.$$

Proof. Immediately follows from the definitions of the equivalences.

The timed event structures shown in Figures 2–4 show that the converse implications of the above theorem do not hold and that the three equivalences are all different.

Now one can ask the obvious question: what happens with all these equivalences if we restrict ourselves to some subclasses of the model under consideration. A timed event structure TS is called *sequential*, if $\smile_{TS}= \emptyset$; TS is *deterministic*, if for all $e, e' \in E_{TS}$ it holds that $e \smile_S e'$ or $e \#_S^1 e' \;\Rightarrow\; l(e) \neq_S l(e')$; TS has *correct timing*, if for all $e, e' \in E_{TS}$ it holds that $e \smile_S e'$ or $e \#_S^1 e' \;\Rightarrow\; D_{TS}(e) \cap D_{TS}(e') \neq \emptyset$.

The next theorem shows that if we only consider timed event structures which represent timed sequential processes, then all the three semantics coincide.

Theorem 2. *Let TS and TS' are timed sequential event structures. Then*

$$TS \sim_{tit} TS' \Rightarrow TS \sim_{tst} TS' \Rightarrow TS \sim_{tpt} TS'.$$

Proof. We will show that $TS \sim_{tit} TS'$ implies $TS \sim_{tpt} TS'$. Take an arbitrary timed pomset $TP = (E = \{e_1, \ldots, e_n\}, \leq, l, D)$ $(n \geq 0)$ and $L \subseteq (Act, \mathbf{R}_0^+)$ such that TS **after** TP **MUST** L. First, consider the case when $\smile_{TP} \neq \emptyset$. Since TS' is a timed sequential event structure, it follows that $TP \notin L_{tp}(TS')$. This implies TS' **after** TP **MUST** L. Next, consider the case when $\smile_{TP} = \emptyset$, i.e., $\forall\, 0 \leq i \leq j \leq n \diamond e_i \leq e_j$. Since TS is a sequential timed event structure, this implies that TS **after** w **MUST** L, where $w = (l_{TP}(e_1), D_{TP}(e_1)) \ldots (l_{TP}(e_n), D_{TP}(e_n))$. According to our assumption, we have TS' **after** w **MUST** L. Then TS' **after** TP **MUST** L, because TS' is a timed sequential event structure.

An arbitrary choice of TP and L guarantees $TS \sim_{tpt} TS'$.

Theorem 3. *Let TS and TS' be timed deterministic event structures. Then*

(*i*) $TS \sim_{tit} TS' \iff L_{ti}(TS) = L_{ti}(TS')$;

(*ii*) $TS \sim_{tst} TS' \iff L_{ts}(TS) = L_{ts}(TS')$;

(*iii*) $TS \sim_{tpt} TS' \iff L_{tp}(TS) = L_{tp}(TS')$.

Proof. We will consider the proof of the item (*i*) (the proofs of the remaining items are similar).

($\Rightarrow$) Assume $TS \sim_{ti} TS'$. Let $w \notin L_{ti}(TS)$. Then TS **after** w **MUST** L for all $L \subseteq (Act, \mathbf{R}_0^+)$. This means that TS' **after** w **MUST** L for all $L \subseteq (Act, \mathbf{R}_0^+)$, according to our assumption. Hence, we have $w \notin L_\alpha(TS')$. An arbitrary choice of w guarantees $L_{ti}(TS) = L_{ti}(TS')$.

($\Leftarrow$) We first show that for all $w = (a_1, d_1) \ldots (a_n, d_n) \in L_{ti}(TS)$ $(n \geq 0)$, it holds that if $TC_{TS} \xrightarrow{w} TC = (C, T)$ and $TC_{TS} \xrightarrow{w} TC_1 = (C_1, T_1)$, then $TC_1 = TC$. We will proceed by induction on n.

$n = 0$. Trivial.

$n > 0$. Let $w' = (a_1, d_1) \ldots (a_{n-1}, d_{n-1})$ and $w = w'(a_n, d_n)$. Then, according to the induction hypothesis, there exists only one $\widehat{TC} = (\widehat{C}, \widehat{T}) \in \mathcal{TC}(TS)$ such that $TC_{TS} \xrightarrow{w'} \widehat{TC}$. Hence, we have that $\widehat{TC} \xrightarrow{(a_n,d_n)} TC$ and $\widehat{TC} \xrightarrow{(a_n,d_n)} TC_1$. Take e and e_1 such that $C = \widehat{C} \cup \{e\}$ and $C_1 = \widehat{C} \cup \{e_1\}$. Clearly, $l(e) = l(e_1) = a_n$. Consider all the possible relations between e and e_1. If $e <_{TS} e_1$ ($e >_{TS} e_1$), then C_1 (C) is not a configuration. If $e \smile_{TS} e_1$ or $e \;\#^1_{TS}\; e_1$, then we get a contradiction to the definition of a timed deterministic event structure having correct timing. Hence, we have $TC = TC_1$.

Further, take an arbitrary L and w such that $\neg(TS$ **after** w **MUST** $L)$. According to Definition 3, there exists $TC \in \mathcal{TC}(TS)$ such that $TC_{TS} \xrightarrow{w}$

TC, and for all $(a,d) \in L$ and $TC' \in \mathcal{TC}(TS)$ it holds that $\neg(TC \xrightarrow{(a,d)} TC')$. This implies $w \in L_{ti}(TS)$, and according to our assumption we have $w \in L_{ti}(TS')$. Suppose the contrary, i.e., TS' **after** w **MUST** L. Then, according to Definition 3, there exist $TC', TC'' \in \mathcal{TC}(TS')$ and $(a,d) \in L$ such that $TC_{TS'} \xrightarrow{w} TC' \xrightarrow{(a,d)} TC''$. Consider $w' \in L_{ti}(TS')$ such that $TC_{TS'} \xrightarrow{w'} TC''$. Since $L_{ti}(TS) = L_{ti}(TS')$, $w' \in L_{ti}(TS)$. This means that $TC \xrightarrow{(a,d)} TC'''$ for some $TC''' \in \mathcal{TC}(TS)$, because TC is the unique timed configuration such that $TC_{TS} \xrightarrow{w} TC$, as shown above. Hence, we get a contradiction to $\neg(TS$ **after** w **MUST** $L)$. Thus, $\neg(TS'$ **after** w **MUST** $L)$. An arbitrary choice of w and L guarantees $TS \sim_{\alpha t} TS'$.

The theorem below establishes that if we only consider timed event structures having correct timing, then timed step and timed partial order semantics coincide.

Theorem 4. *Let TS and TS' be timed deterministic event structures which have correct timing. Then*

$$TS \sim_{tst} TS' \;\Rightarrow\; TS \sim_{tpt} TS'.$$

Proof. Let $TS = (S, D)$ and $TS' = (S', D')$. Assume $L_{ts}(TS) = L_{ts}(TS')$. According to Theorem 3, it is sufficient to show that $L_{tp}(TS) = L_{tp}(TS')$.

Take an arbitrary timed pomset $TP \in L_{tp}(TS)$ such that $TC_{TS} \xrightarrow{TP} TC$. W.l.o.g. assume $E_{TP} = \{e_1, \dots, e_n\}$ $(n \geq 0)$ such that $D_{TP}(e_i) \leq D_{TP}(e_j)$ for all $1 \leq i \leq j \leq n$. Let $l_{TP}(e_i) = a_i$ and $D_{TP}(e_i) = d_i$ for all $1 \leq i \leq n$. Then $TC_{TS} = TC_0 \xrightarrow{(\{a_1\},d_1)} TC_1 \cdots TC_{n-1} \xrightarrow{(\{a_n\},d_n)} TC_n = TC$, where $TC_i = (C_i, T_i)$ and $C_j \setminus C_{j-1} = \{e_j\}$ for all $0 \leq i \leq n$ and $1 \leq j \leq n$. Since $w = (\{a_1\}, d_1) \dots (\{a_n\}, d_n) \in L_{ts}(TS)$, $w \in L_{ts}(TS')$, according to the assumption. Hence, we have $TC_{TS'} = TC'_0 \xrightarrow{(\{a_1\},d_1)} TC'_1 \cdots TC'_{n-1} \xrightarrow{(\{a_n\},d_n)} TC'_n = TC'$, where $TC'_i = (C'_i, T'_i)$ and $C'_j \setminus C'_{j-1} = \{e'_j\}$ for all $0 \leq i \leq n$ and $1 \leq j \leq n$.

We will show that $(S \lceil C_n, T_n) \simeq (S' \lceil C'_n, T'_n)$. Three cases are possible.

$n = 0$. Trivial.

$n = 1$. The result follows from the definition of a deterministic timed event structure.

$n > 1$. It sufficies to show that $e_i \smile_S e_j \iff e'_i \smile_{S'} e'_j$ for all $1 \leq i < j \leq n$. Suppose $e_i \smile_S e_j$ for some $1 \leq i < j \leq n$. W.l.o.g. assume $j = i + 1$. Let us prove that $e'_i \smile_{S'} e'_j$.

We will show that there exists $\widetilde{TC}_{i+1} \in \mathcal{TC}(TS)$ such that $TC_{i-1} \xrightarrow{(\{a_i,a_{i+1}\},d)} \widetilde{TC}_{i+1}$ for some $d \in \mathbf{R}_0^+$. Take $\widetilde{C}_{i+1} = C_{i+1}$. Further, take $\widetilde{T}_{i+1} : \widetilde{C}_{i+1} \longrightarrow \mathbf{R}_0^+$ such that $\widetilde{T}_{i+1}|_{C_{i-1}} = T_{i-1}$ and $\widetilde{T}_{i+1}(e_i) = \widetilde{T}_{i+1}(e_j) = d$, where $d \in D(e_i) \cap D'(e_j)$ such that $d_i \leq d \leq d_j$ (the existence of d is guaranteed by the definitions of a deterministic timed event structure having correct timing, the set $Interv$ and the relation $\rightarrow$ on timed configurations). We have to show that $\widetilde{TC}_{i+1} = (\widetilde{C}_{i+1}, \widetilde{T}_{i+1}) \in \mathcal{TC}(TS)$. Since $TC_{i-1} \in \mathcal{TC}(TS)$ and $d \in D(e_i) \cap D(e_j)$, the truth of item (i) of Definition 2 is obvious. Due to the facts that $TC_i \in \mathcal{TC}(TS)$ and $d_i \leq d$, item (ii) of Definition 2 holds, by the definition of the relation $\rightarrow$ on timed configurations. Since $TC_{i+1} \in \mathcal{TC}(TS)$ and $d \leq d_j$, it is straightforward to show the truth of item (iii) of Definition of 2. According to the construction of $\widetilde{TC}_{i+1}$, it holds that $TC_{i-1} \stackrel{(\{a_i, a_{i+1}\}, d)}{\longrightarrow} \widetilde{TC}_{i+1}$. Hence,

$$w' = (\{a_1\}, d_1) \ldots (\{a_{i-1}\}, d_{i-1})(\{a_i, a_j\}, d) \in L_{ts}(TS).$$

Since $L_{ts}(TS) = L_{ts}(TS')$, then $w' \in L_{ts}(TS')$. This implies that $TC'_{i-1} \stackrel{(\{a_i, a_j\}, d)}{\longrightarrow} \widetilde{TC'}_{i+1}$, because TS' is a deterministic timed event structure (see the proof of Theorem 3). Thus, we have $e'_i \smile e'_j$.

We have shown that $(S\lceil C, T) \simeq (S'\lceil C', T')$. Since $(S\lceil C, T) \simeq TP$ it follows that $(S'\lceil C', T') \simeq TP$. This means that $TC_{TS'} \stackrel{TP}{\longrightarrow} TC'$. So, we get $TP \in L_{tp}(TS')$. An arbitrary choice of TP guarantees $L_{tp}(TS) = L_{tp}(TS')$.

The timed event structures in Figure 5 show that even for timed deterministic event structures having correct timing there is a difference between timed interleaving and timed partial order semantics: $TS_8 \sim_{tit} TS_9$, but $TS_8 \not\sim_{tst} TS_9$ because for $w = (\{a, b\}, 1)$ and $L = \emptyset$ we have TS_9 **after** w **MUST** L and $\neg(TS_8$ **after** w **MUST** $L)$.

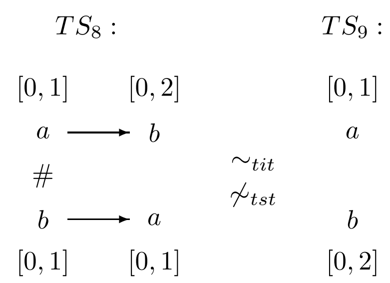

Figure 5

7. Conclusion

In this paper, we have proposed a family of testing equivalences for timed event structures, a model aimed at an explicit representation of concurrency and time. In particular, we have given a flexible abstract mechanism based on the observational techniques studied in [8] which allows us to use timed event structures as the basis of three different approaches to the description of concurrent and real time systems. Apart from giving a concurrency preserving semantics, our approach has an additional advantage of providing a unified framework for the definition of different semantics induced on timed event structures by the equivalences and determines when one semantics is more appropriate than another. The results obtained show that interleaving, multisets and partial ordering semantics in general provide formal tools with an increasing power. Furthermore, when dealing with particular subclasses of the model such as timed sequential and timed nondeterministic processes, there is no difference between a more concrete or a more abstract approach.

References

[1] Aceto L., De Nicola R., Fantechi A. Testing equivalences for event structures // Lect. Notes Comput. Sci. — 1987. — Vol. 280. — P. 1–20.

[2] Alur R., Dill D. The theory of timed automata // Theor. Comput. Sci. — 1994. — Vol. 126. — P. 183–235.

[3] Andreeva M.V., Bozhenkova E.B., Virbitskaite I.B. Analysis of timed concurrent models based on testing equivalence // Fundamenta Informaticae. — 2000. — Vol. 43, N 1–4. — P. 1–20.

[4] Baier C., Katoen J.-P., Latella D. Metric semantics for true concurrent real time // Proc. ICALP'98. — Lect. Notes. Comput. Sci. — 1998. — Vol. 1443. — P. 568–579.

[5] Čerāns K. Decidability of bisimulation equivalences for parallel timer processes // Lect. Notes Comput. Sci. — 1993. — Vol. 663. — P. 302–315.

[6] Cleaveland R., Hennessy. M Testing equivalence as a bisimulation equivalence // Lect. Notes Comput. Sci. — 1989. — Vol. 407. — P. 11–23.

[7] Cleaveland R., Zwarico A.E. A theory of testing for real-time // Proc. LICS'91. — P. 110–119.

[8] De Nicola R., Hennessy M. Testing equiavalence for processes // Theor. Comput. Sci. — 1984. — Vol. 34. — P. 83–133.

[9] van Glabbeek R.J. The linear time — branching time spectrum II: the semantics of sequential systems with silent moves. Extended abstract // Lect. Notes Comput. Sci. — 1993. — Vol. 715. — P. 66–81.

[10] van Glabbeek R.J., Goltz U. Equivalence Notions for concurrent systems and refinement of actions // Lect. Notes Comput. Sci. — 1989. — Vol. 379. — P. 237–248.

[11] Goltz U., Wehrheim H. Causal Testing // Lect. Notes Comput. Sci. — 1996. — Vol. 1113. — P. 394–406.

[12] Katoen J.-P., Langerak R., Latella D., Brinksma E. On specifying real-time systems in a causality-based setting // Lect. Notes Comput. Sci. — 1996. — Vol. 1135. — P. 385–404.

[13] Murphy D. Time and duration in noninterleaving concurrency // Fundamenta Informaticae. — 1993. — Vol. 19. — P. 403–416.

[14] Nicolin X., Sifakis J. An overview and synthesis on timed process algebras // Lect. Notes Comput. Sci. — 1992. — Vol. 600. — P. 526–548.

[15] Steffen B., Weise C. Deciding testing equivalence for real-time processes with dense time // Lect. Notes Comput. Sci. — 1993. — Vol. 711. — P. 703–713.

[16] Valero V., de Frutos D., Cuartero F. Timed processes of timed Petri nets // Lect. Notes Comput. Sci. — 1995. — Vol. 935. — P. 490–509.

[17] Weise C., Lenzkes D. Efficient scaling-invariant checking of timed bisimulation // Lect. Notes Comput. Sci. — 1997. — Vol. 1200. — P. 176–188.

[18] Winskel G. An Introduction to event structures // Lect. Notes Comput. Sci. — 1988. — Vol. 354. — P. 364–397.