Joint NCC & IIS Bull., Comp. Science, 20 (2004), 31–47

Timed testing for dense timed model

E. N. Bozhenkova

Abstract. In this paper, we try to decide a problem of recognising timed testing equivalences for timed event structures with dense internal actions. For this purpose, we construct a formula that characterizes a timed event structure up to the timed *must*-preorder. So, to understand if two timed event structures are in testing relations, it is enough to check if the formula is satisfied.

1. Introduction

Testing equivalences are a class of the major ones [14]. Two processes are considered to be testing equivalent, if there is no test that can distinguish them. A test itself is usually a process applied to another process by computing them together in parallel. A particular computation is considered to be successful, if the test reaches a designated successful state, and the process passes the test if every computation is successful. This notion is intuitively appealing; it has led to a well-developed mathematical theory of processes that ties together the equivalences and preorders [11]. Testing is used to compare two distinct systems, to check conformance of implementation to specification or to check satisfiability of logical formulas [20].

A method of conformance testing consists in the generation of test sequences on the basis of specification and checking them on the implementation system [9, 17, 24]. In this case, it is necessary to make a complete-testing assumption: it is possible, by applying a given test to an implementation system a finite number of times, to pass all possible execution paths of the system that are traversed by the test.

Another approach to testing is to compare two systems with respect to all possible tests. This approach supposes the structure of both systems to be known. Such equivalences have been considered for synchronous and asynchronous formal system models without time delays [1, 11, 10, 14, 15], with time [5, 13, 16, 19, 23] and with probability [12, 18, 21].

Papers [19] and [23] have treated timed testing for dense time transition models. The alternative characterization of timed testing given in these papers uses a notion similar to that of an acceptance set in the testing theory. Paper [23] tries to provide a testing decision procedure that uses the untimed bisimulation between deterministic graphs built from mutually refined time region graphs that are a finite abstraction of the operational semantics of the model under consideration.

Papers [16] and [13] present an investigation of timed testing for the discrete and dense time process algebra. In a model of [13], the time interval $[0, 1]$ is associated with an action. They prove the possibility of discretization in the context of that model and, as a consequence, reduction of dense-timed testing to discrete-timed one. Both papers provide testing decision by reduction to refusal traces.

In [22], Nielsen and Skou present a method for generating a finite and complete set of tests for a special class of timed automata which can be determinized.

This paper is devoted to decidability of timed testing equivalences for timed event structures with dense internal actions. In this model, the causal dependency and nondeterministic choice between events can be naturally expressed. A time interval associated with an event means the interval during which the event can occur. Occurrence of the event does not take any time, and events are not lazy.

As for the characterization and decision procedure of timed testing preorders and equivalences, it turns out that the results mentioned above [23] were not the case for some timed event structures. So, the alternative characterization of the timed testing relations is given in paper [5].

Unlike [13], the proposed discretization is impossible in the case with arbitrary time intervals. So, in [7, 8] the problem of decidability of timed *must*-equivalences is reduced to the model-checking one. As a basic logic, we take the timed logic L_ν. This logic has been defined in [20] and used for construction of a characteristic formula for a timed automaton up to the timed bisimilarity and, as a consequence, for reduction of the timed bisimilarity decidability problem to the model-checking one. It is known that the latter problem is decidable [3, 4]. We construct a characteristic formula up to the timed *must*-preorders. We do this for timed event structures without internal actions and for ones with discrete internal actions.

This paper is devoted to construction of the characteristic formula for timed event structures with dense internal actions. Existence of dense internal actions increases both the number of states reachable by the same word and the number of regions that include such states. So, one of the difficulties in formula construction is organization of the state-space in such a way that all states reachable by one timed word would be collected together. For this purpose, we restrict our model to autodeterministic one, i.e., events labelled by the same action and which are not in causal relation cannot occur at the same time. Also, a modification of a notion of a class is given. Here it is not only τ-closure of regions, but also χ-closure in some cases.

The rest of the paper is organized as follows. In Section 2, we remind the basic notions concerned with timed event structures and timed testing. The timed modal logic L_ν is described in Section 3. In Section 4, we obtain a class graph from the state-space. In Section 5, we construct a formula which characterizes a timed event structure up to the timed *must*-preorders. And in Section 6, we say the concluding words.

2. Timed event structures

In this section, we introduce a model of timed event structures that is a real-time extension of Winskel's model of prime event structures [25] with equipping events with time intervals.

We first recall a notion of an event structure. The main idea behind event structures is to view the distributed computations as action occurrences, called events, together with a notion of the causal dependence between events (which are reasonably characterized via a partial order). Moreover, to model nondeterminism, there is a notion of conflicting (mutually incompatible) events. A labelling function determines which action corresponds to an event.

Let Act be a finite set of visible actions and τ be an internal action. Then $Act_\tau = Act \cup \{\tau\}$.

Definition 1. A (*labelled*) *event structure* over Act_τ 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*: $\forall e \in E\ .\ \{e' \in E \mid e' \leq e\}$ is finite;
- $\# \subseteq E \times E$ is a symmetric and irreflexive relation (*the conflict relation*) satisfying *the principle of conflict heredity*: $\forall e, e', e'' \in E\ .\ e \# e' \leq e'' \Rightarrow e \# e''$;
- $l : E \to Act_\tau$ 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*), $\prec = \leq \setminus (\leq^2 \cup id)$, ${}^\bullet e = \{e' \mid e' \prec e\}$ is a set of immediate predecessors.

Let $C \subseteq E$. Then C is *left-closed* iff $\forall e, e' \in E\ .\ e \in C\ \wedge\ e' \leq e \Rightarrow e' \in C$; C is *conflict-free* iff $\forall e,\ e' \in C\ .\ \neg(e \# e')$; C is a *configuration of* S iff C is left-closed and conflict-free. Let $Conf(S)$ denote the set of all configurations of S. For $C \in Conf(S)$, we define the set of events enabled in C as $En(C) = \{e \in E \mid C \cup \{e\} \in Conf(S)\}$.

In the following, we will consider only finite event structures, i.e., the structures whose sets of events are finite.

Before introducing the concept of a timed event structure, we need to propose some auxiliary notations. Let $\mathbf{N}_0$ be the set of natural numbers with zero, $\mathbf{R}^+$ be the set of positive real numbers, and $\mathbf{R}_0^+$ be the set of nonnegative real numbers. For any $d \in \mathbf{R}_0^+$, $\{d\}$ denotes its fractional part,

$\lfloor d \rfloor$ and $\lceil d \rceil$ — its smallest and largest integer parts, respectively. We define the set $Interv(\mathbf{R}_0^+) = \{[d_1, d_2] \subset \mathbf{R}_0^+ \mid d_1, d_2 \in \mathbf{N}_0\}$.

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

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

- $S = (E, \leq, \#, l)$ is a (labelled) event structure over Act_τ;
- $D : E \to Interv(\mathbf{R}_0^+)$ is a timing function.

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

$$
\begin{array}{lcc}
TS_1 & [0,1]\; a : e_1 \longrightarrow & b : e_2\ [0,1] \\
 & & \# \\
 & & \tau : e_3\ [0,1]
\end{array}
$$

Figure 1. A simple example of a timed event structure

Let $\mathcal{E}_\tau$ denote the set of all labelled timed event structures over Act_τ. For convenience, we fix timed event structures $TS = (S = (E, \leq, \#, l), D)$, $TS' = (S' = (E', \leq', \#', l'), D')$ from the class $\mathcal{E}_\tau$ and work with them further.

A *state* of TS is a pair $M = (C, \delta)$, where $C \in Conf(S)$ and $\delta : E \to \mathbf{R}_0^+$. The *initial state* of TS is $M_{TS} = (C_0, \delta_0) = (\emptyset, 0)$. A state $M = (C, \delta)$ is said to be *terminated*, if $En(C) = \emptyset$. Sometimes we will write $En(M)$ beyond $En(C)$. Let $ST(TS)$ denote the set of all states of TS.

A timed event structure progresses through a sequence of states in one of two ways given below.

Let $M_1 = (C_1, \delta_1), M_2 = (C_2, \delta_2) \in ST(TS)$ such that M_1 is a non-terminated state. An event $e \in En(C_1)$ *may occur* in M_1 (denoted by $M_1 \xrightarrow{e}$) if $\delta_1(e) \in D(e)$ and $\forall e' \in En(C_1)\ \exists d \in \mathbf{R}_0^+ \,.\, \delta_1(e') + d \in D(e)$. We write $M_1 \xrightarrow{a}$, if $M_1 \xrightarrow{e}$ and $l(e) = a$. The occurrence of e in M_1 *leads* to M_2 (denoted by $M_1 \xrightarrow{e} M_2$), if $M_1 \xrightarrow{e}$, $C_2 = C_1 \cup \{e\}$ and

$$\delta_2(e') = \begin{cases} 0, & \text{if } e' \in En(C_2) \setminus En(C_1) \\ \delta_1(e'), & \text{otherwise.} \end{cases}$$

We write $M_1 \xrightarrow{a} M_2$, if $M_1 \xrightarrow{e} M_2$ and $l(e) = a$.

A time $d \in \mathbf{R}^+$ *may pass* in M_1 (denoted by $M_1 \xrightarrow{d}$), if $\forall e \in En(C_1)\ \exists d' \in \mathbf{R}_0^+ (d' \geq d)\ .\ \delta_1(e) + d' \in D(e)$. The passage d in M_1 *leads* to M_2 (denoted $M_1 \xrightarrow{d} M_2$), if $C_2 = C_1$ and $\delta_2(e) = \delta_1(e) + d$ for all $e \in E$.

A *run* r in TS is a sequence of $M_{TS}, M_1, \ldots, M_{n-1}, M_n \in ST(TS)$ and $x_1, \ldots, x_n \in E \cup \mathbf{R}^+$ of the form: $r = M_{TS} \xrightarrow{x_1} M_1 \ \ldots\ M_{n-1} \xrightarrow{x_n} M_n$. Let $R(TS)$ denote the set of all runs in TS. We define the timing duration of a run r as follows: $time(r) = \sum_{1 \leq i \leq n\ \wedge\ x_i \in \mathbf{R}^+} x_i$. For $e \in E$, we let $time(e) = \{time(r) \mid r = M_{TS} \xrightarrow{x_1} M_1 \ \ldots\ M_{n-1} \xrightarrow{x_n} M_n \in R(TS)\ \wedge\ M_n \xrightarrow{e}\}$ (the set of times at which an event e may occur).

The *weak leading relation* $\Rightarrow$ *on the states of* TS is the largest relation defined by: $\overset{\epsilon}{\Rightarrow} \iff \xrightarrow{\tau}{}^*$ and $\overset{x}{\Rightarrow} \iff \overset{\epsilon}{\Rightarrow}\xrightarrow{x}\overset{\epsilon}{\Rightarrow}$, where $\xrightarrow{\tau}{}^*$ is the reflexive and transitive closure of $\xrightarrow{\tau}$ and $x \in Act \cup \mathbf{R}^+$. We consider the relation $\xrightarrow{d}$ as possessing the time continuity property: $M \xrightarrow{d_1+d_2} \iff M \xrightarrow{d_1}\xrightarrow{d_2}$ for some $d_1, d_2 \in \mathbf{R}^+$.

From now on, we will use the following notions and notations. Let $Act(\mathbf{R}_0^+) = \{a(d) \mid \ a \in Act \ \wedge\ d \in \mathbf{R}_0^+\}$ be the set of *timed actions* of Act over $\mathbf{R}_0^+$. Then $(Act(\mathbf{R}_0^+))^*$ is the set of finite *timed words* over $Act(\mathbf{R}_0^+)$. The function $\triangle\ :\ (Act(\mathbf{R}_0^+))^* \to \mathbf{R}_0^+$ measuring the *duration* of a timed word is defined by: $\triangle(\epsilon) = 0$, $\triangle(w.a(d)) = \triangle(w) + d$. The domain for real-time languages is denoted by $Dom(Act,\ \mathbf{R}_0^+) = \{\langle w, d\rangle \mid w \in (Act(\mathbf{R}_0^+))^*,\ d \in \mathbf{R}_0^+,\ d \geq \triangle(w)\}$. The weak leading relation $\Rightarrow$ is extended to timed words from $(Act(\mathbf{R}_0^+))^*$ and $Dom(Act,\ \mathbf{R}_0^+)$ as follows. Let $d \in \mathbf{R}_0^+$, $d' \in \mathbf{R}^+$, $a \in Act$ and $w \in (Act(\mathbf{R}_0^+))^*$. Then

$$\text{if } M \overset{a}{\Rightarrow} M', \text{ then } M \overset{a(0)}{\Rightarrow} M'; \text{ if } M \overset{d'}{\Rightarrow}\overset{a}{\Rightarrow} M', \text{ then } M \overset{a(d')}{\Rightarrow} M';$$
$$\text{if } M \overset{w}{\Rightarrow}\overset{a(d)}{\Rightarrow} M', \text{ then } M \overset{w.a(d)}{\Longrightarrow} M'; \text{ if } M \overset{w}{\Rightarrow} M', \text{ then } M \overset{\langle w,\ \triangle(w)\rangle}{\Longrightarrow} M';$$
$$\text{if } M \overset{\langle w,d\rangle}{\Longrightarrow}\overset{d'}{\Rightarrow} M', \text{ then } M \overset{\langle w,\ d+d'\rangle}{\Longrightarrow} M'.$$

The set $L(TS) = \{\langle w, d\rangle \in Dom(Act, \mathbf{R}_0^+) \mid \ M_{TS} \overset{\langle w,d\rangle}{\Longrightarrow}\}$ is the *language* of TS. For instance, for the timed event structure TS_1 in Figure 1 we have $L(TS_1) = \{\langle \epsilon, d_1\rangle,\ \langle a(d_1), d_1 + d_2\rangle,\ \langle a(d_1)b(d_2), d_1 + d_2\rangle \mid d_1 + d_2 \leq 1\}$.

The timed testing relations may be defined in terms of the responses of timed event structures to a collection of tests. We will, however, use an alternative characterization that relies on the following definitions. Let $M \in ST(TS)$ and $\langle w,\ d\rangle \in Dom(Act,\ \mathbf{R}_0^+)$. Then $S(M) = \{x \in Act_\tau \cup \mathbf{R}^+ \mid \ M \xrightarrow{x}\}$ and $Acc(TS,\ \langle w,\ d\rangle) = \{S(M') \mid M_{TS} \overset{\langle w,\ d\rangle}{\Longrightarrow} M',\ M' \stackrel{\tau}{\not\rightarrow}\}$

(timed acceptance set). Let $N, N' \subset 2^{Act \cup \mathbf{R}^+}$. Then $N' \subset\subset N \iff \forall S' \in N' \; \exists S \in N \; . \; [(S \mid_{Act} \subseteq S' \mid_{Act}) \; \wedge \; (S' \mid_{\mathbf{R}^+} = \emptyset \Rightarrow S \mid_{\mathbf{R}^+} = \emptyset)]$; $N \equiv N' \iff N \subset\subset N' \; \wedge \; N' \subset\subset N$.

Definition 3.

- $TS \leq_{must} TS' \iff \forall \langle w, \; d \rangle \in Dom(Act, \mathbf{R}_0^+) \; . \; Acc(TS', \langle w, d \rangle) \subset\subset Acc(TS, \langle w, d \rangle)$;
- $TS \simeq_{must} TS' \iff TS \leq_{must} TS'$ and $TS' \leq_{must} TS$.

An example of timed *must*-equivalent structures is shown in Figure 2(a). The timed event structures TS_3 and TS_3' shown in Figure 2(b) are not timed *must*-equivalent. Let us consider the timed word $\langle w, d \rangle = \langle a(0.5), 1.5 \rangle \in L(TS_3) \cap L(TS_3')$. We have $Acc(TS_3, \langle w, d \rangle) = \{\{b, c\} \cup (0, 1]\}$, $Acc(TS_3', \langle w, d \rangle) = \{\{b, c\} \cup (0, 1], \{c\}\}$, i.e. $\neg\big(Acc(TS_3', \langle w, d \rangle) \subset\subset Acc(TS_3, \langle w, d \rangle)\big)$.

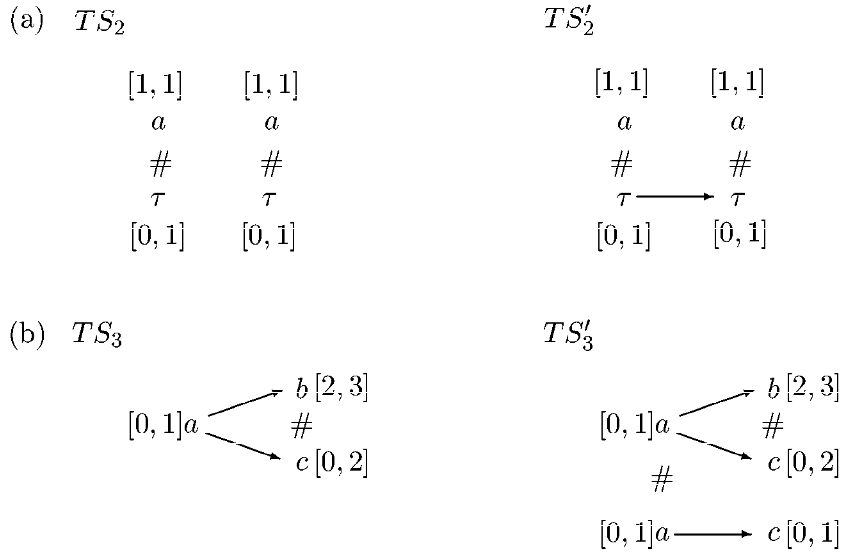

Figure 2. Examples of timed event structures which (a) are *must*-equivalent and (b) are not *must*-equivalent

3. Timed modal logic

Here we will recall a dense-timed logic L_ν [20] and modify a satisfiability relation for timed event structures.

Definition 4. Let K be a finite set of clocks, Id be a set of identifiers and k be an integer. The set of formulas of L_ν over K, Id and k is generated by the abstract syntax with ϕ and ψ ranging over L_ν:

$$\phi := tt \mid ff \mid \phi \wedge \psi \mid \phi \vee \psi \mid \exists\!\!\exists \phi \mid \forall\!\!\forall \phi \mid \langle a \rangle \phi \mid [a]\phi \mid x \text{ in } \phi \mid x + n \bowtie y + m \mid x \bowtie m \mid Z,$$

where $a \in Act$, $x, y \in K$, $n, m \in \{0, 1, \ldots, k\}$, $\bowtie \in \{=, <, \leq, >, \geq\}$ and $Z \in Id$.

The meaning of identifiers from Id is specified by a declaration $\mathcal{D}$ that assigns a formula of L_ν to each identifier. When $\mathcal{D}$ is clear, we write $Z := \phi$ for $\mathcal{D}(\mathcal{Z}) = \phi$. The clocks from K are called formula clocks and a formula ϕ is said to be *closed* if every formula clock x occurs in ϕ in the scope of an "x in ..." operator. Given a timed event structure TS, we interpret the formulas from L_ν over *an extended* state $(C, \delta u)$, where (C, δ) is a state of TS and u is a time assignment for K. Transitions between extended states are defined by: $(C, \delta u) \xrightarrow{\epsilon(d)} (C, (\delta + d)(u + d))$ and $(C, \delta u) \xrightarrow{a} (C', \delta' u')$ iff $(C, \delta) \xrightarrow{a} (C', \delta')$ and $u = u'$. Formally, the satisfaction relation between extended states and formulas is defined just as in [20] and differs from [7] for $\forall\!\!\forall$- and $\exists\!\!\exists$- operators.

Definition 5. Let TS be a timed event structure and D be a declaration. The satisfaction relation $\models_{\mathcal{D}}$ is the largest one that satisfies the following implications:

$$\begin{array}{rcl}
(C, \delta u) \models_{\mathcal{D}} tt & \Rightarrow & \text{true};\\
(C, \delta u) \models_{\mathcal{D}} ff & \Rightarrow & \text{false};\\
(C, \delta u) \models_{\mathcal{D}} \phi \wedge \psi & \Rightarrow & (C, \delta u) \models_{\mathcal{D}} \phi \text{ and } (C, \delta u) \models_{\mathcal{D}} \psi;\\
(C, \delta u) \models_{\mathcal{D}} \exists\!\!\exists \phi & \Rightarrow & \exists d \in \mathbf{R}_0^+ \,.\, (C, \delta) \stackrel{\epsilon(d)}{\Rightarrow} (C', \delta')\\
& & \text{and } (C', \delta' u + d) \models_{\mathcal{D}} \phi;\\
(C, \delta u) \models_{\mathcal{D}} \forall\!\!\forall \phi & \Rightarrow & \forall d \in \mathbf{R}_0^+ (C, \delta) \stackrel{\epsilon(d)}{\Rightarrow} (C', \delta')\\
& & \text{implies } (C', \delta' u + d) \models_{\mathcal{D}} \phi;\\
(C, \delta u) \models_{\mathcal{D}} [a]\phi & \Rightarrow & \forall (C', \delta') \in ST(TS) \,.\, (C, \delta) \xrightarrow{a}\stackrel{\epsilon}{\Rightarrow} (C', \delta')\\
& & \text{implies } (C', \delta' u) \models_{\mathcal{D}} \phi;\\
(C, \delta u) \models_{\mathcal{D}} \langle a \rangle \phi & \Rightarrow & \exists (C', \delta') \in ST(TS) \,.\, (C, \delta) \xrightarrow{a} (C', \delta')\\
& & \text{and } (C', \delta' u) \models_{\mathcal{D}} \phi;\\
(C, \delta u) \models_{\mathcal{D}} x + m \bowtie y + n & \Rightarrow & u(x) + m \bowtie u(y) + n;\\
(C, \delta u) \models_{\mathcal{D}} x \text{ in } \phi & \Rightarrow & (C, \delta u') \models_{\mathcal{D}} \phi, \text{ where } u' = [\{x\} \to 0]u;\\
(C, \delta u) \models_{\mathcal{D}} Z & \Rightarrow & (C, \delta u) \models_{\mathcal{D}} D(Z).
\end{array}$$

Any relation that satisfies the above implications is called a satisfiability relation. We say that TS *satisfies* a closed formula ϕ from L_ν and write

$TS \models_{\mathcal{D}} \phi$, when $(C_0, \delta_0 u) \models_{\mathcal{D}} \phi$ for any u. Note that if ϕ is closed, then $(C, \delta u) \models_{\mathcal{D}} \phi$ iff $(C, \delta u') \models_{\mathcal{D}} \phi$ for any $u, u' \in \mathbf{R}_0^{+K}$.

4. From state-space to class graph

Before constructing a characteristic formula, we need to transform the infinite state-space to its finite representation in such a way that states reachable by the same timed word be collected together in one class. We will do it through this section.

4.1. Region graph

As usually, in order to get a discrete representation of the state-space of a timed event structure, we use the concept of regions (equivalence classes of states) [3]. At the next step, we unite regions into classes and further the characteristic formula will be constructed for classes. But we can get some problems in the process of class construction.

Namely, one of the problems is existence of several regions which contain states reachable by the same timed word. This happens because of both non-deterministism of timed event structures and density of a time interval of internal actions. The case of discrete internal actions has been discovered in [8]. Here we will consider the autodeterministic model with dense internal actions. It turns out that even in such a model we can get different classes containing states reachable by the same timed word. To avoid it, we use the modified notion of a state. We include an additional set of events (*ignored events*), which becomes enabled after execution of internal actions. This set does not influence the progress of a timed event structure from one state to another, i.e., on $\xrightarrow{z}$ on states.

The problem of synchronization of executions in two timed event structures is decided here by including counters into regions of one timed event structure in order to restrict the states of the second one for which a region formula has to be checked.

A timed event structure TS is *deterministic*, if $\forall e, e' \in E$. $(e \# e')$ or $(e \smile e') \Longrightarrow (l(e) \neq l(e'))$ or $(time(e) \cap time(e') = \emptyset)$. Let $\mathcal{DE}_{\mathcal{T}}$ denote the class of timed event structures having the determinacy property.

So, TS_2, TS_2' and TS_3' in Figure 2 are not deterministic. The examples of deterministic timed event structures are shown in Figure 1 and Figure 3.

Further we suppose $TS \in \mathcal{DE}_{\mathcal{T}}$.

Now, a *state* of TS is a triple $M = (C, \delta, I)$, where $I \subset E$. The *initial state* of TS is $M_{TS} = (C_0, \delta_0, I_0) = (\emptyset, 0, \emptyset)$.

Let $M_1 = (C_1, \delta_1, I_1), M_2 = (C_2, \delta_2, I_2) \in ST(TS)$ such that M_1 is a non-terminated state. Then in the process of progressing, I_1 and I_2 are related as follows:

TS_4

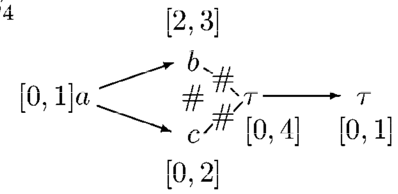

Figure 3. An example of a deterministic timed event structure

if $M_1 \xrightarrow{e} M_2$, then

$$I_2 = \begin{cases} I_1 \cup En(C_2) \setminus En(C_1), & \text{if } l(e) = \tau; \\ I_1 \cup \{e_1 \mid \exists e' \in {}^\bullet e_1 \, . \, l(e') = \tau \wedge \\ sup(D(e')) \neq inf(D(e'))\}, & \text{otherwise;} \end{cases}$$

if $M_1 \xrightarrow{d} M_2$, then $I_2 = I_1$.

Then the notion of a region is defined analogously to Alur's one. Let $M = (C, \delta, I)$, $M_1 = (C_1, \delta_1, I_1) \in ST(TS)$. Then $M \simeq M_1$, if

1. $C = C_1$, $I = I_1$,
2. (i) $\forall 1 \leq i \leq n \, . \, \lfloor \delta(i) \rfloor = \lfloor \delta_1(i) \rfloor$,
   (ii) $\forall 1 \leq i, j \leq n$.
   — $\{\delta(i)\} \leq \{\delta(j)\} \iff \{\delta_1(i)\} \leq \{\delta_1(j)\}$,
   — $\{\delta(i)\} = 0 \iff \{\delta_1(i)\} = 0$,

where $n = \mid E_{TS} \mid$.

A set $R = [M] = \{M_1 \mid M \simeq M_1\}$ is called a *region* of TS. We define $R_0 = [M_{TS}]$.

Let R and R_1 be regions of TS. Then the leading relation on regions is defined as follows:

- $R \xrightarrow{a} R_1$ iff $\exists M \in R,\ M_1 \in R_1 \, . \, M \xrightarrow{a} M_1$ $(a \in Act_\tau)$;
- $R \xrightarrow{\chi} R_1$ iff $\exists M \in R,\ M_1 \in R_1\ \exists d \in \mathbf{R}^+ \, . \, M \xrightarrow{d} M_1 \wedge \forall\ 0 < d' < d\ \ M \xrightarrow{d'} \widetilde{M} \in R \cup R_1$.

We will call a partition of $ST(TS)$ into regions *stable* if the following holds:

- if $R \xrightarrow{a} R_1$, then $\forall M \in R \, . \, M \xrightarrow{a} M_1$ for some $M_1 \in R_1$ $(a \in Act_\tau)$;
- if $R \xrightarrow{\chi} R_1$, then $\forall M \in R\ \exists d \in \mathbf{R}^+ \, . \, M \xrightarrow{d} M_1$ for some $M_1 \in R_1$ and $M \xrightarrow{d'} \widetilde{M} \in R \cup R_1$ for all $0 < d' \leq d$.

So, we can define the notion of a region graph of TS.

Definition 6. *The region graph* of TS is a tuple $RG(TS) = (V_{RG}, E_{RG}, l_{RG})$, where the set of vertices V_{RG} is the stable partition of $ST(TS)$, the set of edges E_{RG} is the leading relation on regions of V_{RG} and the labelling function $l_{RG} : E_{RG} \longrightarrow Act_\tau \cup \{\chi\}$ is defined as: $l(R, R_1) = z \iff R \xrightarrow{z} R'$.

For $R \in V_{RG}$ we define $Der(R,\ z) = \{R' \mid R \xrightarrow{z} R'\}$.

Lemma 1. *Let* $R \in V_{RG}$. *Then* $\forall M,\ M' \in R$

$$S(M) \mid_{Act_\tau} = S(M') \mid_{Act_\tau} \ \wedge\ S(M) \mid_{\mathbf{R}^+} = \emptyset \iff S(M) \mid_{\mathbf{R}^+} = \emptyset.$$

4.2. Adding counters

Let $RG(TS)$ be the region graph and X be a countable set of counters. Let all regions of $RG(TS)$ get a unique number, then with each region R_i we will associate its own counter x_{R_i}. For simplicity, sometimes we will denote x_{R_i} by x_i.

Moreover, with each region R we will associate a triple $(RC(R), M_R, \sigma_R)$, where $RC(R)$ is the set of counters, $M_R = (C, \delta, I) \in R$ is the region representative and the function $\sigma_R : RC(R) \longrightarrow 2^E$ associates the set of events from E with each counter of $RC(R)$.

At first, we suppose $RC(R_0) = \{x_0\}$ and take M_{TS} as a representative of R_0, $\sigma_{R_0}(x_0) = En(C_0)$. For others $R \in RG(TS)$ we suppose $RC(R) = \emptyset$ and take an arbitrary $M \in R$ as its representative, $\sigma_R \equiv \emptyset$. Then the leading relation on regions is modified so that we add x_R into $RC(R)$, if after execution of some non-internal action we get $M = (C, \delta, I) \in R$ and some event becomes enabled in C. Then these events are associated with x_R. More formally:

- $(R, RC(R)) \xrightarrow{z} (R', RC(R'))$ $(z \in Act_\tau)$ iff $R \xrightarrow{z} R'$ (suppose $M_R \xrightarrow{z} \widetilde{M}$ for some $\widetilde{M} \in R'$) and the set $RC(R')$ is modified in two steps:

    1. $RC(R') = RC(R') \cup RC(R)$;
    2. $RC(R') = RC(R') \cup \{x_{R'}\}$ if $\exists e \in En(\widetilde{M}) \setminus En(M_R) \ \wedge\ e \notin \widetilde{I} \wedge \forall e' \in (C \cup En(C)) \setminus \widetilde{I}\ \{\delta(e')\} \neq 0$

    and $\sigma_{R'}$ is modified as follows:

    1. for all $x \in RC(R') \cap RC(R)$ $\sigma_{R'}(x) = \sigma_R(x)$;
    2. if $x_{R'} \in RC(R')$ then $\sigma_{R'}(x_{R'}) = En(\widetilde{C}) \setminus (En(C) \cup \widetilde{I})$;

- $(R, RC(R)) \xrightarrow{\chi} (R', RC(R'))$ iff $R \xrightarrow{\chi} R'$ (suppose $M_R \xrightarrow{d} \widetilde{M}$ for some $d \in \mathbf{R}^+$ and $\widetilde{M} \in R'$) and $RC(R') = RC(R') \cup RC(R)$.

We also need a time assignment of our counters $\Delta_R : RC(R) \to \mathbf{R}_0^+$, which is defined as $\Delta_R(x) = \delta_{M_R}(e)$ for $x \in RC(R)$, where e is taken arbitrary from $\sigma_R(x)$.

In the following, we will use a simple notation R instead of $(R, RC(R))$ and omit the subscript R if it will be clear from context.

4.3. Class graph

We next define the notions of a class and a class graph of a timed event structure.

Let $RG(TS) = (V_{RG}, E_{RG}, l_{RG})$, $R = [M = (C, \delta, I)] \in V_{RG}$, then $\delta\rceil$ denotes $\delta\mid_{E\setminus I}$.

And for regions $R \xrightarrow{\chi} R_1$, we write $R \stackrel{\chi_0}{\Rightarrow} R_1$ iff for all $M = (C, \delta, I) \in R$ and $M_1 = (C_1, \delta_1, I_1) \in R_1$ $(C, \delta\rceil, I) \simeq (C_1, \delta_1\rceil, I_1)$. Otherwise, we write $R \stackrel{\chi_1}{\Rightarrow} R_1$. By this, we distinguish a change of region constraints for events from I.

Let $Q \subseteq V_{RG}$. A set $Q^{\tau\chi} = \{R' \in V_{RG} \mid \exists R \in Q\ .\ R \stackrel{\chi_0}{\Rightarrow} R'\}$ is called a *class* of TS. We define $Q_0 = \{R_0\}^{\tau\chi}$, and $Der(Q,\ z) = \bigcup_{R\in Q} Der(R,\ z)$ for $z \in Act \cup \{\chi_1\}$.

For classes Q, Q_1 and $z \in Act \cup \{\chi_1\}$, the *leading relation on classes* is given by: $Q \xrightarrow{z} Q_1$, if $Q_1 = (Der(Q,\ z))^{\tau\chi}$.

Let us define the following notations.

$S(Q) = \{z \in Act \cup \{\chi_1\} \mid\ Q \xrightarrow{z}\}$, $QC(Q) = \bigcup_{R\in Q} RC(R)$.

Definition 7. The *class graph* of TS is the labelled directed graph $CG(TS) = (V_{CG}, E_{CG}, l_{CG})$. The set of vertices V_{CG} is the set of reachable classes of TS and E_{CG} is the leading relation on classes of the set V_{CG}, the labelling function $l_{CG} : E_{CG} \longrightarrow (Act \cup \{\chi_1\})$.

The class graph of TS_1 is shown in Figure 4.

We need to connect the notions of a state and a class.

Definition 8. Let $\langle w, d\rangle \in L(TS)$ and $CG(TS) = (V_{CG},\ E_{CG},\ l_{CG})$. Let $p = Q_0\ \ldots\ Q$ be a path in $CG(TS)$. Then $M \in ST(TS)$ is *class-reachable by* $\langle w, d\rangle$ *consistent with* p iff $[M] \in Q$ and either

- $p = Q_0$ and $\langle w, d\rangle = \langle \epsilon, 0\rangle$, or
- $p = p' \xrightarrow{z} Q$ and there exists $M' \in ST(TS)$ class-reachable by $\langle w', d'\rangle$ consistent with p', and either
  - $z = a \in Act$, $M' \stackrel{a}{\Rightarrow}\stackrel{d''}{\Rightarrow} M$, and $\langle w, d\rangle = \langle w'a(d' - \Delta(w')), d' + d''\rangle$, for some $d'' \in \mathbf{R}_0^+$, or
  - $z = \chi_1$, $M' \stackrel{d''}{\Rightarrow} M$, and $\langle w, d\rangle = \langle w', d' + d''\rangle$, for some $d'' \in \mathbf{R}^+$.

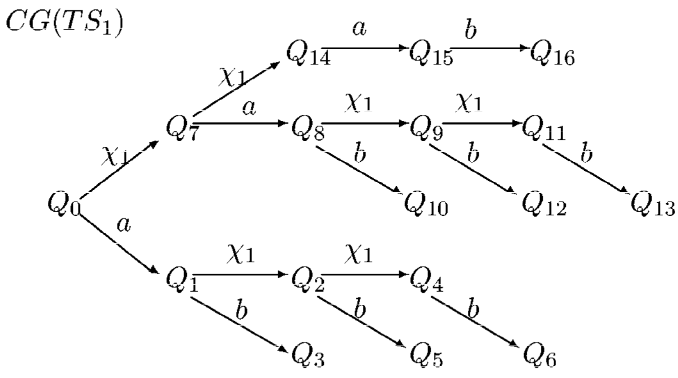

Figure 4. The class graph of TS_1

Lemma 2. *Let* $\langle w, d\rangle \in L(TS)$ *and* $M_{TS} \overset{\langle w,d\rangle}{\Longrightarrow} M$.

(*i*) *Then there exists a (unique) path* p *in* $CG(TS)$ *such that* M *is class-reachable by* $\langle w, d\rangle$ *consistent with* p.

(*ii*) *Let* $p = Q_0 \ \ldots \ Q$ *be a path in* $CG(TS)$ *such that* M *is class-reachable by* $\langle w, d\rangle$ *consistent with* p. *Then*

$$\forall R \overset{\tau}{\not\rightarrow} \in Q \exists M_1 \in ST(TS).\ M_{TS} \overset{\langle w,d\rangle}{\Longrightarrow} M_1 \overset{\tau}{\not\rightarrow}$$

and

$$S(M_1) \mid_{Act} \subseteq S(M_R) \mid_{Act} \ \wedge\ S(M_1) \mid_{\mathbf{R}^+} = \emptyset \iff S(M_R) \mid_{\mathbf{R}^+} = \emptyset.$$

Now, let us suppose we use a definition of a class as only τ-closure [5] and consider what we can get in this situation. Then in the class graph of TS_4 (see Figure 3) there exist vertices Q and Q_1 with regions $[M] \in Q$ ($[M_1] \in Q_1$) such that $M (M_1$, respectively) is reachable by $\langle \epsilon(1), 1\rangle$ consistent with a path from Q_0 to Q (to Q_1, respectively).

We get such a situation because in TS_4 we can execute an action τ and get a new state at time $0 < d < 1$, but also only time d can pass. If only time passes, then the set of values of the function δ in a new state consists of d. If τ is executed at time $0 < d < 1$ then the set of values of function δ in a new state consists of 0 and d. So, there exist several paths in the region graph $RG(TS_4)$ from R_0 to regions that include the states reachable by $\langle \epsilon(1), 1\rangle$. Namely, one of them consists of a sequence of two χ-transitions, the other consists of a sequence of χ-, τ- and two χ-transitions. Thus, by construction of the class graph, these regions belong to different classes.

Since we wish to avoid these cases, we have defined a class as τ- and χ_0-closure.

5. Formula construction

Now we can construct a formula for each class Q. In the formula, we use the notations $Q \xrightarrow{a} Q_a$ and $Q \xrightarrow{\chi_1} Q_\chi$ and write its optional parts in $\langle\!\langle$ and $\rangle\!\rangle$. With each counter $x_i \in QC(Q)$ we associate a formula clock $\hat{x}_i$. In addition, we suppose $\widehat{Q} = \{R \mid R \in Q,\ R \not\xrightarrow{\mathcal{I}}\}$.

$$\begin{aligned} F_Q = &\ \forall\!\!\!\forall \beta(Q) \Rightarrow \psi_Q; \\ \psi_Q = &\ \bigwedge_{a \notin S(Q)|_{Act}} [a]\mathit{ff} \ \wedge\ \bigwedge_{a \in S(Q)|_{Act}} [a](\langle\!\langle XQ_a\ in\rangle\!\rangle\ \widehat{F}_{Q_a}) \ \wedge \\ &\ (ACC(Q)\ \vee\ \langle\tau\rangle \mathit{tt})\ \wedge\ \langle\!\langle \forall\!\!\!\forall \beta^{>}(Q) \Rightarrow F_{nil}\rangle\!\rangle\ \wedge\ \langle\!\langle F_{Q_\chi}\rangle\!\rangle; \end{aligned}$$

(informally, $\psi_Q = \begin{bmatrix}\text{a part for actions that} \\ \text{are not executable in } Q\end{bmatrix} \wedge \begin{bmatrix}\text{a part for actions that} \\ \text{are executable in } Q\end{bmatrix} \wedge$ [simulation of $Acc(TS, \langle w, d\rangle)$] $\wedge$ $\langle\!\langle$ Q_χ does not exist$\rangle\!\rangle$ $\wedge$ $\langle\!\langle$ Q_χ exists$\rangle\!\rangle$);

$$\widehat{F}_Q = \begin{cases} F_Q, & \text{if } \exists R \in Q\ \wedge\ \exists M \in R\ \exists d \in \mathbf{R}^+\ .\ M_R \xrightarrow{d} M, \\ \psi_Q, & \text{otherwise.} \end{cases}$$

(We use ψ_Q as $\widehat{F}_Q$ if there is no possibility to pass time inside regions of Q).

Here the conditions $\beta(Q) = \bigvee_{R \in Q} \beta(R)$ that hold for the time assignment of regions only from Q are constructed in the following way:

1. $\beta(R) = \mathit{tt}$;
2. for all $x_i, x_j (x_i \neq x_j) \in RC(R)$ let $\lfloor \Delta_R(x_i) \rfloor = a$, $\lfloor \Delta_R(x_j) \rfloor = b$, then
$$\beta(R) = \beta(R) \wedge \begin{cases} \hat{x}_i = a, & \text{if } \Delta_R(x_i) = \lfloor \Delta_R(x_i) \rfloor, \\ a < \hat{x}_i < a + 1, & \text{otherwise;} \end{cases}$$
3. $$\beta(R) = \beta(R) \wedge \begin{cases} \hat{x}_i + b = \hat{x}_j + a, & \text{if } \{\Delta_R(x_i)\} = \{\Delta_R(x_j)\}, \\ \hat{x}_i + b < \hat{x}_j + a, & \text{if } \{\Delta_R(x_i)\} < \{\Delta_R(x_j)\}, \\ \hat{x}_i + b > \hat{x}_j + a, & \text{if } \{\Delta_R(x_j)\} < \{\Delta_R(x_i)\}. \end{cases}$$

The conditions $\beta^{>}(Q)$ mean that the values of counters are larger than the appropriate time assignments in regions from Q. For each $R \in Q$, the conditions $\beta^{>}(R)$ are constructed as follows:

$$\beta^{>}(R) = \begin{cases} \beta(R)\ \vee\ \bigvee_{x_i \in RC(R)} \hat{x}_i \geq \lceil \Delta_R(x_i) \rceil, & \text{if } M_R \text{ is terminated,} \\ \bigvee_{\{x_i \in RC(R) \mid \{\Delta_R(x_i)\} = 0\}} \hat{x}_i > \lceil \Delta_R(x_i) \rceil \\ \vee\ \bigvee_{\{x_i \in RC(R) \mid \{\Delta_R(x_i)\} \neq 0\}} \hat{x}_i \geq \lceil \Delta_R(x_i) \rceil, & \text{otherwise.} \end{cases}$$

Below we give the subformulas of ψ_Q and conditions on including them into ψ_Q.

- $XQ_a = \{\hat{x} \mid x \in QC(Q_a) \setminus QC(Q)\}$ is added if it is not empty;
- $\forall\!\!\!\forall \beta^>(Q) \Rightarrow F_{nil}$ is added into ψ_Q if the class Q_χ does not exist;
- F_{Q_χ} is added into ψ_Q if the class Q_χ exists;
- $ACC(Q) = \bigvee_{R\in\hat{Q}}\big((\bigwedge_{a\in S(M_R)}\langle a\rangle tt) \ \wedge \ \langle\!\langle \chi_{M_R}\rangle\!\rangle \ \wedge \ \langle\!\langle F_{nil}\rangle\!\rangle\big)$ simulates the timed acceptance set $Acc(TS, \langle w, d\rangle)$;
- $F_{nil} = \bigwedge_{a\in Act}[a]ff$ is added into $ACC(Q)$ for all $R \in \hat{Q}$ such that $S(M_R) \mid_{Act}= \emptyset$, i.e., no action can be executed in this region;
- $\chi_M = \begin{cases} \exists\beta(Q_\chi) \Rightarrow (\bigwedge_{a\in S(M)}\langle a\rangle tt), & \text{if } S(M) \mid_{Act}\neq \emptyset, \\ \exists\beta^>(Q) \Rightarrow (\bigvee_{a\in Act_\tau}\langle a\rangle tt), & \text{otherwise;} \end{cases}$
- χ_{M_R} is added into $ACC(Q)$ for all $R \in \hat{Q}$ such that $S(M_R) \mid_{\mathbf{R}^+}\neq \emptyset$, i.e., some time can pass in the states from R.

Note that it is easy to transform our formula with implication ($\Rightarrow$) into a correct formula from L_ν, because negation of $\beta(Q)$ and $\beta^>(Q)$ can be expressed in L_ν. Also, XQ_a *in* F means $(\hat{x}_1 \ in \ (\hat{x}_2 \ in \ (\ldots \ (\hat{x}_n \ in \ F)))$ for $XQ_a = \{\hat{x}_1,\ \hat{x}_2,\ \ldots,\ \hat{x}_n\}$. The formula ψ_Q contains three obligatory groups. The first group of conjunctions contains an $[a]$-formula for any action that cannot be executed in Q. The second group of conjunctions contains an $[a]$-formula for any action that can be executed in Q. The third group is a group of disjunctions over all states in $\hat{Q}$ and each disjunction part contains conjunctions of $\langle a\rangle$-formulas for each action that can be executed in some state, and an optional part which characterizes the possibility of some amount of time to pass in this state. Existence of the class Q_χ influences the inclusion of optional groups of ψ_Q.

For a timed event structure TS, *a characteristic must-formula* is defined as $F^{must}_{TS} = \hat{x}_0 \ in \ F_{Q_0}$.

Below, we give a part of a characteristic *must*-formula for TS_1 in Figure 1. For simplicity, we suppose here $Act = \{a,\ b\}$.

$$
\begin{aligned}
F^{must}_{TS_1} &= \hat{x}_0 \ in \ \Big(\forall\!\!\!\forall \ \hat{x}_0 = 0 \Rightarrow \big[[b]ff \wedge [a]\psi_{Q_1} \wedge (ACC_a \vee \langle\tau\rangle tt) \wedge F_{Q_7}\big]\Big),\\
\psi_{Q_1} &= [a]ff \ \wedge \ [b]\psi_{Q_3} \ \wedge \ (ACC_0 \ \vee \ \langle\tau\rangle tt) \ \wedge \ F_{Q_2},\\
\psi_{Q_3} &= [a]ff \ \wedge \ [b]ff \ \wedge \ (ACC_0 \ \vee \ \langle\tau\rangle tt) \ \wedge \ (\forall\!\!\!\forall \ \hat{x}_0 \geq 0 \Rightarrow F_{nil}),\\
F_{Q_7} &= \forall\!\!\!\forall \ 0 < \hat{x}_0 < 1 \Rightarrow \big[[b]ff \wedge [a] \ \hat{x}_1 \ in \ \psi_{Q_8} \wedge (ACC_a \vee \langle\tau\rangle tt) \wedge F_{Q_{14}}\big],\\
ACC_0 &= F_{nil},\\
ACC_a &= \langle a\rangle tt,\\
F_{nil} &= [a]ff \ \wedge \ [b]ff.
\end{aligned}
$$

Lemma 3. *Let $TS \in \mathcal{DE}_\tau$, $TS' \in \mathcal{E}_\tau$. Let $(C'_0, \delta'_0 \ u) \models_D F^{must}_{TS}$, where $(C'_0, \delta'_0) = M_{TS'}$, $u \equiv 0$. For all $\langle w,\ d\rangle \in L(TS) \cap L(TS')$ and $(C'_0, \delta'_0) \stackrel{\langle w,\ d\rangle}{\Longrightarrow} (C', \delta')$ it holds that $(C', \delta' \ u') \models_D \psi_Q$, where Q and u' are such that there* *exists M which is class-reachable by $\langle w,\ d\rangle$ consistent with a path from Q_0 to Q in the class graph of TS, and $u' \mid_{RC([M])} \equiv \Delta_{[M]}$.*

Lemma 4. *Let $TS \in \mathcal{DE}_\mathcal{T}$, $TS' \in \mathcal{E}_\mathcal{T}$. Let $(C'_0, \delta'_0\ u) \models_\mathcal{D} F_{TS}^{must}$, where $(C'_0, \delta'_0) = M_{TS'}$, $u \equiv 0$. Then $L(TS') \subseteq L(TS)$.*

Now, we are ready to prove the following theorem.

Theorem 1. *Let $TS \in \mathcal{DE}_\mathcal{T}$, $TS' \in \mathcal{E}_\mathcal{T}$. $TS \leq_{must} TS' \iff TS' \models_\mathcal{D} F_{TS}^{must}$, where $\mathcal{D}$ corresponds to the previous definition of F_Q for each Q from $V_{CG(TS)}$.*

Proof sketch.
($\Leftarrow$) Take an arbitrary $\langle w,\ d\rangle \in L(TS')$ and (C', δ') such that $(C'_0, \delta'_0) \overset{\langle w,\ d\rangle}{\Longrightarrow} (C', \delta')$. According to Definition 3, we will show that there exists $(C, \delta) \in ST(TS)$ such that $(C_0, \delta_0) \overset{\langle w,\ d\rangle}{\Longrightarrow} (C, \delta)$ and $S(C, \delta) \mid_{Act} \subseteq S(C', \delta') \mid_{Act}$, $S(C', \delta') \mid_{\mathbf{R}^+} = \emptyset \Rightarrow S(C, \delta) \mid_{\mathbf{R}^+} = \emptyset$.

By Lemma 4, $\langle w,\ d\rangle \in L(TS)$. Then using Lemma 2($i$) we can find $M \in ST(TS)$, Q in $CG(TS)$ with a path p from Q_0 to Q and u' such that M is class-reachable by $\langle w,\ d\rangle$ consistent with p, $u' \mid_{RC([M])} = \Delta_{[M]}$. By Lemma 3, $(C', \delta'\ u') \models_D \psi_Q$. By construction of the formula ψ_Q, Lemma 2(ii) and Lemma 1, there exists $R \in Q$, $R \not\xrightarrow{\tau}$, for which $S(M_R) \mid_{Act} \subseteq S(C', \delta') \mid_{Act} \wedge S(C', \delta') \mid_{\mathbf{R}^+} = \emptyset \Rightarrow S(M_R) \mid_{\mathbf{R}^+} = \emptyset$.
($\Rightarrow$) Follows from construction of the formula F_{TS}^{must}, since we stepwise check the requirement on timed acceptance sets. □

6. Conclusion

This article is concentrated on constructing a characteristic formula for deterministic timed event structures with dense internal actions. This formula allows us to decide the problem of recognizing the timed *must*-equivalences by reducing it to the model-checking one. In general, we can use a usual timed event structure from $\mathcal{E}_\mathcal{T}$ as a model on which we check formulas. Restriction to the deterministic model is necessary for formula construction.

To reduce the number of regions and classes, an attempt of using different techniques ([2], [6]) can be made. Besides, the way of construction of the characteristic formula may be applied to other timed testing equivalences, for example, to *may*-equivalences. This can lead to a decision of the inclusion problem for timed languages of a model under consideration.

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. Timed automata// Lect. Notes Comput. Sci. — 1999. — Vol. 1633. — P. 8–22.

[3] Alur R., Courcoubetis C., Dill D. Model checking in dense real time // Inform. and Comput. — 1993. — Vol. 104. — P. 2–34.

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

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

[6] Behrmann G., Larsen K.G., Pearson J., Weise C., Yi W. Efficient timed reachability analysis using clock difference diagrams // Lect. Notes Comput. Sci. — 1999. — Vol. 1633. — P. 341–353.

[7] Bozhenkova E.N. Towards decidability of timed testing // Joint NCC&IIS Bull. Ser.: Comput. Sci. — Novosibirsk, 2001. — Is. 15. — P. 17–29.

[8] Bozhenkova E.N. Timed testing for models with internal actions // Joint NCC&IIS Bull., Comput. Science. — Novosibirsk, 2002. — Is. 18. — P. 27–43.

[9] Brinksma E. Formal methods for conformance testing: theory can be practical // Lect. Notes Comput. Sci. — 1999. — Vol. 1633. — P. 44–46.

[10] Castellani I., Hennessy M. Testing theories for asynchronous langugages // Lect. Notes Comput. Sci. — 1998. — Vol. 1530. — P. 90–101.

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

[12] Cleaveland R., Lee I., Lewis P.M., Smolka S.A. A theory of testing for soft real time processes // Proc. 8th Int. Conf. Software Engineering and Knowledge Engineering, SEKE'96, Lake Tahoe, Nevada, USA, June 1996. — P. 474–479.

[13] Corradini F., Vogler W, Jenner L. Comparing the worst-case efficiency of asynchronous systems with PAFAS. — Augsburg, 2000. — (Tech. Rep. / Inst. für Informatik / Univ. of Augsburg; N 2000-6).

[14] De Nicola R., Hennessy M. Testing equivalence for processes // Theoretical Comput. Sci. — 1984. — Vol. 34. — P. 83–133.

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

[16] Hennessy M., Regan T. A process algebra for timed systems // Inform. and Comput. — 1995. — Vol. 117. — P. 221–239.

[17] Khoumsi A. A method for testing the conformance of real time systems // Lect. Notes Comput. Sci. — 2002. — Vol. 2469. — P. 331–251.

[18] Kumar K.N., Cleaveland R., Smolka S.A. Infinite probabilistic and non-probabilistic testing // Lect. Notes Comput. Sci. — 1998. — Vol. 1530. — P. 209–220.

[19] Llana L., de Frutos D. Denotational semantics for timed testing // Lect. Notes Comput. Sci. — 1997. — Vol. 1231. — P. 368–382.

[20] Laroussinie F., Larsen K.L., Weise C. From timed automata to logic — and back. — Århus, 1995. — (Tech. Rep. / BRICS, Dept. Comput. Sci., Univ. of Århus; N RS-95-2). — Accesible through http://www.brics.aau.dk/BRICS.

[21] López N., Núñez M. A testing theory for generally distributed stochastic processes // Lect. Notes Comput. Sci. — 2001. — Vol. 2154. — P. 321–335.

[22] Nielsen B., Skou A. Automated test generation from timed automata // Lect. Notes Comput. Sci. — Berlin etc., 2001. — Vol. 2031. — P. 343–357.

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

[24] Tretmans J. Test generation with input, outputs and quiscene // Lect. Notes Comput. Sci. — 1996. — Vol. 1055. — P. 127–146.

[25] Winskel G. An introduction to event structures // Lect. Notes Comput. Sci. — 1989. — Vol. 354. — P. 364–397.