Decidability of Bisimulation Equivalences for Parallel Timer Processes

Kārlis Čerāns
Institute of Mathematics and Computer Science,
University of Latvia
Riga, Rainis blvd. 29, Latvia LV-1050
E-mail: karlis%cs.lu.riga.lv@ussr.eu.net

Abstract. In this paper an abstract model of parallel timer processes (PTPs), allowing specification of temporal quantitative constraints on the behaviour of real time systems, is introduced. The parallel timer processes are defined in a dense time domain and are able to model both concurrent (with delay intervals overlapping on the time axis) and infinite behaviour. Both the strong and weak (abstracted from internal actions) bisimulation equivalence problems for PTPs are proved *decidable*. It is proved also that, if one provides the PTP model additionally with memory cells for moving timer value information along the time axis, the bisimulation equivalence (and even the vertex reachability) problems become undecidable.

1 Introduction

The problem of specification of quantitative timed aspects of real time systems has been widely studied over the last years. This research has resulted in a number of timed specification formalisms covering various aspects of real time system specification process, for some impression of what has been done one can see, for example, [MF76, GMMP89, CC88, AD90, ABBCK91, RR86, Wan91]. There are also a lot of interesting results devoted to the *analysis* of quantitative timed behaviour of real time systems. One can recall here at first the enumerative approach to the Time Petri net [MF76] analysis in [BM83] (actually showing the decidability of the reachability problem for bounded Time Petri Nets). In [AD90] it is showed that for Timed Büchi automata the language emptiness problem is decidable whilst the language inclusion problem is not (it is easy to extend this result also to show the undecidability of the language equivalence problem). A model checking algorithm for branching time temporal logic formulae over Timed Graphs is given in [ACD90].

In [ABBCK91, Cer92a] the analysis automation possibilities (decidability of reachability, complete branch covering, finite and infinite path feasibility problems) are investigated for r.t.s. with dependencies on both quantitative timing constraints and external (integer-valued) data.

All the abovementioned timed specification formalisms are based on the assumption of the *density* of the used space of time moments (time domain) when the action or transition firing is allowed. In this paper we also consider real time systems over dense time domains; in particular the obtained results apply to the domain of *rational* numbers (the discrete-time constraints can often be handled by standard FSM analysis techniques due to the finiteness of the generated "state space").

The abovementioned positive analysis results can be considered as dealing with some kind of "extended reachability" problems for the timed system specifications. The intention of this paper is to find whether decidable can be showed any nontrivial algorithmic problem concerning the *equivalence* properties of r.t.s.. One can mention some already existing work on deciding *bisimulation* equivalence for timed processes, however, the obtained results apply only to rather simple cases (in [HLW91] the bisimulation equivalence has been shown decidable for *regular* (in fact, one-timer) real timed processes and in [Che91] the decidability is obtained for *recursion-free* processes).

The main point of this paper is to prove the *decidability* of both strong and weak (abstracted from internal actions) bisimulation equivalence problems for a class of timed processes with both possibly infinite behaviour and time constraints naturally representing overlapping delays in process components (it seems that these requirements altogether provide the minimum level of specification power needed in more or less practical examples (the specification language SDL [CC88], which is widely used in the practical specification of telecommunication systems, also contain means for quantitative time constraint specification of exactly this kind)). We study the deciding of the bisimulation equivalences in the formalism of Parallel Timer Processes (PTPs), see Section 2 for definitions. The PTP formalism is similar to already considered model of Timed Graphs [ACD90] (or, rather, Action Timed Graphs [NSY91]), however, it differs in some design decisions (use of decreasing timers vs. increasing clocks, well-defined (time-stop free) labelled transition system semantics for all the class of PTPs, explicit firing enforcement of transitions along some edges) borrowed to some extent both from the calculi Timed CCS [Wan90] and the specification language SDL.

We obtain also some *undecidability* results (see Section 4) for PTPs provided additionaly with memory cells for moving the timer value information along the time axis, so showing also the difficulties in analysis of processes in Timed CCS with the expansion theorem [Wan91]. In the conclusions some brief points about the compositionality and possible enrichments of PTPs are given.

This paper is a generalisation of a previous author's work [Cer91], where only the strong bisimulation equivalence for an analogical (slightly weaker) specification model was considered and proved decidable. For a more detailed treatment of the problems addressed here the reader can see [Cer92b] (with the reported decidability results slightly weaker) and [Cer92c].

2 Parallel Timer Processes

Let $G = \langle V, E, L, lab \rangle$ be a finite edge-labelled graph with the set of *vertexes* V, the set of *edges* E, the set of *labels* L and the edge labelling function $lab : E \rightarrow L$. For every $e \in E$ let $start(e) \in V$ and $end(e) \in V$ denote the *source* and *target* vertexes of e respectively. Let every edge $e \in E$ be coloured either *red* (instantaneous) or *black* (possibly waiting).

Given such a graph G and a finite set of timers (time variables) $\mathcal{T}$, we define a *timer automaton* by associating with every $e \in E$:

- a set $\gamma(e) \subseteq \mathcal{T}$ of timers, called the edge *condition* (on what timers the transitions along e depend) and
- a *timer setting function* $\phi(e) : \mathcal{T} \rightarrow \mathcal{T} \cup Q^{+0}$.

For $\Phi = \langle V, E, L, lab, \mathcal{T}, \gamma, \phi \rangle$ being a timer automaton we define the set of its *states* to be

$$\mathcal{S}^{\Phi} = \{\langle v, \delta \rangle \mid v \in V,\ \delta : \mathcal{T} \rightarrow Q^{+0}\}.$$

The *Parallel Timer Process* (PTP, for short, called also *timed process*) is defined as a pair $P = \langle \Phi, s \rangle$ for the timer automaton Φ and $s \in \mathcal{S}^{\Phi}$ defined to be the P *initial state*. We denote the set of all PTPs by $\mathcal{P}$, let P, Q range over $\mathcal{P}$ and let σ range over L.

The semantics of PTPs is given by labelled transition systems, based on the relations $\xrightarrow{\sigma}$ and $\xrightarrow{\epsilon(d)}$ with $d \in Q^{+0}$ between processes (the label σ is interpreted as the action, performed by the process; the interpretation of $P \xrightarrow{\epsilon(d)} Q$ is, as in [Wan90], that the process P can become Q just by letting time to pass for d units).

For the timer automaton $\Phi = \langle V, E, L, lab, \mathcal{T}, \gamma, \phi \rangle$ define $\langle \Phi, \langle v, \delta \rangle \rangle \xrightarrow{\sigma} \langle \Phi, \langle v', \delta' \rangle \rangle$ iff there exists an edge $e \in E$ leading from v to v', labelled with $lab(e) = \sigma$, such that

- $\delta(t_i) = 0$ for every $t_i \in \gamma(e)$ (a transition along an edge is enabled when all timers this edge depends on have reached 0 values) and
- for every timer $t \in \mathcal{T}$ its new value $\delta'(t)$ is computed via the setting $\phi(e)$ in a way:
  - if $\phi(e)(t) = c \in Q^{+0}$, then $\delta'(t) = c$,
  - if $\phi(e)(t) = t' \in \mathcal{T}$, then $\delta'(t) = \delta(t')$.

For delay transitions, $\langle \Phi, \langle v, \delta \rangle \rangle \xrightarrow{\epsilon(d)} \langle \Phi, \langle v, \delta' \rangle \rangle$ iff

- for every red edge e with $start(e) = v$ there exists $t_i \in \gamma(e)$ with $\delta(t_i) \geq d$ (no red edge will be enabled during the waiting of d seconds) and
- for every $t \in \mathcal{T}$ $\delta'(t) = \delta(t) \ominus d$, where $x \ominus y \overset{def}{=} max\{0, x - y\}$ for every x, y (all timer values are synchronously decreasing downto 0).

The Parallel Timer Processes obey the following useful semantical properties:

- *time determinacy* ([Wan90]) meaning that, if $P \xrightarrow{\epsilon(d)} P'$ and $P \xrightarrow{\epsilon(d)} P''$, then $P' = P''$;
- *time continuity* ([Wan90]) meaning that $P \xrightarrow{\epsilon(d+e)} P'$ if and only if $P \xrightarrow{\epsilon(d)} P'' \xrightarrow{\epsilon(e)} P'$ for some P'';

- *time-stop freeness* (this property is similar to the deadlock-freeness considered in [NSY91]) meaning that for every PTP P always either

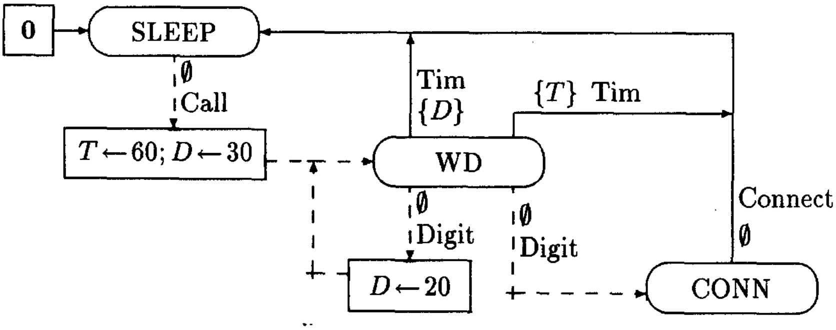
$$P \xrightarrow{\epsilon(d)} P(d) \text{ for all } d \in Q^{+0}, \text{ or } P \xrightarrow{\epsilon(d)} P' \xrightarrow{\sigma} P'' \text{ for some } d \in Q^{+0}, \sigma \in L.$$

2.1 Example: Dialling Timing Control

We describe as a PTP a toy version of a process controlling in telephone exchanges the timing aspects of phone number dialling by subscribers. We assume that a phone number can be any nonempty sequence of digits, dialled by the subscriber with some time intervals in between. The duty of the timing control process is to interrupt the number dialling in any of the following three cases:

- the first digit of the number does not arrive in 30 seconds after the beginning of the dialling (picking up the receiver);
- the current digit which is not the first does not arrive in 20 seconds after the arrival of the previous digit; or
- the total time delay from the beginning of the number dialling reaches 60 seconds.

In the process the edge label "Call" denotes the beginning of the number dialling, "Digit" denotes the receiption of the current digit, "Tim" stands for the dialling interruption and "Connect" initiates the connection seeking process between two subscribers, the process itself is depicted, as follows (we show the black edges of the process as dashed):



2.2 Bisimulations

Let $\Lambda \overset{def}{=} L \cup \{\epsilon(d) | d \in Q^{+0}\}$ be the set of all actions ranged over by ν.

We define the strong *timed bisimulation equivalence* in the set $\mathcal{P}$ of timed processes following [HLW91]:

Definition 2.1 *Let $F(R)$ be the set of all $\langle P, Q\rangle \in \mathcal{P} \times \mathcal{P}$ satisfying*

i) whenever $P \xrightarrow{\nu} P'$ then $Q \xrightarrow{\nu} Q'$ with $\langle P', Q'\rangle \in R$ for some Q',
ii) whenever $Q \xrightarrow{\nu} Q'$ then $P \xrightarrow{\nu} P'$ with $\langle P', Q'\rangle \in R$ for some P'.

Then R is a timed bisimulation, if $R \subseteq F(R)$. We define the timed bisimulation equivalence, written $\sim$, to be the greatest fixpoint of F.

Theorem 2.2 *There is an algorithm which, given two Parallel Timer Processes A and B, decides whether $A \sim B$ or not.*

We consider also the possibility to abstract from internal actions when observing a system modelled by a PTP. For this purpose we assume that every process can have a special (internal, invisible) action (label) $\tau \in L$, let $Vis = L \setminus \{\tau\}$ be ranged over by α.

Let $P \overset{\epsilon}{\Longrightarrow} Q$ if and only if $P(\xrightarrow{\tau})^* Q$. Define $P \overset{\alpha}{\Longrightarrow} Q$ as $P \overset{\epsilon}{\Longrightarrow} \xrightarrow{\alpha} \overset{\epsilon}{\Longrightarrow} Q$. Following [Wan90], for delay transitions let $P \overset{\epsilon(d)}{\Longrightarrow} Q$ whenever

$$P \overset{\epsilon}{\Longrightarrow} \xrightarrow{\epsilon(d_1)} \overset{\epsilon}{\Longrightarrow} \cdots \xrightarrow{\epsilon(d_k)} \overset{\epsilon}{\Longrightarrow} Q$$

for some $d_1, d_2, \ldots, d_k$ with $d_1 + d_2 + \cdots d_k = d$.

Letting ν to range over $Vis \cup \{\epsilon(d) | d \in Q^{+0}\}$ we define the weak timed bisimulation for PTPs (observe that $P \overset{\epsilon}{\Longrightarrow} Q$ iff $P \overset{\epsilon(0)}{\Longrightarrow} Q$):

Definition 2.3 *Let $F(R)$ be the set of all $\langle P, Q \rangle \in \mathcal{P} \times \mathcal{P}$ satisfying*

i) whenever $P \overset{\nu}{\Longrightarrow} P'$ then $Q \overset{\nu}{\Longrightarrow} Q'$ with $\langle P', Q' \rangle \in R$ for some Q',
ii) whenever $Q \overset{\nu}{\Longrightarrow} Q'$ then $P \overset{\nu}{\Longrightarrow} P'$ with $\langle P', Q' \rangle \in R$ for some P'.

Then R is a weak timed bisimulation, if $R \subseteq F(R)$. We define the weak timed bisimulation equivalence, written $\approx$, to be the greatest fixpoint of F.

Theorem 2.4 *There is an algorithm which, given two Parallel Timer Processes A and B, decides whether $A \approx B$ or not.*

Theorem 2.2 follows from Theorem 2.4 in the case of no edges labelled with the τ action.

Let for arbitrary timed process $P = \langle \Phi, \langle v, \delta \rangle \rangle$ $\tilde{d}(P)$ be the set of processes $\langle \Phi, \langle v', \delta' \rangle \rangle$ with v' being a vertex in Φ graph and $\delta'(t) \leq c^P$ for every P timer t, where c^P is defined to exceed both all timer values from δ and all constants used in Φ edge timer settings. It is easy to see that every P derivative (i.e. every process which can be reached during the execution of P) falls into the set $\tilde{d}(P)$ (the converse might not be true).

Without loosing generality in deciding whether $A \sim B$, $A \approx B$ one can consider $\sim$ and $\approx$ to be the maximal bisimulations in the set $\tilde{d}(A) \times \tilde{d}(B) \subseteq \mathcal{P} \times \mathcal{P}$ (observe that a process does not change its graph when executed).

In this paper we do not consider yet another interesting equivalence which is based on the abstraction from the actual time interval length in the semantical transition relation between the processes. In fact, the deciding procedure for this "time-abstracted equivalence" appears to be even simpler than the algorithms for the "time-sensitive" equivalences considered here, this procedure can be based on a rather direct comparing of "region graphs" (see [ACD90]) of the processes under the test.

3 Deciding of Bisimulations

Without the loss of generality we assume that all the explicit constants $c \in Q^{+0}$, used in the edge timer settings $\phi(e)$ in the graphs of A and B, are *integers* (were it not so one could change the scale of the number line to ensure it; easy to see that the behaviour of the processes is not affected by the scale change).

In order to decide whether $A \sim B$, $A \approx B$ we give an effective characteristic (via a finite partitioning) of all bisimilar process pairs within $\tilde{d}(A) \times \tilde{d}(B)$ (it turns out to be too rough for the proof to consider partitionings of $\tilde{d}(A)$ and $\tilde{d}(B)$ independently: the analogue of the proof cornerstone Lemma 3.7 does not hold for *any* nontrivial partitioning of $\tilde{d}(A) \times \tilde{d}(B)$ obtained as the product of independent partitionings of $\tilde{d}(A)$ and $\tilde{d}(B)$).

For $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ being a finite set of timers, let us represent every $\mathcal{T}$ timer value assignment $\delta : \mathcal{T} \rightarrow Q^{+0}$ as the vector $\langle \delta(t_1), \ldots, \delta(t_m) \rangle \in (Q^{+0})^m$. Let $\cong$ be an equivalence relation in the set $\Delta_{\mathcal{T}}$ of $\mathcal{T}$ timer value assignments s.t. $\delta^1 \cong \delta^2$ iff

- $\lfloor \delta^1(t_i) \rfloor = \lfloor \delta^2(t_i) \rfloor$ for every $i = 1, \ldots, n$ and
- for every i, j $\{\delta^1(t_i)\} \geq \{\delta^1(t_j)\}$ if and only if $\{\delta^2(t_i)\} \geq \{\delta^2(t_j)\}$, and
  $\{\delta^1(t_i)\} = 0$ if and only if $\{\delta^2(t_i)\} = 0$

(here $\lfloor x \rfloor$ denotes the "integral part" of x, i.e. the largest integer, which is not greater than x, and $\{x\}$ stands for the fractional part of x (i.e. $\{x\} = x - \lfloor x \rfloor$)).

Given $\delta \in \Delta_{\mathcal{T}}$, let us denote the equivalence class $C \subseteq \Delta_{\mathcal{T}}$ w.r.t. $\cong$ with $\delta \in C$ by $C(\delta)$ and call it the *time region* of $\mathcal{T}$, corresponding to δ. Given a timer value assignment one can easily compute its corresponding time region (one can use the time region representations, say, by linear inequality systems in order to make all computations with them effective).

Example 3.1 *If* $\mathcal{T} = \{t_1, t_2, \ldots, t_7\}$ *and* $\delta = \langle 0.7, 1, 1.23, 4, 17.23, 17.75, 17.75 \rangle$, *then the time region* $C(\delta)$ *can be described as the inequality system*
$C(\delta) = (0 < t_1 < 1 = t_2 < t_3 < 2 < 4 = t_4 < 17 < t_5 < t_6 = t_7 < 18,$
$0 = \{t_2\} = \{t_4\} < \{t_3\} = \{t_5\} < \{t_1\} < \{t_6\} = \{t_7\})$.

Clearly, if for every $t \in \mathcal{T}$ and every timer value assignment $\delta \in \Delta \subseteq \Delta_{\mathcal{T}}$ always $\delta(t) \in [0, c] \subseteq Q^{+0}$, then the set of the corresponding time regions $\{C(\delta) \mid \delta \in \Delta\}$ is *finite*.

The presented time region construction is actually the same, as used in [ACD90] for demonstrating the effectivity of the model checking procedure over Timed Graphs. One may see also [ABBCK91] for a survey, how similar ideas of variable value space partitioning have worked in deciding reachability for various classes of data-dependent programs.

Definition 3.2 *Let* $P_i = \langle \Phi_A, \langle v^{P_i}, \delta^{P_i} \rangle \rangle \in \tilde{d}(A)$, $Q_i = \langle \Phi_B, \langle v^{Q_i}, \delta^{Q_i} \rangle \rangle \in \tilde{d}(B)$ *for* $i = 1, 2$, *we say that* $\langle P_1, Q_1 \rangle \cong \langle P_2, Q_2 \rangle$ *iff*

- $v^{P_1} = v^{P_2}$ *and* $v^{Q_1} = v^{Q_2}$ *(i.e. the vertexes of corresponding processes coincide) and*

– $\delta^{P_1} :: \delta^{Q_1} \cong \delta^{P_2} :: \delta^{Q_2}$, *where :: denotes the concatenation of two vectors (i.e. the "concatenated" vectors belong to the same "time region").*

It is important to notice that for $\langle P_1, Q_1\rangle \cong \langle P_2, Q_2\rangle$ it is not enough that $v^{P_1} = v^{P_2}$, $v^{Q_1} = v^{Q_2}$ and $\delta^{P_1} \cong \delta^{P_2}$, $\delta^{Q_1} \cong \delta^{Q_2}$, also the timer values in P_1 have to be ordered w.r.t. the timer values in Q_1 the same way as the timer values in P_2 are ordered w.r.t. those in Q_2. In order to show the relations between $\cong$ and the defined transition relations $\xrightarrow{\sigma}, \overset{\alpha}{\Longrightarrow}, \xrightarrow{\epsilon(d)}, \overset{\epsilon(d)}{\Longrightarrow}$ between the processes (see Proposition 3.6 and Lemma 3.7) we use an "invariant relation" technique, characterizing first every $\cong$-equivalence class via the following notion of a uniform mapping:

Definition 3.3 *We call a strongly monotone mapping $\rho : Q^{+0} \to Q^{+0}$ uniform if $\rho(0) = 0$ and $\rho(x) + c = \rho(x + c)$ for every natural c.*

We extend any mapping $\rho : Q^{+0} \to Q^{+0}$ in a polymorphic manner to any structures containing nonnegative rationals as elements in a way by applying ρ to every component $a \in Q^{+0}$ and not changing any component of other type, e.g.

$$\rho(\delta_1, \delta_2, \ldots, \delta_m) = (\rho(\delta_1), \rho(\delta_2), \ldots, \rho(\delta_m)),$$

as well as for $P \in \tilde{d}(A)$, $Q \in \tilde{d}(B)$ $\rho\langle P, Q\rangle = \langle P', Q'\rangle$, where P' and Q' have the same vertices as P and Q respectively, but the corresponding timer value vector $\delta^{P'} :: \delta^{Q'} = \rho(\delta^{P} :: \delta^{Q})$, etc. The proofs of the following two facts easily follow from definitions:

Fact 3.4 *$\langle P_1, Q_1\rangle \cong \langle P_2, Q_2\rangle$ if and only if there exists a uniform mapping ρ, such that $\langle P_1, Q_1\rangle = \rho(\langle P_2, Q_2\rangle)$.*

Fact 3.5 *Whenever $\rho : Q^{+0} \to Q^{+0}$ is a uniform mapping, then for every $d \in Q^{+0}$ the mapping ρ_d, defined $\rho_d(x) = \rho(x + d) - \rho(d)$ for every x, is also uniform.*

Proposition 3.6 *Whenever $P_2 = \rho(P_1)$ for the processes $P_1, P_2 \in \tilde{d}(A) \cup \tilde{d}(B)$ and some uniform mapping ρ, we have,*

$$\textit{if } P_1 \overset{\alpha}{\Longrightarrow} P_1', \textit{ then } P_2 \overset{\alpha}{\Longrightarrow} \rho(P_1'), \quad \textit{and,} \quad \textit{if } P_1 \overset{\epsilon(d)}{\Longrightarrow} P_1', \textit{ then } P_2 \overset{\epsilon(\rho(d))}{\Longrightarrow} \rho_d(P_1').$$

Proof: Consider first the untimed transitions. Since P_1 and P_2 have the same vertex, and the same timers with 0 values and, so, the transitions along the same edges enabled, the result follows by observing that for every possible newly appearing timer value $c \in N$ $\rho(c) = c$, use induction along the $(\xrightarrow{\sigma})^*$ derivation for the transition $\overset{\alpha}{\Longrightarrow}$.

As to the timed transitions, consider first the case $P_1 \xrightarrow{\epsilon(d)} P_1'$. Let P_1 have a state $\langle v, \delta\rangle$, then the state of P_2 is $\langle v, \rho(\delta)\rangle$. By the definition of $\xrightarrow{\epsilon(d)}$ for every red edge e, outgoing from v, there exists $t_i \in \gamma(e)$, such that $\delta(t_i) \geq d$. By the monotonicity of ρ for every such t_i $\rho(\delta(t_i)) \geq \rho(d)$, so $P_2 \xrightarrow{\epsilon(\rho(d))} P_2'$ for some P_2'.

In order to prove that $P_2' = \rho_d(P_1')$ it remains to consult the definitions of the transition relation $\xrightarrow{\epsilon(d)}$ and the mapping ρ_d (consider 2 cases whether $\delta(t_i) \leq d$ or $\delta(t_i) > d$).

As to the general case of $P_1 \overset{\epsilon(d)}{\Longrightarrow} P_1'$, it remains to notice that

$$(\rho_d)_{d'}(x) = \rho_d(x+d') - \rho_d(d') = \rho(x+d+d') - \rho(d) - (\rho(d+d') - \rho(d)) = \rho_{d+d'}(x)$$

and to use the induction along the elementary transition chain in the derivation $\overset{\epsilon(d)}{\Longrightarrow}$.□

Lemma 3.7 *Let $P_1, P_2 \in \tilde{d}(A)$, $Q_1, Q_2 \in \tilde{d}(B)$, such that $\langle P_1, Q_1 \rangle \cong \langle P_2, Q_2 \rangle$. Then $P_1 \approx Q_1$ if and only if $P_2 \approx Q_2$.*

Proof: Define the relation $\approx' \subseteq \tilde{d}(A) \times \tilde{d}(B)$ in a way $P \approx' Q$ iff $P_1 \approx Q_1$ for some $\langle P_1, Q_1 \rangle \cong \langle P, Q \rangle$. We obtain the proof by showing that $\approx'$ is a weak bisimulation.

Take some $P \approx' Q$, let $P_1 \approx Q_1$ and $\langle P_1, Q_1 \rangle \cong \langle P, Q \rangle$, then $\langle P_1, Q_1 \rangle = \rho(\langle P, Q \rangle)$ for some uniform ρ (Fact 3.4). By Proposition 3.6 whenever $P \overset{\alpha}{\Longrightarrow} P'$ then $P_1 \overset{\alpha}{\Longrightarrow} \rho(P')$. Since $P_1 \approx Q_1$, then also $Q_1 \overset{\alpha}{\Longrightarrow} Q_1'$ for some Q_1' with $\rho(P') \approx Q_1'$. Since the inverse of a uniform mapping is also uniform, Proposition 3.6 gives $Q \overset{\alpha}{\Longrightarrow} \rho^{-1}(Q_1')$, easy to see that $\langle P', \rho^{-1}(Q_1') \rangle \cong \langle \rho(P'), Q_1' \rangle$ and, so, $P' \approx' \rho^{-1}(Q_1')$, as requested.

All the other cases (including the timed ones) are very similar to the considered one, their detailed analysis is omitted.□

Consider a partitioning $\mathcal{X}_{A,B}$ of the set $\tilde{d}(A) \times \tilde{d}(B)$, generated by $\cong$, easy to see that it is finite (for every $P \in \tilde{d}(A) \cup \tilde{d}(B)$ any its timer value does not exceed $max\{c^A, c^B\}$). For arbitrary $P \in \tilde{d}(A)$, $Q \in \tilde{d}(B)$ let us denote by $X(P, Q)$ the element in this partitioning to which the pair $\langle P, Q \rangle$ belongs to and call it a *region process*, corresponding to $\langle P, Q \rangle$.

3.1 Deciding Strong Equivalence

We consider first the decidability of the strong (i.e. non-abstracted) bisimulation equivalence. We begin with some results, characterizing the "waiting behaviour" of the processes.

Let for $P \in \mathcal{P}$ $P(d)$ be the process which is obtained from the process P by letting time to pass for d units ($P \xrightarrow{\epsilon(d)} P(d)$) provided P can perform such a waiting. We let $\mu(P)$ for $P \in \mathcal{P}$ to denote the minimal nonzero P timer value fractional part (if all timer values in P are integers, let $\mu(P) = 1$), let $\mu(P, Q) = min\{\mu(P), \mu(Q)\}$. We call a process P is *stable*, written $P \xrightarrow{WT}$, if and only if there exists $d > 0$, such that $P \xrightarrow{\epsilon(d)} P(d)$.

Fact 3.8 *For $P \in \mathcal{P}$, if $P \xrightarrow{WT}$, then for all $d \leq \mu(P)$ $P \xrightarrow{\epsilon(d)} P(d)$.*
For $P \in \tilde{d}(A)$ and $Q \in \tilde{d}(B)$, if $P \xrightarrow{WT}$ and $Q \xrightarrow{WT}$, then for all $d, d' \in]0, \mu(P, Q)[$ always $P(d), Q(d), P(d'), Q(d')$ exist and $X(P(d), Q(d)) = X(P(d'), Q(d'))$.

Definition 3.9 *Let for $X = X(P,Q) \in \mathcal{X}_{A,B}$ such that $P \xrightarrow{WT}$ and $Q \xrightarrow{WT}$, $next_0(X) = X(P(\mu/2), Q(\mu/2))$ and $next_1(X) = X(P(\mu), Q(\mu))$, where $\mu = \mu(P,Q)$.*

Clearly, the operations $next_i$ for region processes are well-defined and effective.

Let for $X, X' \in \mathcal{X}_{A,B}$ $X \xrightarrow{\sigma} X'$ iff there exist $\langle P,Q \rangle \in X$ and $\langle P',Q' \rangle \in X'$ such that $P \xrightarrow{\sigma} P'$ and $Q \xrightarrow{\sigma} Q'$.

Definition 3.10 *The set $\mathcal{X} \in \mathcal{X}_{A,B}$ is a strong symbolic bisimulation if and only if for all $X = X(P,Q) \in \mathcal{X}$*

- *whenever $P \xrightarrow{\sigma} P'$ then $X \xrightarrow{\sigma} X(P',Q') \in \mathcal{X}$ for some Q';*
- *whenever $Q \xrightarrow{\sigma} Q'$ then $X \xrightarrow{\sigma} X(P',Q') \in \mathcal{X}$ for some P';*
- *whenever $P \xrightarrow{WT}$, or $Q \xrightarrow{WT}$, then both $next_0(X) \in \mathcal{X}$ and $next_1(X) \in \mathcal{X}$.*

Due to the finiteness of $\mathcal{X}_{A,B}$ and according to Lemma 3.7 the following two results complete the proof of Theorem 2.2 (see [Cer92c] for the proof details omitted here).

Lemma 3.11 *The set $R_{\mathcal{X}} = \{\langle P,Q \rangle \mid X(P,Q) \in \mathcal{X}\}$ is a strong timed bisimulation if and only if the set $\mathcal{X}$ is a strong symbolic bisimulation.*

Proof: See Lemma 3.15.□

Lemma 3.12 *It is decidable whether given set $\mathcal{X} \subseteq \mathcal{X}_{A,B}$ is a strong symbolic bisimulation.*

Proof: Follows from the definitions of $\xrightarrow{\sigma}$, $next_0$ and $next_1$. Proposition 3.6 and Fact 3.8 guarantee the independence on the choice of the representants.□

3.2 Deciding Weak Equivalence

In the general case of the deciding weak bisimulation we follow the same lines, as in the case of the strong bisimulation. Let for $P \in \tilde{d}(A)$, $Q \in \tilde{d}(B)$ and $\mu = \mu(P,Q)$:
whenever $P \xrightarrow{WT}$ then $\mathcal{N}_0^A(P,Q) = \{X(P(d), Q_d) \mid Q \overset{\epsilon(d)}{\Longrightarrow} Q_d \ \&\ 0 < d < \mu\}$ and $\mathcal{N}_1^A(P,Q) = \{X(P(\mu), Q') \mid Q \overset{\epsilon(\mu)}{\Longrightarrow} Q'\}$ (the sets $\mathcal{N}_0^B(P,Q)$ and $\mathcal{N}_1^B(P,Q)$ are defined in a similar way).

Fact 3.13 *For every $X \in \mathcal{N}_0^A(P,Q)$ and for every $d \in]0, \mu(P,Q)[$ there exists Q_d with both $Q \overset{\epsilon(d)}{\Longrightarrow} Q_d$ and $\langle P(d), Q_d \rangle \in X$.*

We introduce for $\nu \in Vis \cup \{\epsilon\}$ in the set $\mathcal{X}_{A,B}$ of region processes the relations $X \overset{\nu}{\Longrightarrow} X'$ iff there exist $\langle P,Q \rangle \in X$ and $\langle P',Q' \rangle \in X'$ such that $P \overset{\nu}{\Longrightarrow} P'$ and $Q \overset{\nu}{\Longrightarrow} Q'$.

Definition 3.14 *Let for any $\mathcal{X} \subseteq \mathcal{X}_{A,B}$ $F^*(\mathcal{X})$ be the set of all $X(P,Q)$ satisfying*

- *if $P \overset{\nu}{\Longrightarrow} P'$, then $X(P,Q) \overset{\nu}{\Longrightarrow} X(P',Q') \in \mathcal{X}$ for some Q';*
- *if $Q \overset{\nu}{\Longrightarrow} Q'$, then $X(P,Q) \overset{\nu}{\Longrightarrow} X(P',Q') \in \mathcal{X}$ for some P';*
- *if $P \xrightarrow{WT}$ then both $X' \in \mathcal{N}_0^A(P,Q) \cap \mathcal{X}$ and $X'' \in \mathcal{N}_1^A(P,Q) \cap \mathcal{X}$ for some X', X''; and*
- *if $Q \xrightarrow{WT}$ then both $X' \in \mathcal{N}_0^B(P,Q) \cap \mathcal{X}$ and $X'' \in \mathcal{N}_1^B(P,Q) \cap \mathcal{X}$ for some X', X''.*

Then $\mathcal{X}$ is a weak symbolic bisimulation, if $\mathcal{X} \subseteq F^*(\mathcal{X})$.

The proof of Theorem 2.4 is obtained by showing the following two lemmas:

Lemma 3.15 *For $\mathcal{X} \in \mathcal{X}_{A,B}$ and $R_{\mathcal{X}} = \{\langle P,Q\rangle \mid X(P,Q) \in \mathcal{X}_{A,B}\}$ $\mathcal{X}$ is a weak symbolic bisimulation if and only if $R_{\mathcal{X}}$ is a weak (timed) bisimulation.*

Lemma 3.16 *It is decidable, whether a given set $\mathcal{X} \subseteq \mathcal{X}_{A,B}$ is a weak symbolic bisimulation.*

Proof of Lemma 3.15 (outline): Let $\mathcal{X}$ be a weak symbolic bisimulation. In order to prove that $R_{\mathcal{X}}$ is a weak (timed) bisimulation, take $\langle P,Q\rangle \in R_{\mathcal{X}}$. All α- and ϵ- moves of P can be matched by corresponding moves of Q (and vice versa) due to Proposition 3.6.

Consider the timed cases. Given that for every $X(P,Q) \in \mathcal{X}$ whenever $P \xrightarrow{WT}$ then also $\mathcal{N}_0^A \cap \mathcal{X}$ and $\mathcal{N}_1^A \cap \mathcal{X}$ are nonempty, we prove first that for every $\langle P,Q\rangle \in R_{\mathcal{X}}$ and every $d > 0$, if $P \xrightarrow{\epsilon(d)} P(d)$, then also $Q \overset{\epsilon(d)}{\Longrightarrow} Q_d$ for some Q_d with $\langle P(d), Q_d\rangle \in R_{\mathcal{X}}$. For this purpose we, given $\langle P,Q\rangle \in R_{\mathcal{X}}$, consider a sequence of process pairs $\langle P_i, Q_i\rangle$ such that

- $P_0 = P$ and $Q_0 = Q$,
- $P_{i+1} = P_i(\mu_i)$ and $Q_i \overset{\epsilon(\mu_i)}{\Longrightarrow} Q_{i+1}$ (we abbreviate $\mu_i = \mu(P_i,Q_i)$), and
- $\langle P_i, Q_i\rangle \in R_{\mathcal{X}}$ (we can require this due to the definition of $\mathcal{N}_1^A$).

One can show that $\mu_0 + \mu_1 + \ldots + \mu_k \geq d$ for some k, the result follows.

In the case of $P \overset{\epsilon(d)}{\Longrightarrow} P'$, the matching Q' with $\langle P',Q'\rangle \in R_{\mathcal{X}}$ is found inductively along the derivation of P' from P.

The proof that for every weak timed bisimulation $R_{\mathcal{X}}$ the corresponding set $\mathcal{X}$ is a weak symbolic bisimulation can be obtained from the definitions and Fact 3.8.□

Proof of Lemma 3.16: Since for every pair of timed processes $\langle P,Q\rangle \in \tilde{d}(A) \times \tilde{d}(B)$ the set of process pairs $\langle P',Q'\rangle$ with $P \overset{\nu}{\Longrightarrow} P'$ and $Q \overset{\nu}{\Longrightarrow} Q'$ for any $\nu \in Vis \cup \{\epsilon\}$ is finite and effectively computable from $\langle P,Q\rangle$ (the set of all $\overset{\nu}{\Longrightarrow}$ derivations can be infinite due to the repeating τ-loops; all newly appearing timer values in the processes in these derivations are integers from a bounded interval), Proposition 3.6 guarantees the decidability of untimed match existence for all processes $X(P,Q)$ with either $P \overset{\nu}{\Longrightarrow}$, or $Q \overset{\nu}{\Longrightarrow}$.

Let us demonstrate the effectivity of the check, whether for given $X \in \mathcal{X}$ with $P \xrightarrow{WT}$ for some $\langle P,Q\rangle \in X$ both the sets $\mathcal{N}_0^A(P,Q) \cap \mathcal{X}$ and $\mathcal{N}_1^A(P,Q) \cap \mathcal{X}$ are not empty. For this purpose we show the algorithms, *generating* the sets $\mathcal{N}_0^A(P,Q)$ and $\mathcal{N}_1^A(P,Q)$ from the given processes $P \in \tilde{d}(A)$ and $Q \in \tilde{d}(B)$. For the sake of simplicity assume that there exist a timer in P with the value fractional part being $\mu(P,Q)$ (the general case is dealt with in [Cer92c] by a slight refinement of the region processes).

We define for region processes $X \in \mathcal{X}$ the transitions $X \xrightarrow{*} next_0(X)$, $X \xrightarrow{**} next_1(X)$ and $X(P',Q') \xrightarrow{\tau} X(P',Q'')$ whenever $Q' \xrightarrow{\tau} Q''$.

Let $R^0_{P,Q}$ be the set of region processes which are reachable from $X(P,Q)$ using the transitions $\xrightarrow{*}$ and $\xrightarrow{\tau}$. Let $R_{P,Q} \subseteq R^0_{P,Q}$ be the set of processes with a derivation containing at least one $\xrightarrow{*}$ transition. Let $R'_{P,Q}$ be the set of region processes, reachable from those in $R^0_{P,Q}$ by *one* $\xrightarrow{**}$ transition, followed by a number of $\xrightarrow{\tau}$ transitions.

Observing the effectivity of the set $R_{P,Q}$ and $R'_{P,Q}$ computation, the following result completes the proof of Lemma 3.16, so completing also the proof of Theorem 2.4:

Proposition 3.17 $\mathcal{N}_A^0(P,Q) = R_{P,Q}$ *and* $\mathcal{N}_A^1(P,Q) = R'_{P,Q}$.

Proof: Rather technical, see [Cer92c].□□

Note: The weak bisimulation deciding algorithm can be simplified in the case, if both the processes A and B satisfy the *maximal progress* assumption, stating that τ-labels in the processes can be ascribed only to red (instantaneous) edges. For instance, we could have $\mathcal{N}_A^0(P,Q) = \{X(P(d),Q_d) \mid Q \overset{\epsilon}{\Longrightarrow} Q' \xrightarrow{\epsilon(d)} Q_d \ \&\ 0 < d < \mu\}$, what clearly simplifies the bisimulation deciding procedure in Lemma 3.16.

4 Timed Processes with Memory

We consider an enrichment of PTPs with memory for moving timer value information along the time axis (storing timer value at one moment and retrieving it afterwards).

A parallel timer process with memory (PTPM) is obtained by adding to a given PTP $A = \langle\langle V,E,L,lab,\mathcal{T},\gamma,\phi\rangle,\langle v,\delta\rangle\rangle$ a finite set $\mathcal{M}$ of *memory cells* and extending every edge timer setting $\phi(e)$ for $e \in E$ with some "remember" operations $m_i \leftarrow t_j$ and some "retrieve" operations $t_j \leftarrow m_i$ for $t_j \in \mathcal{T}, m_i \in \mathcal{M}$ (formally, $\phi(e) : (\mathcal{T} \cup \mathcal{M}) \to (\mathcal{T} \cup \mathcal{M}) \cup Q^{+0}$).

The semantics (labelled transition system) of PTPMs is easily obtained as a generalization of that of PTPs: every state for a PTPM A consists of its graph's vertex and a value assignment both for timers and memory cells. Every time a transition along an edge e fires, the timer and memory cell setting $\phi(e)$ defines the new values $\delta'(u)$ for $u \in \mathcal{T} \cup \mathcal{M}$ from the old ones, $\delta(u)$, as for PTPs: $\delta'(u) = \delta(\phi(e)(u))$ (here $\delta(c) \overset{def}{=} c$ for $c \in Q^{+0}$).

The main difference of the memory cells from the timers is that the values of the memory cells *do not decrease during the passage of time* as the timer values do (whenever $\langle \Phi, \langle v, \delta \rangle \rangle \xrightarrow{\epsilon(d)} \langle \Phi, \langle v, \delta' \rangle \rangle$, then $\delta'(m) = \delta(m)$ for every $m \in \mathcal{M}$).

Let for any PTPM A $d(A)$ denote the set of *derivatives* of A (the least set containing A and closed under the transition relation). We call a vertex v in a given PTPM $A = \langle \Phi, s_0 \rangle$ *reachable* if $\langle \Phi, \langle v, \delta \rangle \rangle \in d(A)$ for some timer and memory cell value assignment δ.

Theorem 4.1 *The vertex reachability problem for parallel timer processes with* 5 *timers and* 1 *memory cell is undecidable.*

Corollary 4.2 *The bisimulation equivalence problem for parallel timer processes with* 5 *timers and* 1 *memory cell is undecidable.*

For a proof of Theorem 4.1 the reader may consult [Cer92b]. We just point out here that it is based on the modelling of two 2-way counter machine. One timer in the modelling is used to generate regular "ticks", the values of counters are modeled as ratious $\delta(t_2)/\delta(t_1)$ and $\delta(t_3)/\delta(t_1)$ of timer t_1, t_2 and t_3 values at the appropriate "tick" moments. One can easily implement in a PTPM both the timer addition/subtraction and timer halving, all what is needed to model the counter machine's instructions simultaneously not letting the timers to grow bigger than the largest constant used in the timer settings.

It can be pointed out that the undecidability results of Theorem 4.1 and Corollary 4.2 still retain in force also, if one forbids the assignment operations between the timers, as well as replaces the assignments between timers and memory cells by simply holding and releasing operations over the timers (see [Cer92b] or [Cer92c] for some details).

Both the decidability results, obtained in the previous sections of the paper (see Theorem 2.2 and Theorem 2.4), and the undecidability results, expressed by Theorem 4.1 and Corollary 4.2 have noteworthy implications regarding the possibilities to decide the bisimulation equivalences for various classes of Timed CCS processes (see [Wan90, Wan91]). First, we can model a class of TCCS processes, reasonably called "TCCS-nets", into the PTP formalism, so showing the decidability of the bisimulation equivalence for these net processes (see [Cer91] and [Cer92c] for two different possible modelling strategies). On the other hand, if one considers the version of Timed CCS with the expansion theorem, as presented in [Wan91], it becomes easy to encode every PTPM as a term in these generalised calculi (in fact, rather simple subcalculi of interleaving TCCS are sufficient to have this encoding possibility), so showing that all nontrivial interesting algorithmic problems for this kind of processes are undecidable. It can be an interesting question, whether one can come up with another timed specification formalism which would combine both the interleaving nature of the specifications and the decidability of the bisimulation equivalence problems.

5 Conclusions

This paper does not contain any discussion on the *compositional* properties of Parallel Timer processes because its main aim is to discuss the *decidability* issues. However, the definition of parallel composition (and other static process algebra combinators) for PTPs can be done in a quite straightforward way (in [Cer92c] one can find the rules for combining PTPs in parallel both according to CCS and CSP kinds of inter-process communication).

Despite some design differences of PTPs from Action Timed Graphs [NSY91], the presented algorithms for the bisimulation deciding for PTPs with slight modifications can be applied also for ATGs with "linear" predicates over the clock values, see [Cer92c] for details.

The complexity issues of the obtained bisimulation deciding algorithms are not explicitely discussed here, however, as these algorithms are in fact presented as computing some simple symbolic greatest fixpoint relations, the techniques from [Lar92] are hopefully to be applicable to obtain efficient computations of these relations.

It is possible to consider also various more or less principal enrichments of the PTP model which, unlike the PTPMs, have decidable at least the vertex reachability problem, and in some cases also the bisimulation equivalence problem. One such class is so-called PTPs with Nondeterministic timer settings (PTPNs) which are obtained from PTPs be redefining the timer setting functions in a way $\phi(e) : \mathcal{T} \rightarrow \mathcal{T} \cup I(Q^{+0})$, where $I(Q^{+0})$ denotes the set of all intervals over Q^{+0} and defining the early choice semantics of the nondeterminism in the timer value settings. One can show that for PTPNs the vertex reachability problem is decidable; the decidability of the strong and weak bisimulation equivalence problems for PTPNs can be shown provided all the intervals in process edge timer setting functions are *finite*, see [Cer92c] for details. In [Cer92c] also some enrichments of PTPs with external data (both integer- and rational-valued, simple tests on being less or greater for the values of data variables are allowed) are considered, the decidability of the bisimulation equivalences for PTPs with dependencies on rational-valued data is showed.

6 Acknowledgements

I want to thank K.V.S Prasad, Alan Jeffrey, Wang Yi and Uno Holmer in Göteborg as well as Kim G. Larsen and Jens Chr. Godskesen in Aalborg for a lot of very interesting and constructive discussions on process algebras, real time and decidability.

My thanks are also to the anonymous referees of this paper for providing a number of useful comments on an earlier version of it.

The final version of this paper was processed using LaTeX macro package with LLNCS style and printed at Chalmers University of Technology, Göteborg, Sweden.

References

[AD90] R. Alur and D.Dill, *Automata for Modelling Real-Time Systems*, LNCS No. 443, 1990.

[ACD90] R. Alur, C.Courcoubetis and D.Dill, *Model-Checking for Real-Time Systems*, Proceedings from LICS'90 pp. 414-425, 1990.

[ABBCK91] A. Auziņš, J. Bārzdiņš, J. Bičevskis, K. Čerāns and A. Kalniņš, *Automatic Construction of Test Sets: Theoretical Approach*, Baltic Computer Science, LNCS, No. 502, 1991.

[BM83] B. Berthomieu and M.Menasche, *An Enumerative Approach for Analyzing Time Petri Nets*, Proc. IFIP Congress, 1983, North-Holland, 1983.

[CC88] **CCITT Specification and Description Language (SDL)**, Recomendations Z.100, 1988.

[Cer91] K. Čerāns, *Decidability of Bisimulation Equivalence for Parallel Timed Processes*, in Proc. of Chalmers Workshop on Concurrency, Göteborg, Report PMG-R63, Chalmers University of Technology, 1992.

[Cer92a] K. Čerāns, *Feasibility of Finite and Infinite Paths in Data Dependent Programs*, in Proc. for LFCS'92, Russia, Tver, LNCS No. 620, 1992.

[Cer92b] K. Čerāns, **Decidability of Bisimulation Equivalences for Processes with Parallel Timers**, Technical report, Institute of Mathematics and Computer Science, University of Latvia, Riga, 1992.

[Cer92c] K. Čerāns, **Algorithmic Problems in Analysis of Real Time System Specifications**, Dr.Sc.comp theses, University of Latvia, Riga, 1992.

[Che91] L. Chen, *Decidability and Completeness in Real Time Processes*, LFCS, Edinburgh University, 1991.

[GMMP89] C.Ghezzi, D.Mandrioli, S.Morasca and M.Pezze *A General Way To Put Time in Petri Nets*, ACM SIGSOFT Eng. Notes, Vol. 14, No. 3, 1989.

[HLW91] U. Holmer, K.Larsen and Yi Wang, *Deciding Properties for Regular Real Timed Processes*, CAV'91, 1991.

[Lar92] K.G.Larsen, *Efficient Local Correctness Checking*, this Workshop, 1992.

[MF76] P. Merlin and D.J. Farber, *Recoverability of Communication Protocols*, IEEE Trans. on Communication Protocols, Vol. COM-24, No. 9, 1976.

[NSY91] X.Nicollin, J.Sifakis and S.Yovine, *From ATP to Timed Graphs and Hybrid Systems*, in Proc. of REX Workshop "Real-Time: Theory in Practice", 1991.

[RR86] G.M. Reed and A.W. Roscoe, *A Timed Model for Communicating Sequential Processes*, LNCS No. 226, 1986.

[Wan90] Yi Wang, *Real Time Behaviour of Asynchronous Agents*, LNCS No. 458, 1990.

[Wan91] Yi Wang, *CCS + Time = an Interleaving Model for Real Time Systems*, ICALP'91, Madrid, 1991.