Deciding Properties of Regular Real Timed Processes

Uno Holmer * Kim Larsen † Wang Yi *

Abstract

We discuss the *decidability* problem associated with verifying properties of processes expressed in the real time process calculus TCCS of [W90]. A regular subcalculus TC of TCCS is considered. Two operational semantics, and associated timed notions of bisimulation, are given: a standard infinite semantics, and a symbolic finite semantics. The consistency between the two semantics is proved. We show that both the equivalences are decidable for regular processes relative to comparisons between real numbers.

As an alternative specification formalism, we present a timed modal logic. It turns out that this logic characterises timed bisimulation equivalence in the sense that equivalent processes enjoy exactly the same properties expressed within the logic. Moreover, we prove that the problem of deciding whether a given regular real timed process satisfies a given property of the logic is decidable, relative to first order assertions about real numbers. Two interpretations of the modal logic are offered, based on the standard and symbolic operational semantics of TC respectively and the consistency between these interpretations is proved.

1 Motivation

Recently, numerous models within the frameworks for timed processes based on process calculi and temporal logic have been developed [ACD90, AD90, HR90, MT90, RR86, S90, W90]. In this paper, we discuss the *decidability* problem associated with verifying properties of processes expressed in the real time process calculus TCCS of Wang [W90].

As the specification language for expressing such properties one may choose to use the timed calculus itself, with the notion of correctness given in terms of some time–sensitive abstracting equivalence (a timed version of bisimulation equivalence, say). In a discrete timed model such as [S90, HR90], only one unique time event is introduced into the untimed model of CCS [M89] to deal with timing information, which represents a clock tick. In these models the addition of time does not contribute to the *infiniteness* of the labelled transition system in terms of which the the operational semantics of processes is given. As a consequence one may readily apply the existing standard decision algorithm for bisimulation equivalences [CPS89, PT87] to decide the correctness of an implementation P with respect to a specification S (i.e. whether they are equivalent). However, in a dense timed model such as [W90], it is not obvious that bisimulation equivalence is decidable: for each time instant, a process will have a corresponding state; consequently the state–space of any process will certainly be infinite (in fact there will be a continuum of states). Let us specify a coffee machine in timed CCS [W90].

$$S_0 = coin.S_1$$
$$S_1 = coffee.S_0 + \epsilon(30).\tau.S_0$$

Note that we have used the delay construct $\epsilon(d).P$ of timed CCS [W90], which means "waits for d seconds and then behaves like P". The τ models a time-out event.

Informally, the machine waits for the user to insert a coin after which it is willing to offer a cup of coffee. If the user takes the drink within 30 seconds, it returns to the initial state. After 30 seconds, the

*Address: Department of Computer Sciences, Chalmers University of Technology, S–412 96 Göteborg, Sweden. E–mail: holmer@cs.chalmers.se, yi@cs.chalmers.se

†Address: Department of Mathematics and Computer Science, Aalborg University, Fredrik Bajersvej 7, 9220 Aalborg, Denmark. E–mail: kgl@iesd.auc.dk

machine will time-out and return autonomously to its initial state to collect another coin. According to the operational semantics of [W90], we have a continuum of time-transitions for S_1,

$$S_1 \xrightarrow{\epsilon(d)} coffee.S_0 + \epsilon(30-d).\tau.S_0$$

for all $d \in]0,30]$. For instance,

$$S_1 \xrightarrow{\epsilon(0.5)} coffee.S_0 + \epsilon(29.5).\tau.S_0$$

That is, the machine has a dense and infinite state space. Hence, given a process M we cannot directly use the existing algorithmic techniques such as [CPS89, PT87] to decide whether $M \sim S_0$.

In this paper, we shall consider the *regular real timed processes* of [W90] which are the regular part of CCS plus a delay construct $\epsilon(d).P$ where $d \in \mathcal{R}^+$. We formally define a notion of timed bisimulation equivalence, and show that this equivalence is decidable for regular processes relative to comparisons between real numbers.

As an alternative (logical) specification formalism, we present a timed modal logic. This logic allows one to specify properties such as: "After a coin has been inserted, coffee will be continuously available for 30 seconds". It turns out that this logic characterises (timed) bisimulation equivalence in the sense that equivalent processes enjoy exactly the same properties expressed within the logic. Moreover, we prove that the problem of deciding whether a given regular real time process satisfies a given property of the logic is decidable (relative to first order assertions about real numbers).

The outline of the paper is as follows: In section 2 we introduce the calculus TC of regular real time processes. Two operational semantics (and associated timed notions of bisimulation) are given: the standard (infinite) semantics of [W90] and a (finite) symbolic semantics. The consistency between the two semantics provides the key to our decidability results. In section 3 we introduce the timed modal logic TML. Again two interpretations are offered, based on the standard and symbolic operational semantics of TC respectively. The consistency between these interpretations again leads to our decidability results. In section 4 we state our conclusions and directions for future work.

2 A Timed Calculus

In this section we present a subcalculus TC of [W90]. Assume a set of action names Δ ranged over by $a, b, \ldots$ and a special action $\tau \notin \Delta$. Let $\mathcal{A}ct = \Delta \cup \{\tau\}$ and $\mu, \nu, \ldots$ range over $\mathcal{A}ct$. Let $\mathcal{A} = \mathcal{A}ct \cup \{\epsilon(c) \mid c \in \mathcal{R}^+\}$ where $\mathcal{R}^+$ are the positive real numbers, and let σ range over $\mathcal{A}$. Assume a set of process variables $\mathcal{V}$ ranged over by $x, y, \ldots$. The regular process expressions are given by the following abstract syntax:

$$E ::= NIL \mid x \mid \sigma.E \mid E + E \mid rec\, x : E$$

Closed process expressions will be denoted by the letters $P, Q, \ldots$ We will restrict all processes to be *action guarded* in the following sense:

Definition 2.1 *x is action guarded in E iff every free occurrence of x in E is within a subexpression (a guard) of the form $\mu.F$ in E. E is action guarded iff every free variable in E is action guarded in E, and for every subexpression of the form $rec\, x : F$ in E, x is action guarded in F.* □

We denote by $\mathcal{P}r$ the set of all closed and action guarded process expressions.

Example 2.2 *$a.NIL$ and $\tau.(rec\, y : x + b.y)$ are action guarded whereas $rec\, x : x + b.x$ and $rec\, x : \epsilon(c).x + a.x$ are not. The latter is not action guarded because $\epsilon(c).x$ is not a guard.* □

2.1 Standard Operational Semantics

The standard semantics is given by the transition system $\langle \mathcal{P}r, \mathcal{A}, \longrightarrow \rangle$ where $\longrightarrow$ is the least relation generated by the rules in table 1.

Inaction	$\dfrac{}{NIL \xrightarrow{\epsilon(c)} NIL}$
Prefix	$\dfrac{}{\mu.P \xrightarrow{\mu} P}$ $\quad$ $\dfrac{}{a.P \xrightarrow{\epsilon(c)} a.P}$ $\dfrac{}{\epsilon(c).P \xrightarrow{\epsilon(c)} P}$ $\quad$ $\dfrac{P \xrightarrow{\epsilon(c)} P'}{\epsilon(d).P \xrightarrow{\epsilon(c+d)} P'}$ $\quad$ $\dfrac{}{\epsilon(c+d).P \xrightarrow{\epsilon(c)} \epsilon(d).P}$
Summation	$\dfrac{P \xrightarrow{\mu} P'}{P+Q \xrightarrow{\mu} P'}$ $\quad$ $\dfrac{Q \xrightarrow{\mu} Q'}{P+Q \xrightarrow{\mu} Q'}$ $\quad$ $\dfrac{P \xrightarrow{\epsilon(c)} P' \qquad Q \xrightarrow{\epsilon(c)} Q'}{P+Q \xrightarrow{\epsilon(c)} P'+Q'}$
Recursion	$\dfrac{E\{rec\, x : E/x\} \xrightarrow{\sigma} P}{rec\, x : E \xrightarrow{\sigma} P}$

Table 1: Standard operational semantics for TC.

For most processes P, the state space and/or the set of transitions which can be generated from P, via $\longrightarrow$, is (wildly) infinite. The most immediate example is NIL for which we can derive $NIL \xrightarrow{\epsilon(c)} NIL$ for any positive real number c. Thus the transition relation is infinite although the state space is not in this case.

Another example is $\epsilon(c).NIL$ which has the ϵ-derivatives $\epsilon(c-d).NIL$ for any $d < c$. Here both the state space and the transition relation are infinite.

The most discouraging consequence of this is that we cannot apply existing techniques for deciding *bisimulation equivalence* based on the standard semantics. Another serious defect is that we cannot draw transition diagrams for simple behaviours such as the coffee machine.

The following lemma gives some crucial properties of the semantics:

Lemma 2.3

i)	$P \xrightarrow{\tau} \Rightarrow P \not\xrightarrow{\epsilon(\varsigma)}$	*(maximal progress)*
ii)	$P \xrightarrow{\tau} \vee P \xrightarrow{\epsilon(e)}$	*(transition liveness)*
iii)	$P \xrightarrow{\epsilon(c)} P' \wedge P \xrightarrow{\epsilon(c)} P'' \Rightarrow P' \equiv P''$	*(time determinacy)*
iv)	$\forall c, d > 0. P \xrightarrow{\epsilon(c+d)} P'' \Leftrightarrow \exists P'. P \xrightarrow{\epsilon(c)} P' \wedge P' \xrightarrow{\epsilon(d)} P''$	*(time continuity)*
v)	$P \xrightarrow{\epsilon(d)} P' \wedge P \xrightarrow{a} Q \Rightarrow P' \xrightarrow{a} Q$	*(persistency)*

where $\equiv$ *in iii is syntactical identity and* $P \xrightarrow{\sigma}$ *means* $\exists P'. P \xrightarrow{\sigma} P'$. □

From the standard operational semantics we define *timed bisimulation equivalence* as usual.

Definition 2.4 *Let* $\mathcal{F}(R)$ *be the set of all* (P, Q) *satisfying*

i) Whenever $P \xrightarrow{\sigma} P'$ *then* $Q \xrightarrow{\sigma} Q'$ *with* $(P', Q') \in R$ *for some* Q'
ii) Whenever $Q \xrightarrow{\sigma} Q'$ *then* $P \xrightarrow{\sigma} P'$ *with* $(P', Q') \in R$ *for some* P'

Then R *is a timed bisimulation if* $R \subseteq \mathcal{F}(R)$ *and timed bisimulation equivalence, written* $\sim$, *is defined to be the greatest fixpoint of* $\mathcal{F}$. □

2.2 Symbolic Operational Semantics

In this section we give an alternative operational semantics which turns out to be equivalent to the standard operational semantics—up to bisimulation equivalence. It is called "symbolic" because now every process will only give rise to a finite state space and a finite transition relation and we may draw a graph to represent it. The intuition behind the symbolic interpretation is based on the persistency property. Due to this property the behaviour of a process may be completely inferred from the first time-instant at which a transition is enabled.

First we define the *maximal life-time* of a process to be the (unique) time-instant at which a τ-action is enabled. If no τ-action is ever possible, the maximal life-time is ∞.

Example 2.5 *The process $a.P + \epsilon(3).\tau.Q$ remains stable for at most 3 time units. If the environment has not offered a up to time 3 then the process will autonomously become Q at time 3. The process $a.P + \epsilon(3).b.Q$ on the other hand, remains stable forever if the environment never offers a or b.* □

Definition 2.6 *For a process expression E we define the maximal-life-time function $M(E) : (\mathcal{V} \to [0,\infty]) \to [0,\infty]$ inductively as follows:*

$$\begin{array}{lcllcl} M(NIL)\rho & = & \infty & M(E+F)\rho & = & min(M(E)\rho, M(F)\rho) \\ M(a.E)\rho & = & \infty & M(x)\rho & = & \rho(x) \\ M(\tau.E)\rho & = & 0 & M(rec\, x : E)\rho & = & \mu t.M(E)\rho[x \mapsto t] \\ M(\epsilon(c).E)\rho & = & c + M(E)\rho & & & \end{array}$$

where $\mu t.f(t)$ denotes the least fixpoint of f. For closed process expressions we define $M(P) = M(P)\rho_0$ where ρ_0 is the time environment mapping any process variable to 0. □

The following lemma relates M with the standard semantics.

Lemma 2.7

i) $\forall c \in]0, M(P)] - \{\infty\}.P \xrightarrow{\epsilon(c)}$
ii) $P \xrightarrow{\tau} \Leftrightarrow M(P) = 0$
iii) $P \xrightarrow{\epsilon(c)} P' \Rightarrow M(P') = M(P) - c$
iv) $P \xrightarrow{\epsilon(c)} \Rightarrow M(P) \geq c$
v) $P \xrightarrow{\epsilon(c)} P' \xrightarrow{\tau} \Rightarrow M(P) = c$
vi) $P \sim Q \Rightarrow M(P) = M(Q)$

□

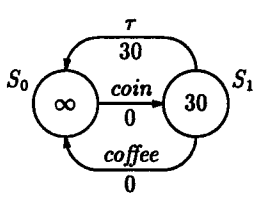
The new semantics is defined by the transition system

$$\langle Pr, Act \times [0,\infty[, \to^* \rangle$$

where $\to^*$ is generated by the rules in table 2. We use the notation $P \xrightarrow{\mu}_c{}^* P'$ for $(P, (\mu, c), P') \in \to^*$, which may be interpreted as "the transition $P \xrightarrow{\mu} P'$ is enabled at time c".

The most important property of the symbolic semantics is that the state space as well as the set of transitions of a process will now be *finite*. We can draw circles to represent states and let numbers inside the circles denote the maximal life-times of the corresponding states. The transitions are drawn in the obvious way.

Example 2.8 *The coffee machine example from the motivation may be expressed in* TC *as $rec\, x : coin.(coffee.x + \epsilon(30).\tau.x)$. The symbolic transition system for the coffee machine can then be drawn as follows:*



where $S_0 = rec\, x : coin.(coffee.x + \epsilon(30).\tau.x)$ and $S_1 = coffee.S_0 + \epsilon(30).\tau.S_0$. □

Prefix	$\dfrac{}{\mu.P \xrightarrow{\mu}{}^*_0 P}$ $\qquad$ $\dfrac{P \xrightarrow{\mu}{}^*_c P'}{\epsilon(d).P \xrightarrow{\mu}{}^*_{c+d} P'}$
Summation	$\dfrac{P \xrightarrow{\mu}{}^*_c P'}{P+Q \xrightarrow{\mu}{}^*_c P'}\ [M(Q) \geq c]$ $\qquad$ $\dfrac{Q \xrightarrow{\mu}{}^*_c Q'}{P+Q \xrightarrow{\mu}{}^*_c Q'}\ [M(P) \geq c]$
Recursion	$\dfrac{E\{rec\, x : E/x\} \xrightarrow{\mu}{}^*_c P}{rec\, x : E \xrightarrow{\mu}{}^*_c P}$

Table 2: Symbolic operational semantics for TC.

Definition 2.9 *Let derivatives(P) be the least set of processes satisfying the following:*

$$P \in derivatives(P)$$
$$Q \in derivatives(P) \wedge Q \xrightarrow{\mu}{}^*_c R \Rightarrow R \in derivatives(P)$$

and let

$$\begin{aligned} labels(P) &= \{(\mu, c) \mid \exists Q, R \in derivatives(P).Q \xrightarrow{\mu}{}^*_c R\} \\ \rightarrow^*(P) &= (derivatives(P) \times labels(P) \times derivatives(P)) \cap \rightarrow^* \end{aligned}$$

□

Proposition 2.10 *For every action guarded process P, derivatives(P) and labels(P) are finite.* □

Thus every $P \in Pr$ generates a finite *local transition system* given by

$$\langle derivatives(P), labels(P), \rightarrow^*(P) \rangle$$

This property of the symbolic semantics is crucial for the decidability results which follow later.

Starting from the symbolic semantics instead of the standard one we define *symbolic timed bisimulation*. A symbolic timed bisimulation is a much coarser relation than an ordinary timed bisimulation in that it only contains the "important" states. As before, when matching two processes against each other, every transition of one of them must be matched by a corresponding transition of the other—and vice versa—and leading to equivalent states. However, the requirement on the matching transition is now relaxed: its enabling time may *precede* the enabling time of the other transition.

Example 2.11 *Consider the following processes:*

$$P = \epsilon(1).a.P_1 + \epsilon(2).a.P_2 \qquad \text{and} \qquad Q = \epsilon(1).a.Q_1 + \epsilon(3).a.Q_2$$

A symbolic bisimulation containing (P, Q) must also contain (P_1, Q_1), (P_2, Q_1) and either (P_1, Q_2) or (P_2, Q_2). □

We define symbolic timed equivalence as follows:

Definition 2.12 *Let $\mathcal{F}^*(R)$ be the set of all (P, Q) satisfying*

i) Whenever $P \xrightarrow{\mu}{}^*_c P'$ then $Q \xrightarrow{\mu}{}^*_d Q'$ with $(P', Q') \in R$ for some Q' and $d \leq c$
ii) Whenever $Q \xrightarrow{\mu}{}^*_c Q'$ then $P \xrightarrow{\mu}{}^*_d P'$ with $(P', Q') \in R$ for some P' and $d \leq c$

Then R is a symbolic timed bisimulation if $R \subseteq \mathcal{F}^*(R)$ and symbolic timed equivalence, written $\sim^*$, is defined to be the greatest fixpoint of $\mathcal{F}^*$. □

We are now ready for our first main theorem:

Theorem 2.13 *Symbolic timed equivalence between action guarded* TC *processes is decidable relative to inequations between positive real numbers.* □

2.3 Relating the two Semantics

The following relationships between the two transition relations will turn out to be useful:

Lemma 2.14

i) $P \xrightarrow{\mu}_0^* Q \Leftrightarrow P \xrightarrow{\mu} Q$

ii) $c > 0 \wedge P \xrightarrow{\mu}_c^* Q \Rightarrow \exists P'.P \xrightarrow{\epsilon(c)} P' \xrightarrow{\mu} Q$

iii) $P \xrightarrow{\epsilon(c)} Q \xrightarrow{\mu} R \Rightarrow \exists d \leq c.P \xrightarrow{\mu}_d^* R$

iv) $P \xrightarrow{\mu}_c^* Q \wedge P \xrightarrow{\epsilon(d)} Q' \Rightarrow Q' \xrightarrow{\mu}_{c \ominus d}^* Q$

Where $x \ominus y$ equals $x - y$ if $x \geq y$ and 0 otherwise. □

As one would expect, symbolic bisimilar processes have equal life-times.

Lemma 2.15 *Using lemmas 2.7 and 2.14(i-iii) we can prove that* $P \sim^* Q \Rightarrow M(P) = M(Q)$. □

Our second main theorem reveals the fact that timed bisimulation equivalence and symbolic timed equivalence coincide on action guarded TC processes:

Theorem 2.16 $\sim = \sim^*$ □

Corollary 2.17 *Timed bisimulation equivalence ($\sim$) between* TC *processes is decidable relative to inequations between positive real numbers.* □

3 A Timed Modal Logic

We introduce a logic which allows constraints on the timed behaviour of processes to be expressed explicitly. The logic TML is an extension of the well known Hennessy–Milner Logic [HM85], and the formulae of the logic are given by the following abstract syntax:

$$F ::= tt \mid \neg F \mid F \wedge G \mid \langle \mu \rangle_{\forall \phi} F \mid \langle \mu \rangle_{\exists \phi} F$$

where μ is an action and ϕ is a time–set, i.e. $\phi \subseteq \mathcal{R}_0^+$.[1]

Intuitively, for a process to satisfy $\langle \mu \rangle_{\forall \phi} F$ *any* state reached after time–delays within the set ϕ must have a μ–derivative satisfying F. Thus, $\langle \mu \rangle_{\forall \phi} F$ specifies a property which holds invariantly for all time–delays in ϕ. Similarly, to satisfy $\langle \mu \rangle_{\exists \phi} F$ the process must after *some* time–delay within the set ϕ reach a state with a μ–derivative satisfying F. Thus, $\langle \mu \rangle_{\exists \phi} F$ specifies a property which holds eventually for some time–delay in ϕ.

3.1 Standard Interpretation

Below we give an interpretation of TML with respect to the standard semantics of the calculus TC:

Definition 3.1 $\models$ *is the (satisfaction) relation between* TC *and* TML *defined inductively as*[2]*:*

i) $P \models tt \Leftrightarrow true$

ii) $P \models \neg F \Leftrightarrow not\ (P \models F)$

iii) $P \models F \wedge G \Leftrightarrow P \models F$ *and* $P \models G$

iv) $P \models \langle \mu \rangle_{\forall \phi} F \Leftrightarrow \forall d \in \phi. \exists P' \exists P''. P \xrightarrow{\epsilon(d)} P' \xrightarrow{\mu} P'' \wedge P'' \models F$

v) $P \models \langle \mu \rangle_{\exists \phi} F \Leftrightarrow \exists d \in \phi. \exists P' \exists P''. P \xrightarrow{\epsilon(d)} P' \xrightarrow{\mu} P'' \wedge P'' \models F$

□

[1] We use $\mathcal{R}_0^+$ as abbreviation for $\mathcal{R}^+ \cup \{0\}$

[2] *Here we apply the convention that* $P \xrightarrow{\epsilon(0)} P'$ *if and only if* $P = P'$.

Note that $\langle\mu\rangle_{\forall\phi}F$ requires that the process can delay for any time-instant of ϕ. We shall often use the following derived operators:

$$i)\ ff = \neg tt,\quad ii)\ F \vee G = \neg(\neg F \wedge \neg G),\quad iii)\ [\mu]_{\exists\phi}F = \neg\langle\mu\rangle_{\forall\phi}\neg F,\quad iv)\ [\mu]_{\forall\phi}F = \neg\langle\mu\rangle_{\exists\phi}\neg F$$

Intuitively $[\mu]_{\exists\phi}F$ specifies a process which after some time-delay within ϕ may reach a state where all μ-derivatives satisfy F. Similarly, $[\mu]_{\forall\phi}F$ specifies the processes for which all μ-derivatives of states reachable by time-delays within ϕ satisfy F.

Obviously we have the following two equivalences[3]:

$$\langle\mu\rangle_{\forall\emptyset}F \equiv tt \qquad\qquad \langle\mu\rangle_{\exists\emptyset}F \equiv ff$$

Also, whenever $\phi \subseteq \psi$ then it is easy to see that the following implications hold[4]:

$$\langle\mu\rangle_{\forall\phi}F \Leftarrow \langle\mu\rangle_{\forall\psi}F \qquad\qquad \langle\mu\rangle_{\exists\phi}F \Rightarrow \langle\mu\rangle_{\exists\psi}F$$

Now, for $\phi \subseteq \mathcal{R}_0^+$ define the closure set $\phi\downarrow$ as follows:

$$\phi\downarrow \ = \ \{t \in \mathcal{R}_0^+ \mid \exists c \in \phi.\, t \leq c\}$$

Then as $\phi \subseteq \phi\downarrow$ it follows that:

$$\langle\mu\rangle_{\exists\phi}F \ \Rightarrow \ \langle\mu\rangle_{\exists\phi\downarrow}F \tag{1}$$

Note, that (1) is not an equivalence in general for the simple reason that $\langle\mu\rangle_{\exists\phi}F$ requires the life time of a process to exceed some time-instant of ϕ, and hence imposes a lower bound on the life time. In contrast the formula $\langle\mu\rangle_{\exists\phi\downarrow}F$ makes no such requirements.

Recall that the lifetime of a TC process P is the infimum over time-instants $t \in \mathcal{R}_0^+$ such that:

$$P \xrightarrow{\epsilon(t)} P' \xrightarrow{\tau} P''$$

for some P' and P''. To express that d is a lower bound for the life time of a process one may thus use the following formula:

$$\neg(\langle\tau\rangle_{\exists\{t \mid t \leq d\}}tt)$$

For ϕ a dense set[5] we are now able to turn the implication (1) into an equivalence by adding the required lower bound on life time:

$$\langle\mu\rangle_{\exists\phi}F \ \equiv \ \langle\mu\rangle_{\exists\phi\downarrow}F \wedge \neg(\langle\tau\rangle_{\exists S(\phi)}tt)$$

where $S(\phi) = \{d \in R_0^+ \mid \forall c \in \phi.\, d < c\}$. Note that for any set ϕ, both $\phi\downarrow$ and $S(\phi)$ are simple intervals either of the form $[0, t[$ for $t \in \mathcal{R}_0^+ \cup \{\infty\}$ or of the form $[0, t]$ for $t \in \mathcal{R}_0^+$.

Furthermore, it is also possible to show that the universal modality $\langle\mu\rangle_{\forall\phi}$ is indeed a derived operator. In fact the following equivalence holds:

$$\langle\mu\rangle_{\forall\phi}F \ \equiv \ \langle\mu\rangle_{\exists B(\phi)}F \wedge \neg(\langle\tau\rangle_{\exists E(\phi)}tt)$$

where $B(\phi) = \{t \in R_0^+ \mid \forall c \in \phi.\, t \leq c\}$ and $E(\phi) = \{t \in R_0^+ \mid \exists c \in \phi.\, t < c\}$. Here the first conjunct expresses that a process satisfying $\langle\mu\rangle_{\forall\phi}F$ must have an F-satisfying μ-derivative being enabled before the time-instants of ϕ. Due to the persistency property this μ-derivative will exist for all future time-instants of the process (including those of ϕ). The second conjunct ensures that the life time is greater than or equal to any time-instant of ϕ (in accordance with the semantic definition of $\langle\mu\rangle_{\forall\phi}F$).

Now consider the sublogic TML of TML which only permits *existential* quantification. Then from the above discussion we may state the following expressiveness result:

[3] Here $F \equiv G$ means that F and G are satisfied by the same TC processes.

[4] Here $F \Rightarrow G$ means that any TC process satisfying the formula F also satisfies the formula G.

[5] A set $\phi \subseteq \mathcal{R}_0^+$ is *dense* if whenever $c, d \in \phi$ then also $[c, d],]d, c] \subseteq \phi$

Theorem 3.2 *For any* TML *formula* F *there exists an equivalent* TML *formula* G. □

From the results of the next subsection it will furthermore follow that TML (and hence TML) provides an alternative characterisation of timed bisimulation equivalence between TC processes.

Theorem 3.3 *Let* P *and* Q *be* TC *processes. Then* $P \sim Q$ *holds if and only* P *and* Q *satisfy the same* TML *formulae.* □

3.2 Symbolic Interpretation

In order to provide an effective means for determining whether or not a TC process satisfies a given TML formula we offer in this section what turns out to be an equivalent interpretation of TML based on the symbolic operational semantics of TC (see section 2.2). First, for $\bowtie$ a binary relation on $[0,\infty]$, we make the obvious extension to sets. That is, for $\phi, \psi \subseteq [0,\infty]$:

$$\phi \bowtie \psi \;\Leftrightarrow\; \forall c \in \phi \forall d \in \psi.\, c \bowtie d$$

Also, we shall make no distinction between an element $d \in [0,\infty]$ and the singleton set $\{d\}$.

Definition 3.4 $\models\!\!\!\models$ *is the (satisfaction) relation between* TC *and* TML *defined inductively as:*

$$\begin{array}{ll}
i) & P \models\!\!\!\models tt \;\Leftrightarrow\; true \\
ii) & P \models\!\!\!\models \neg F \;\Leftrightarrow\; not\,(P \models\!\!\!\models F) \\
iii) & P \models\!\!\!\models F \wedge G \;\Leftrightarrow\; P \models\!\!\!\models F \text{ and } P \models\!\!\!\models G \\
iv) & P \models\!\!\!\models \langle\mu\rangle_{\forall\phi} F \;\Leftrightarrow\; (\phi \leq M(P)) \wedge \exists d \exists P'.\, d \leq \phi \wedge P \xrightarrow{\mu}{}^{*}_{d} P' \wedge P' \models\!\!\!\models F \\
v) & P \models\!\!\!\models \langle\mu\rangle_{\exists\phi} F \;\Leftrightarrow\; \exists d \exists c \in \phi. \exists P'.\, d \leq c \leq M(P) \wedge P \xrightarrow{\mu}{}^{*}_{d} P' \wedge P' \models\!\!\!\models F
\end{array}$$

□

First we verify that $\models\!\!\!\models$ is indeed equivalent to $\models$.

Theorem 3.5 *Let* P *be a* TC *process and let* F *be a* TML *formula. Then* $P \models F$ *if and only if* $P \models\!\!\!\models F$. □

Example 3.6 *Consider the coffee machine from example 2.8. We want to show that after a coin has been inserted, coffee is continuously available for 30 seconds. This property may be expressed in* TML *as:*

$$[coin]_{\forall[0,\infty[}\langle coffee\rangle_{\forall[0,30]} tt$$

As $S_0 \xrightarrow{coin}{}^{*}_{0} S_1$ *is the only symbolic coin–transition of* S_0 *and* $M(S_0) = \infty$, S_0 *will satisfy the above property just in case:*

$$S_1 \models\!\!\!\models \langle coffee\rangle_{\forall[0,30]} tt$$

which is true as $S_1 \xrightarrow{coffee}{}^{*}_{0} S_0$ *with* $0 \leq [0,30]$ *and* $[0,30] \leq M(S_1) = 30$.

The property that after any coin–insertion, there will be no coffee available after 30 seconds can be expressed in TML as:

$$[coin]_{\forall[0,\infty[}\neg\langle coffee\rangle_{\exists]30,\infty[} tt$$

To demonstrate that S_0 *satisfies this property reduces to demonstrating* $S_1 \not\models\!\!\!\models \langle coffee\rangle_{\exists]30,\infty[} tt$. *However, this is obvious as the life time of* S_1 *is exactly* 30. □

We are now able to show that TML does indeed characterise timed bisimulation between TC processes. We first consider the symbolic interpretation case.

Theorem 3.7 *Let P and Q be* TC *processes. Then the following equivalence holds:*

$$P \sim^* Q \quad \textit{if and only if} \quad \forall F.\, P \models F \Leftrightarrow Q \models F$$

□

Corollary 3.8 *Let P and Q be* TC *processes. Then the following equivalence holds:*

$$P \sim Q \quad \textit{if and only if} \quad \forall F.\, P \models F \Leftrightarrow Q \models F$$

□

Example 3.9 *Consider the following processes:*

$$P = a.\epsilon(1).b.NIL + a.\epsilon(2).b.NIL \qquad \textit{and} \qquad Q = a.(\epsilon(1).b.NIL + \epsilon(2).b.NIL)$$

Then clearly $P \not\sim^* Q$: the only possible match for the transition $P \xrightarrow{a}_0^* \epsilon(2).b.NIL$ is $Q \xrightarrow{a}_0^* \epsilon(1).b.NIL + \epsilon(2).b.NIL$. However, this is clearly not an acceptable match as $\epsilon(1).b.NIL + \epsilon(2).b.NIL \xrightarrow{b}_1^* NIL$, whereas $\epsilon(2).b.NIL \xrightarrow{b}_d^*$ for no $d \leq 1$. A property satisfied by Q but not by P is:

$$[a]_{\forall[0,\infty[}\langle b\rangle_{\exists[1,2[}tt$$

□

An important consequence of the equivalence between the standard and symbolic interpretation of TML is that the associated satisfaction problem becomes (relative) decidable.

Theorem 3.10 *The problem of satisfaction $P \models F$ for a given* TC *process P and a given* TML *formula F is* decidable *relative to first–order assertions about sets and elements of $\mathcal{R}_0^+$.* □

Moreover, it can easily be seen that the above satisfaction problem becomes decidable for TML when the intervals $]0,t[$, $]0,t]$, $[0,t[$ and $[0,t]$ are restricted to rational time–instants t.

3.3 Extended Timed Modal Logic

In this section we introduce an extension of TML where the quantification over action–transitions and time–transitions has been separated. The formulae of the logic is given by the following abstract syntax:

$$F ::= tt \mid \neg F \mid F \wedge G \mid \langle\mu\rangle F \mid \exists\phi.F$$

where $\phi \subseteq \mathcal{R}_0^+$. Below we give an interpretation of ETML with respect to the standard semantics of the calculus TC:

Definition 3.11 $\models$ *is the (satisfaction) relation between* TC *and* ETML *defined inductively as:*

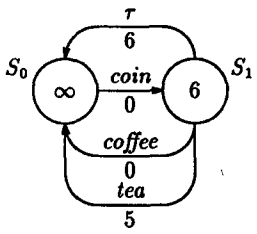
$$\begin{array}{ll}
i) & P \models tt \Leftrightarrow true \\
ii) & P \models \neg F \Leftrightarrow not\ (P \models F) \\
iii) & P \models F \wedge G \Leftrightarrow P \models F \textit{ and } P \models G \\
iv) & P \models \langle\mu\rangle F \Leftrightarrow \exists P'.\, P \xrightarrow{\mu} P' \wedge P' \models F \\
v) & P \models \exists\phi.F \Leftrightarrow \exists d \in \phi.\exists P'.\, P \xrightarrow{\epsilon(d)} P' \wedge P' \models F
\end{array}$$

□

It is easy to see that ETML is an extension of TML in the sense that for any formula of TML there exists an equivalent formula of ETML. In particular we note the following equivalence:

$$\langle\mu\rangle_{\exists\phi}F \equiv \exists\phi.\langle\mu\rangle F$$

Example 3.12 *Consider the coffee machine from example 2.8 extended with a choice for tea. The new machine behaves a little strange—for some reason—tea will not be available until after five minutes after a coin insertion.*[6] *It can be expressed in* TC *as* $rec\, x : coin.(coffee.x + \epsilon(5).tea.x + \epsilon(6).\tau.x)$ *and its symbolic transition system can then be drawn as follows*



where $S_0 = rec\, x : coin.(coffee.x + \epsilon(5).tea.x + \epsilon(6).\tau.x)$ *and* $S_1 = coffee.S_0 + \epsilon(5).tea.S_0 + \epsilon(6).\tau.S_0$. *The state* S_1 *obviously satisfies the properties: "sometimes coffee is available but no tea" and "whenever tea is available, then so is coffee" which can be expressed in* ETML *as:*

$$\exists[0,\infty[.(\langle coffee\rangle tt \wedge \neg\langle tea\rangle tt) \qquad\qquad \forall[0,\infty[.(\langle tea\rangle tt \supset \langle coffee\rangle tt)$$

where $\forall\phi.F = \neg\exists\phi.\neg F$ *and* $F \supset G = \neg F \vee G$. *These properties cannot be expressed in* TML *since* $\exists$ *does not in general distribute over* $\wedge$. *However, there are restricted forms of* $\exists$*-distributivity (see the lemma below).* □

Lemma 3.13 *The basic properties of* TC *processes given in lemma 2.3 correspond to the following laws of* ETML*:*

i)	$\langle\tau\rangle tt \Rightarrow \neg\exists]0,\infty[.tt$	*(maximal progress)*
ii)	$\neg\langle\tau\rangle tt \Rightarrow \exists]0,\infty[.tt$	*(transition liveness)*
iiia)	$\exists\{c\}.(F \wedge G) \equiv \exists\{c\}.F \wedge \exists\{c\}.G$	*(time determinacy)*
iiib)	$\exists\phi.(\langle\mu\rangle F \wedge \langle\nu\rangle G) \equiv \exists\phi.\langle\mu\rangle F \wedge \exists\phi.\langle\nu\rangle G$	*(time convergence)*
iv)	$\exists\phi.\exists\psi.F \equiv \exists(\phi+\psi).F$	*(time continuity)*
v)	$(\exists\phi.tt \wedge \langle a\rangle G) \Rightarrow \exists\phi.\langle a\rangle G$	*(persistency)*

where $\phi + \psi = \{c + d \mid c \in \phi \wedge d \in \psi\}$. □

Unfortunately, we do not know whether it is possible to provide ETML with an equivalent semantic interpretation based on the *symbolic* semantics of TC. Thus whether the satisfaction problem for ETML is (relative) decidable is left as an interesting open problem that we hope to settle in near future.

4 Conclusion and Future Work

In this paper, we have shown that the timed bisimulation equivalence for the regular real time processes of [W90] is decidable relative to comparisons between positive real numbers. Moreover, a timed modal logic has been presented, which characterises the equivalence. Model checking with respect to this logic has been shown to be decidable relative to the first-order assertions about the positive reals.

The key behind our decidability results is the introduction of a symbolic transition system, which provides a *finite* representation for each regular real time process. The symbolic semantics turns out to be consistent with the standard operational semantics of [W90] for the regular processes — for all P and Q, we have $P \sim Q$ iff $P \sim^* Q$. An open problem is how to extend the present method to deal

[6] *The machine maybe uses instant coffee powder and tea bags which must draw for five minutes.*

with parallel composition. It is not obvious whether we can give a symbolic semantics for the parallel operator, while at the same time preserving the consistency with the standard one of [W90].

In [W91] a timed action prefix $\mu@t.P$ has been introduced to achieve an expansion theorem for parallel composition, where t is a time variable and P may depend on t. Intuitively, $\mu@t.P$ denotes a process which may perform μ and become $P\{d/t\}$ where d is the time delay before μ is actually performed. The regular processes of [W91] are generated by the grammar:

$$E ::= NIL \mid x \mid \mu@t.E \mid \epsilon(e).E \mid E + F \mid rec\, x : E$$

where e ranges over the time expressions built out of the positive reals, time variables and the binary operators $+$ and $\ominus$. An exciting challenge is to develop a symbolic semantics for this set of regular processes, which is equivalent to the standard semantics. Then, by the expansion theorem, a composite process can be transformed to a regular one and we may achieve a finite graphical representation even for a composite process. This would permit the decidability results presented in this paper to be extended to composite processes. Certainly, this will be one line of research that we intend to pursue in the future.

The decidability question for model checking with respect to the extended timed modal logic also provides an interesting subject for future work.

References

[AD90] R. Alur and D.Dill, *Automata for Modelling Real-Time Systems*, LNCS 443, 1990.

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

[CPS89] R. Cleaveland, J. Parrow and B. Steffen, *The Concurrency Workbench*, LNCS 407, 1989.

[HM85] M. Hennessy and R. Milner, *Algebraic Laws for Nondeterminism and Concurrency*, JACM, Vol. 32, pp. 137-161, 1985.

[HR90] M. Hennessy and T. Regan, *A Temporal Process Algebra*, Technical Report 2/90, University of Sussex, 1990.

[M89] R. Milner, *Communication and Concurrency*, Prentice Hall International Series in Computer Science, 1989.

[MT90] F. Moller and C. Tofts, *A Temporal Calculus of Communicating Systems*, LNCS 458, 1990.

[PT87] R. Page and R.T. Tarjan, *Three Partition Refinement Algorithms*, SIAM Journal of Computing, Vol 16, no 6 Dec. 1987.

[RR86] G.M. Reed and A.W. Roscoe, *A Timed Model for Communicating Sequential Processes*, LNCS 226, 1986.

[S90] J. Sifakis etc. *ATP: an Algebra for Timed Processes*, Laboratoire de Genie Informatique, IMAG-Campus, B.P.53X, 38041 Grenoble Cedex, France, 1990.

[W90] Y. Wang, *Real Time Behaviour of Asynchronous Agents*, LNCS 458, 1990.

[W91] Y. Wang, *CCS + Time = an Interleaving Model for Real Time Systems*, ICALP'91, Madrid, 1991.