# Refinement and Consistency
# of Timed Modal Specifications[*]

Nathalie Bertrand[1] and Sophie Pinchinat[2] and Jean-Baptiste Raclet[3]

[1] INRIA Rennes, France
[2] IRISA & Université Rennes 1, France
[3] INRIA Rhône-Alpes, France

**Abstract.** In the application domain of component-based system design, developing theories which support compositional reasoning is notoriously challenging. We define *timed modal specifications*, an automata-based formalism combining modal and timed aspects. As a stepping stone to compositional approaches of timed systems, we define the notions of refinement and consistency, and establish their decidability.

## 1 Introduction

The increasing complexity of computer systems has led to methodologies almost universally based on component assembling. Because in the system development process, some pieces may not be completed or are not yet available, analysis methodologies must rely on an abstract description of the components behaviour.

Logic-based formalisms, such as modal and temporal logics, are robust formal tools to express statements about the behaviours of computer systems. Unfortunately, logics do not relate well in general to compositional approaches; the description of a system as a collection of interacting components cannot be exploited. However, by conceding a loss of expressiveness, like confining attention to safety properties, satisfactory frameworks can be developed.

A convincing proposal is the *modal specification* approach of [1][4], inspired by [3]. Modal specifications are deterministic automata equipped with two types of transitions: *may*-transitions, that are optional, as opposed to *must*-transitions, that are obligatory. Arbitrary safety properties can be expressed, as well as some elementary liveness ones by using must-transitions. Moreover, the formalism subsumes interface automata of [4] as shown in [5]. The algebraic setting developed by [1] has nice features that lead to effective methods, which we recall here.

The *consistency* of a modal specification, i.e. whether it has a model, is decidable, and the finite model property holds. Inclusion of the sets of models can also be decided since it coincides with the *refinement preorder* on modal specifications. Also, the *greatest lower bound* of two modal specifications, referred

---

[4] see [2] for a complete exposition.

to as *shared refinement* in [6], can be effectively computed. From a logic perspective, satisfiability, implication, and conjunction correspond to consistency, refinement, and greatest lower bound, respectively.

Moreover, modal specifications behave well with regard to compositional reasoning: Raclet in [1] has introduced a *product* combinator between modal specifications which reflects the parallel product of models. Furthermore, the dual *quotient* combinator is extremely relevant for the incremental design of component-based systems. Modal specification-based approaches seem thus very promising to develop formal tools in this application domain. In particular, they should be amenable to apply to the challenging domain of embedded systems, provided the framework can take real-time aspects into account.

Towards this end, the present contribution extends the algebraic framework of [1] to a timed setting. *Timed modal specifications* provide a logical formalism which combines modal and timed statements. They generalize both modal specifications and timed automata, just as timed automata generalize ordinary automata, and modal specifications generalize ordinary automata, respectively. Decision methods for refinement and consistency are achieved by bridging timed modal specifications and (untimed) modal specifications, via a region-based construction. These results are stepping stones to compositional reasoning on timed systems, by validating a timed extension of the algebraic theory of Raclet [1].

The paper is organized as follows: we define timed modal specifications in Sect. 2, and present their semantics in Sect. 3. Section 4 is dedicated to the refinement preorder, and is followed by consistency issues in Sect. 5.

## 2    Timed Automata and Timed Modal Specifications

As a modal specification is a simple automaton with a distinction between may and must transitions, a timed modal specification can be seen as a timed automaton equipped with may and must edges. In the following, we recall basics on timed automata [7], and then introduce timed modal specifications.

### 2.1    Timed Automata

Let $\mathcal{X}$ be a finite set of *clocks*. A *clock valuation* over $\mathcal{X}$ is a mapping $\nu : \mathcal{X} \to \mathbb{R}_+$, where $\mathbb{R}_+$ is the set of nonnegative reals. By $\mathcal{V}$, we represent the set of clock valuations over $\mathcal{X}$, and define $\overline{0} \in \mathcal{V}$ by $\overline{0}(x) = 0$ for all $x \in \mathcal{X}$. A *guard* over $\mathcal{X}$ is a finite conjunction of expressions of the form $x \sim c$ where $x \in \mathcal{X}$, $c \in \mathbb{N}$ is a *constant*, and $\sim \in \{<, \leq, =, \geq, >\}$. We denote by $\xi[\mathcal{X}]$ the set of guards over $\mathcal{X}$. For some fixed $N \in \mathbb{N}$, $\xi_N[\mathcal{X}]$ represents the set of guards involving expressions where the constants are smaller than or equal to $N$. The satisfaction relation $\models \subseteq \mathcal{V} \times \xi[\mathcal{X}]$ between clock valuations and guards is defined in a natural way: we write $\nu \models g$ whenever $\nu$ satisfies $g$. In the following, we will often abuse notation and write $g$ to denote the set of valuations which satisfy the guard $g$.

We consider timed automata with possibly infinitely many states (also called locations), and with a slight abuse of terminology we still call them timed automata.

**Definition 1.** *Given $\mathcal{X}$ a set of clocks, $\Sigma$ an alphabet and $N \in \mathbb{N}$, a* timed automaton *(*TA*) is a structure $\mathcal{C} = (C, c^0, \mathcal{X}, \Sigma, \delta)$ where $C$ is a (possibly infinite) set of states, $c^0$ is the initial state, and $\delta \subseteq C \times \xi_N[\mathcal{X}] \times \Sigma \times 2^{\mathcal{X}} \times C$ is a transition relation. We call $(\Sigma, \mathcal{X}, N)$ the* signature *of $\mathcal{C}$.*

From now on, we fix a signature $(\Sigma, \mathcal{X}, N)$. A *region*, denoted $\theta$, is a set of clock-valuations which satisfy exactly the same guards of $\xi_N[\mathcal{X}]$. Given a region $\theta$, we write $\mathsf{Succ}(\theta)$ for the union of all regions that can be obtained from $\theta$ by letting time elapse. We let $\Theta$ be the set of all regions.

**Definition 2.** *Given a timed automaton $\mathcal{C} = (C, c^0, \mathcal{X}, \Sigma, \delta)$, we can build its associated* region automaton *$R(\mathcal{C}) = (C \times \Theta, (c^0, \overline{0}), \Delta)$ over the alphabet $\Theta \times \Sigma \times 2^{\mathcal{X}}$. The transitions in $R(\mathcal{C})$ stem from those in $\mathcal{C}$ in the following way: for all $(c, g, a, r, c') \in \delta$, for each region $\theta$ such that $(c, \theta)$ is reachable from $(c^0, \overline{0})$, for each region $\theta'' \in \mathsf{Succ}(\theta) \cap g$, there exists $((c, \theta), \theta'', a, r, (c'\theta')) \in \Delta$, where $\theta'$ is the region obtained from $\theta''$ by resetting clocks in $r$, which we write $\theta''[r := 0]$.*

Without loss of generality, we assume that any region automaton $R(\mathcal{C})$ is pruned, i.e. all its states are reachable. As an example, a TA $\mathcal{C}$ and its region automaton $R(\mathcal{C})$ are represented in the left part of Fig. 2, on page 8.

Note that any automaton over the alphabet $\Theta \times \Sigma \times 2^{\mathcal{X}}$ can be seen as a timed automaton over the signature $(\Sigma, \mathcal{X}, N)$. We introduce the operator $T$ which transforms the former into the latter in a straightforward manner, in order to distinguish the two distinct interpretations of the same syntactic object.

**Definition 3.** *A* TA *$\mathcal{C}$ is in* normal form *if it is isomorphic to (the reachable part) of $(T \circ R)(\mathcal{C})$.*

A direct consequence of Definition 3 is the ability to associate a unique region to any state of a TA in normal form. One can easily be convinced that given a timed automaton $\mathcal{C}$, $(T \circ R)(\mathcal{C})$ is isomorphic to $(T \circ R)^2(\mathcal{C})$. As a consequence, any TA $\mathcal{C}$ can be *normalized* by letting its *normal form* be $\mathcal{C}{\downarrow} := (T \circ R)(\mathcal{C})$. Notice that normalizing a TA is safe with respect to its semantics, as stated by the following proposition:

**Proposition 1.** *Let $\mathcal{C}$ be a* TA*. The timed languages of $\mathcal{C}$ and $\mathcal{C}{\downarrow}$ coincide.*

*Proof.* We prove that the configuration graphs of $\mathcal{C}$ and $\mathcal{C}{\downarrow}$ are bisimilar, the above proposition is then a direct consequence. We recall that the configuration graph of a TA $\mathcal{C}$, defined in [7] is as follows. Vertices are configurations, and there are two kinds of edges. For all $d \in \mathbb{R}$ there is a delay-edge labeled by $d$ between configurations $(c, \nu)$ and $(c', \nu')$ whenever $c = c'$ and $\nu' = \nu + d$ (defined as $\forall x \in \mathcal{X}, (\nu + d)(x) = \nu(x) + d$). There is an action edge labeled with $a$ between $(c, \nu)$ and $(c', \nu')$ if there exists $c \xrightarrow{g,a,r} c' \in \delta_{\mathcal{C}}$ such that $\nu$ satisfies the guard $g$ and $\nu' = \nu[r := 0]$.

Given a valuation $\nu$, we denote $[\nu]$ the region it belongs to. We define the following binary relation between configurations of $\mathcal{C}$ and of $(T \circ R)(\mathcal{C})$:

$$\sim := \{((c, \nu), (c, [\nu_1]), \nu) \mid \nu \in \mathsf{Succ}([\nu_1])\}$$

and show that $\sim$ is a bisimulation. By definition, $(c^0, \overline{0}) \sim ((c^0, \overline{0}), \overline{0})$. Assume now $(c, \nu) \sim ((c, [\nu_1]), \nu)$.

Any edge $(c, \nu) \xrightarrow{d} (c, \nu + d)$ in the configuration graph of $\mathcal{C}$ can be simulated by $((c, [\nu_1]), \nu) \xrightarrow{d} (c, [\nu_1]), \nu + d)$ in the configuration graph of $R(\mathcal{C})$, and vice versa. Moreover, we indeed have $(c, \nu + d) \sim (c, [\nu_1]), \nu + d)$, since $[\nu] \in \mathsf{Succ}([\nu'])$ entails $[\nu + d] \in \mathsf{Succ}([\nu'])$.

Assume $(c, \nu) \xrightarrow{a} (c', \nu')$, because of some transition $c \xrightarrow{g, a, r} c' \in \delta_\mathcal{C}$. Necessarily, $\nu \in g$ and $\nu' = \nu[r := 0]$. Because $[\nu]$ is a time successor of $[\nu_1]$, the transition $(c, [\nu_1]) \xrightarrow{[\nu], a, r} (c', [\nu'])$ exists in the region graph $R(\mathcal{C})$ and justifies the transition $((c, [\nu_1]), \nu) \xrightarrow{[\nu], a, r} ((c', [\nu']), \nu')$.

Reciprocally, assume $((c, [\nu_1]), \nu) \xrightarrow{a} ((c', \theta'), \nu')$ in the configuration graph of $R(\mathcal{C})$. Then, there exists $(c, [\nu_1]) \xrightarrow{\theta, a, r} (c', \theta')$ in $R(\mathcal{C})$, such that (1) $\nu \in \theta$, (2) $\nu' = \nu[r := 0] \in \theta'$, and (3) $\theta \subseteq \mathsf{Succ}([\nu_1])$.

Transition $(c, [\nu_1]) \xrightarrow{\theta, a, r} (c', \theta')$ in $R(\mathcal{C})$, stem from some $c \xrightarrow{g, a, r} c'$ in $\mathcal{C}$ with (4) $\theta' = \theta[r := 0]$ and (5) $\theta \subseteq g$.

By (1) and (5), $\nu \in g$. Since $c \xrightarrow{g, a, r} c'$ is a transition in $\mathcal{C}$, there must be an edge $(c, \nu) \xrightarrow{a} (c', \nu[r := 0])$ in $\mathcal{C}$'s configuration graph.

(2), $\nu[r := 0] = \nu'$, showing that $(c, \nu) \xrightarrow{a} (c', \nu')$. It remains to establish that $(c', \nu') \sim ((c', \theta'), \nu')$, that is $[\nu'] \in \mathsf{Succ}(\theta')$, which is immediate by (4).     □

Even if this amounts to performing a normalization operation, we assume from now on that every TA is in normal form.

## 2.2   Timed modal specifications

**Definition 4.** *A* timed modal specification (TMS) *over the signature* $(\Sigma, \mathcal{X}, N)$ *is a structure* $\mathcal{S} = (Q, q^0, \mathcal{X}, \Sigma, \delta^m, \delta^M)$, *where*

- *$Q$ is a finite set of states, and $q^0 \in Q$ is the unique initial state;*
- *$\delta^m, \delta^M \subseteq Q \times \xi_N[\mathcal{X}] \times \Sigma \times 2^{\mathcal{X}} \times Q$ are finite sets of transitions, with the requirements $\delta^M \subseteq \delta^m$, and $\delta^m$ and $\delta^M$ are deterministic.*
  - *$\delta^m$ is the set of* may-*transitions representing the allowed transitions. Given a may-transition $(q, g, a, r, q') \in \delta^m$, $q$ is the source state, $q'$ is the destination state, $g \in \xi_N[\mathcal{X}]$ is the guard that specifies the valuations for which the transition can be taken, $a \in \Sigma$ is the action labeling the transition and $r \subseteq \mathcal{X}$ is the set of clocks reset by the transition.*
  - *$\delta^M$ is the set of* must-*transitions representing the required transitions.*

Determinism of the transition relation means that for every state $q$, every action $a$ and every region $\theta$, there exists at most one transition $(q, g, a, r, q') \in \delta^m$. Assuming that modal specifications are deterministic is common in the untimed case since it allows to relate refinement and inclusion of sets of models. This is not the case when nondeterminism is allowed [8]. We will often write $q \xrightarrow{g, a, r} q'$ to denote $(q, g, a, r, q') \in \delta^M$ and $q \xdashrightarrow{g, a, r} q'$ to denote $(q, g, a, r, q') \in \delta^m$.

In the next section, we give the semantics of TMS in terms of a collection of timed automata-like models.

## 3   Timed Modal Specification semantics

### 3.1   Models of Timed Modal Specifications

Models of untimed modal specifications [1] are prefix-closed languages and can be represented by (a priori infinite-state) automata. Given a TMS $\mathcal{S}$, its models are TA which relate to $\mathcal{S}$ via a simulation-like relation. Formally,

**Definition 5.** *Let $\mathcal{C} = (C, c^0, \mathcal{X}, \Sigma, \delta)$ be a TA and $\mathcal{S} = (Q, q^0, \mathcal{X}, \Sigma, \delta^m, \delta^M)$ be a TMS. $\mathcal{C}$ is a* model *of $\mathcal{S}$, written $\mathcal{C} \models \mathcal{S}$, if there exists a binary relation $\rho \subseteq C \times Q$ such that $(c^0, q^0) \in \rho$, and for all $(c, q) \in \rho$, the following holds:*

- *for every $q \xrightarrow{g,a,r} q' \in \delta^M$, and every region $\theta$ such that both $(c, \theta)$ and $(q, \theta)$ are reachable (in $C$ and $\mathcal{S}$ respectively), there exist $n \in \mathbb{N}$, states $c_1 \cdots c_n \in C$, and guards $g_1, \cdots, g_n \in \xi[\mathcal{X}]$ with*
    - $\mathsf{Succ}(\theta) \cap g \subseteq \mathsf{Succ}(\theta) \cap \bigcup_{i=1}^{n} g_i$,
    - $c \xrightarrow{g_i,a,r} c_i \in \delta, \ \forall 1 \le i \le n$, *and*
    - $(c_i, q') \in \rho, \ \forall 1 \le i \le n$.
- *for all $c \xrightarrow{g,a,r} c' \in \delta$, there exist a state $q' \in Q$ and a guard $g' \in \xi[\mathcal{X}]$ with*
    - $g \subseteq g'$,
    - $q \xdashrightarrow{g',a,r} q' \in \delta^m$, *and*
    - $(c', q') \in \rho$.

Intuitively, the first condition of Definition 5 ensures that any move required by the specification (a must-transition) is reflected in the model, potentially split in several transitions; the second condition guarantees that any transition of the model is allowed in the specification (as a may-transition). In this latter condition, notice that because $\mathcal{C}$ is in normal form, the guard $g$ in the transition $c \xrightarrow{g,a,r} c'$ is necessarily a region. Let us illustrate Definition 5 on an example.

*Example 1.* Consider the TMS $\mathcal{S}$ and TA $\mathcal{C}$ represented on Fig. 1, where dashed arrows denote transitions in $\delta^m \setminus \delta^M$ and plain arrows transitions in $\delta^M$. Also, the action alphabet $\Sigma$ is left implicit, and transitions are just labeled by guards and optional resets.
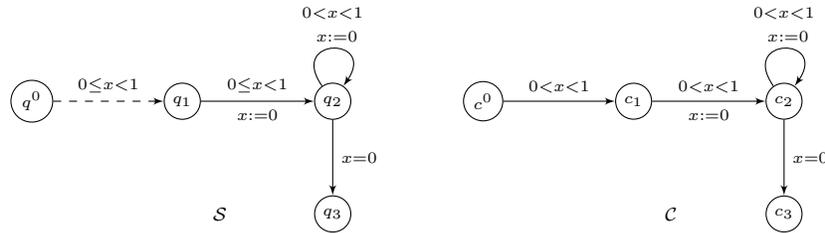


**Fig. 1.** A TMS $\mathcal{S}$ and a TA $\mathcal{C}$ with $\mathcal{C} \models \mathcal{S}$

In order to show that $\mathcal{C}$ is a model of $\mathcal{S}$ we consider the intuitive simulation relation that associates all pairs $(c_i, q_i)$ and show that this relation satisfies the conditions of Definition 5. We draw the reader's attention to two arguments: First, the transition $c^0 \xrightarrow{0 < x < 1} c_1$ is justified in $\mathcal{S}$ by a may-transition with a looser guard, namely $q^0 \xdashrightarrow{0 \le x < 1} q_1$. Second, the must-transition $q_1 \xrightarrow{0 \le x < 1, x := 0} q_2$ in $\mathcal{S}$ is correctly reflected in $\mathcal{C}$ since $c_1$ is only reachable in $\mathcal{C}$ within region $(0, 1)$. Hence it suffices to consider the intersection of the must-transition guard $(0 \le x < 1)$ with the time-successors of the reachable region $(0, 1)$ as guard for the transition in the model $\mathcal{C}$.

Let us now define the following particular TMS $\mathcal{S}_\top$ that denotes the true formula (i.e. for every TA $\mathcal{C}$, $\mathcal{C} \models \mathcal{S}_\top$) by:

$$\mathcal{S}_\top := (\{q^0\}, q^0, \delta_\top^m, \delta_\top^M) \text{ with } \delta_\top^m = \{(q^0, \mathbf{true}, a, \emptyset, q^0) \mid a \in \Sigma\} \text{ and } \delta_\top^M = \emptyset$$

In the following, we write $\mathsf{Mod}(\mathcal{S})$ for the set of models of $\mathcal{S}$ and emphasize the fact that only TA in normal form are considered.

### 3.2   The Region-based Interpretation

We show here that the semantics of TMS can be characterized by that of modal specifications via the transformation from timed automata to region automata. Modal specifications (MS), also called modal automata [3], correspond to the untimed variant of TMS. As we already dedicated significant space to TMS, we expect the reader to understand their untimed variants as objects like $\mathcal{R} = (P, p^0, Act, \Delta^m, \Delta^M)$, with an obvious interpretation of the components (states, initial state, set of actions, may-transitions, must-transitions).

An automaton $\mathcal{M} = (M, m^0, Act, \Delta)$ is a model of a MS $\mathcal{R} = (P, p^0, Act, \Delta^m, \Delta^M)$ (written $\mathcal{M} \models \mathcal{R}$) if there exists a binary relation $\rho \subseteq (M \times P)$ such that $(m^0, p^0) \in \rho$, and for all $(m, p) \in \rho$, the following holds:

– for every $p \xrightarrow{a} p' \in \Delta^M$ there is a transition $m \xrightarrow{a} m' \in \Delta$ and $(m', p') \in \rho$;
– for every $m \xrightarrow{a} m' \in \Delta$ there is a transition $p \xrightarrow{a} p' \in \Delta^m$ and $(m', p') \in \rho$.

The natural untimed object associated with a TMS $\mathcal{S}$, is its region modal automaton, which is obtained by generalizing the construction of the region automaton $R(\mathcal{C})$ for a TA $\mathcal{C}$ (Definition 2). An example of $\mathcal{S}$ and $R(\mathcal{S})$ is represented in Fig. 2 on page 8.

Notice that for every TA $\mathcal{C}$ and TMS $\mathcal{S}$ over the signature $(\Sigma, \mathcal{X}, N)$, the automaton $R(\mathcal{C})$ and the modal specification $R(\mathcal{S})$ share alphabet $\Theta \times \Sigma \times 2^{\mathcal{X}}$.

**Proposition 2.** *Let $\mathcal{M}$ be an automaton over the alphabet $\Theta \times \Sigma \times 2^{\mathcal{X}}$ and $\mathcal{S}$ be a TMS. If $\mathcal{M} \models R(\mathcal{S})$ then $T(\mathcal{M})$ is in normal form; moreover $T(\mathcal{M}) \models \mathcal{S}$.*

*Proof.* Assume $\mathcal{M}$ is a model of the modal specification $R(\mathcal{S})$ over the alphabet $\Theta \times \Sigma \times 2^{\mathcal{X}}$: each state $m$ in $\mathcal{M}$ can be decorated with a unique region $\theta_m$ reflecting the valuation of the clocks when entering $m$. By definition of $T$, $T(\mathcal{M})$ and

$\mathcal{M}$ have the same set of states. In order to distinguish between the set of states of $\mathcal{M}$ and the one of $T(\mathcal{M})$, we write the latter $T(M)$. From the simulation relation $\rho'$ between $\mathcal{M}$ and $R(\mathcal{S})$, we define $\rho \subseteq T(M) \times Q$ by $(T(m), q) \in \rho$ if there exists $\theta$ such that $(m, (q, \theta)) \in \rho'$. Note that in this case, $\theta$ is uniquely determined. Firstly, $(T(m^0), q^0) \in \rho$ since $(m^0, (q^0, \bar{0})) \in \rho'$. Let us pick $(T(m), q) \in \rho$. We prove that all requirements of the specification $\mathcal{S}$ appear in $T(\mathcal{M})$, and that all transitions in $T(\mathcal{M})$ are allowed in $\mathcal{S}$:

For any transition $q \xrightarrow{g,a,r} q'$ in $\mathcal{S}$, for any region $\theta$ such that $(q, \theta)$ is reachable in $R(\mathcal{S})$, and any region $\theta'' \in g \cap \mathsf{Succ}(\theta)$, there exists in $R(\mathcal{S})$ a must-transition $(q, \theta) \xrightarrow{\theta'',a,r} (q', \theta')$. Since $\mathcal{M}$ is a model of $R(\mathcal{S})$, there must be some transition $m \xrightarrow{\theta'',a,r} m'$ in $\mathcal{M}$ with $(m', (q', \theta')) \in \rho'$. Hence $(m', q') \in \rho$ (by definition of $\rho$). State $m$ is solely reachable for a given clock region $\theta$. Since the latter transition in $\mathcal{M}$ exists for all regions $\theta'' \in \mathsf{Succ}(\theta)$, we come with a collection of transitions in $\mathcal{M}$: $m \xrightarrow{g_i,a,r} m_i$ such that $(m_i, q') \in \rho$, and $\mathsf{Succ}(\theta) \cap g = \mathsf{Succ}(\theta) \cap \bigcup_i g_i$.

Any transition $T(m) \xrightarrow{g,a,r} T(m')$ comes from some $m \xrightarrow{g,a,r} m'$. Since $\mathcal{M}$ is a model of $R(\mathcal{S})$, there exists a may-transition $(q, \theta) \xdashrightarrow{g,a,r} (q', \theta')$ in $R(\mathcal{S})$ with $(m', (q', \theta')) \in \rho'$. Note that this implies that $g$ is a region. This transition appears in $\mathcal{S}$ under the form $q \xdashrightarrow{g',a,r} q'$, with $g \subseteq g'$ and $(m', q') \in \rho$.        $\square$

**Theorem 1.** *Let $\mathcal{C}$ be a* TA *(in normal form) and $\mathcal{S}$ be a* TMS. *Then*

$$\mathcal{C} \models \mathcal{S} \text{ if, and only if, } R(\mathcal{C}) \models R(\mathcal{S})$$

*Proof.* Let us first assume that $\mathcal{C} \models \mathcal{S}$. There exists a simulation relation $\rho$ between $\mathcal{C}$ and $\mathcal{S}$ meeting requirements of Definition 5. We define $\rho' \subseteq (C \times \Theta) \times (Q \times \Theta)$ by: $((c, \theta), (q, \theta')) \in \rho'$ if $(c, q) \in \rho$ and $\theta = \theta'$. We show that $\rho'$ is a simulation between $R(\mathcal{C})$ and $R(\mathcal{S})$, entailing $R(\mathcal{C}) \models R(\mathcal{S})$. First $((c^0, \bar{0}), (q^0, \bar{0})) \in \rho'$ by definition of $\rho'$ and since $(c^0, q^0) \in \rho$. Let us pick $((c, \theta), (q, \theta)) \in \rho'$.

Any must-transition $(q, \theta) \xrightarrow{\theta'',a,r} (q', \theta')$ in $R(\mathcal{S})$ comes from some must-transition in $\mathcal{S}$: $q \xrightarrow{g,a,r} q'$ with $\theta'' \subseteq g$. Note that $\theta'' \in \mathsf{Succ}(\theta)$. Since $\mathcal{C}$ is a model of $\mathcal{S}$, and $(c, q) \in \rho$ there are states $c_1, \cdots, c_n$ and regions $\theta_1, \cdots, \theta_n$ such that $\mathsf{Succ}(\theta) \cap g \subseteq \mathsf{Succ}(\theta) \cap \bigcup_i \theta_i$, and for all $i$ there is some transition $c \xrightarrow{\theta_i,a,r} c_i$ with $(c_i, q') \in \rho$. In particular, there is a transition in $\mathcal{C}$ of the form $c \xrightarrow{g',a,r} c'$ with $\theta'' \subseteq g'$ and $(c', q') \in \rho$. Back to $R(\mathcal{C})$, there must be a transition $(c, \theta) \xrightarrow{\theta'',a,r} (c', \theta')$ and $((c', \theta'), (q', \theta')) \in \rho'$.

Any transition $(c, \theta) \xrightarrow{\theta'',a,r} (c', \theta')$ in $R(\mathcal{C})$, comes from some $c \xrightarrow{g,a,r} c'$ in $\mathcal{C}$ with $\theta'' \subseteq g$. Since $\mathcal{C}$ is a model for $\mathcal{S}$, there must be several may-transitions in $\mathcal{S}$ of the form $q \xdashrightarrow{g_i,a,r} q_i \in \delta_{\mathcal{S}}^m$ with $g \subseteq \bigcup_i g_i$ and $(c', q_i) \in \rho$. In particular, there is a transition $q \xdashrightarrow{g',a,r} q'$ with $\theta'' \subseteq g'$ and $(c', q') \in \rho$. This transition appears in $R(\mathcal{S})$ as several transitions guarded by regions, one of which is $(q, \theta) \xdashrightarrow{\theta'',a,r} (q', \theta')$. Since $(c', q') \in \rho$, it follows that $((c', \theta'), (q', \theta')) \in \rho'$.

Assume now $R(\mathcal{C})$ is a model of the untimed modal specification $R(\mathcal{S})$ over the alphabet $\xi[\mathcal{X}] \times \Sigma \times 2^{\mathcal{X}}$. By Proposition 2, $(T \circ R)(\mathcal{C}) \models \mathcal{S}$. Since $\mathcal{C}$ is in normal form, $(T \circ R)(\mathcal{C})$ and $\mathcal{C}$ are isomorphic, hence $\mathcal{C} \models \mathcal{S}$.                    □

Note that Theorem 1 does not hold for arbitrary TA since its 'if'-part relies on the assumption that TA are in normal form. This assumption cannot be dispensed with altogether: the TA $\mathcal{C}$ of Fig. 2 is not in normal form, and is not a model of the TMS $\mathcal{S}$. And yet $R(\mathcal{C})$ is a model of $R(\mathcal{S})$. Indeed, $R(\mathcal{C})$ is obtained from $R(\mathcal{S})$ by cutting the may-transition $(q^0, 0) \xrightarrow{x=0} (q_1, 0)$, and keeping transition $(q^0, 0) \xrightarrow{x=0} (q_1, (0,1))$.
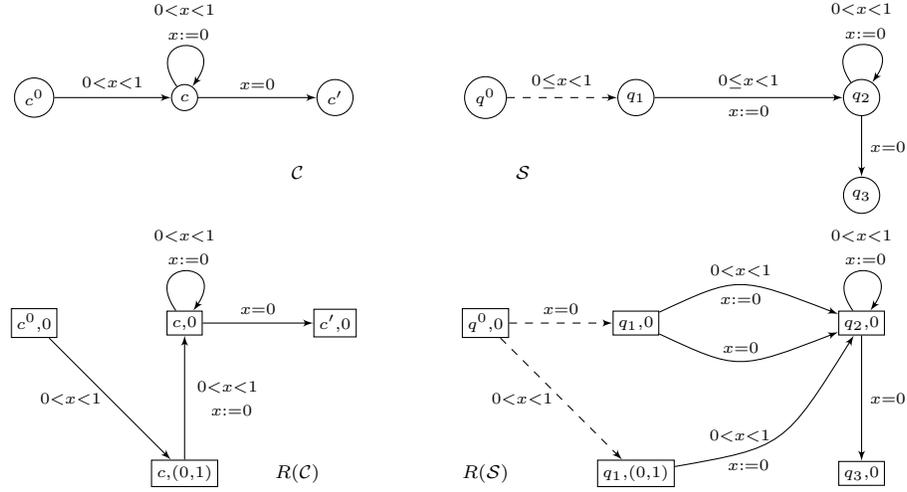


**Fig. 2.** A TA $\mathcal{C}$, a TMS $\mathcal{S}$ and their region automata

**Corollary 1.** *It is decidable, given a TMS $\mathcal{S}$ and a TA $\mathcal{C}$, whether $\mathcal{C} \models \mathcal{S}$.*

*Proof.* This is an immediate consequence of Theorem 1 and the decidability result in [1] for the model relation of untimed modal specifications.                    □

Until now, TMS are provided with a clear semantics. We now wish to provide the framework with standard logic concepts. We successively define an implication-like relation, the *refinement*, and a conjunction-like operation, *the greatest lower bound*, that yield effective methods. Recall that we fixed a signature $(\Sigma, \mathcal{X}, N)$; this is necessary when we want to talk about refinement and consistency of timed modal specifications.

## 4   Refinement

*Refinement* is a preorder between specifications, and expresses that whenever $\mathcal{S}_1$ refines $\mathcal{S}_2$, $\mathcal{S}_1$ guarantees the properties $\mathcal{S}_2$ does, and maybe others. It happens to correspond in the untimed case to the inclusion of sets of models [1]. We now investigate the refinement preorder for TMS.

Thanks to the semantics of TMS, a good candidate for the refinement preorder is the one naturally inherited from the refinement preorder between MS, which we recall here.

Given two MS $\mathcal{R}_1 = (P_1, p_1^0, Act, \Delta_1^m, \Delta_1^M)$ and $\mathcal{R}_2 = (P_2, p_2^0, Act, \Delta_2^m, \Delta_2^M)$, $\mathcal{R}_1$ is *a refinement* of (or *refines*) $\mathcal{R}_2$, written $\mathcal{R}_1 \preceq \mathcal{R}_2$, if there exists a (simulation) relation $\rho \subseteq P_1 \times P_2$ such that $(p_1^0, p_2^0) \in \rho$, and for all $(p_1, p_2) \in \rho$, the following holds:

- for every $p_2 \xrightarrow{a} p_2' \in \Delta_2^M$ there exists $p_1 \xrightarrow{a} p_1' \in \Delta_1^M$ with $(p_1', p_2') \in \rho$;
- for every $p_1 \dashrightarrow^{a} p_1' \in \Delta_1^m$ there exists $p_2 \dashrightarrow^{a} p_2' \in \Delta_2^m$ with $(p_1', p_2') \in \rho$.

**Definition 6.** *Given two* TMS $\mathcal{S}_1$ *and* $\mathcal{S}_2$, $\mathcal{S}_1$ *refines* $\mathcal{S}_2$, *written* $\mathcal{S}_1 \preceq \mathcal{S}_2$, *whenever* $R(\mathcal{S}_1) \preceq R(\mathcal{S}_2)$. *We write* $\mathcal{S}_1 \equiv \mathcal{S}_2$ *whenever* $\mathcal{S}_1 \preceq \mathcal{S}_2$ *and* $\mathcal{S}_2 \preceq \mathcal{S}_1$.

**Lemma 1.** *For every* TMS, $\mathcal{S} \equiv (T \circ R)(\mathcal{S})$

*Proof.* Lemma 1 relies on the observation that given a TMS $\mathcal{S}$, $R(\mathcal{S})$ and $R((T \circ R)(\mathcal{S}))$ are isomorphic. Therefore, they are equivalent according to the refinement preorder on (untimed) modal specifications. $\square$

Until now, we have required TMS to be finite-state (Definition 4). Actually, the framework smoothly extends to infinite-state TMS, so that TA, which may have infinitely many states, naturally embed into TMS as follows: given a TA $\mathcal{C} = (C, c^0, \mathcal{X}, \Sigma, \delta)$, we define the TMS $\mathcal{C}^* := (C, c^0, \mathcal{X}, \Sigma, \delta, \delta)$.

**Lemma 2.** *Let* $\mathcal{C}$ *be a* TA *and* $\mathcal{S}$ *a* TMS. *Then* $\mathcal{C} \models \mathcal{S}$ *if, and only if,* $\mathcal{C}^* \preceq \mathcal{S}$.

*Proof.* By Theorem 1, since $\mathcal{C}$ is in normal form, $\mathcal{C} \models \mathcal{S}$ is equivalent to $R(\mathcal{C}) \models R(\mathcal{S})$. Moreover, according to [1] for the untime setting, $R(\mathcal{C}) \models R(\mathcal{S})$ is equivalent to $R(\mathcal{C}) \models R(\mathcal{S})$, which by definition is equivalent to $\mathcal{C}^* \preceq \mathcal{S}$, since $R(\mathcal{C})$ $R(\mathcal{C}^*)$ are isomorphic. $\square$

**Theorem 2.** *It is decidable whether, for any two* TMS $\mathcal{S}_1$ *and* $\mathcal{S}_2$, $\mathcal{S}_1 \preceq \mathcal{S}_2$. *Moreover,* $\mathcal{S}_1 \preceq \mathcal{S}_2$ *if, and only if,* $\mathsf{Mod}(\mathcal{S}_1) \subseteq \mathsf{Mod}(\mathcal{S}_2)$.

*Proof.* $R(\mathcal{S}_i)$ are *deterministic* modal automata (that is, their may and must transition functions are deterministic) thus the (untimed) refinement relation coincide with the inclusion of models [1]. This entails the decidability of the refinement relation for TMS.

Let us prove now that $\mathcal{S}_1 \preceq \mathcal{S}_2 \Leftrightarrow \mathsf{Mod}(\mathcal{S}_1) \subseteq \mathsf{Mod}(\mathcal{S}_2)$. The 'only if'-part is easy: Assume $\mathcal{S}_1 \preceq \mathcal{S}_2$, and consider $\mathcal{C} \models \mathcal{S}_1$. Then by Lemma 2 $\mathcal{C}^* \preceq \mathcal{S}_1$. Therefore $\mathcal{C}^* \preceq \mathcal{S}_2$, and again by Lemma 2 $\mathcal{C} \models \mathcal{S}_2$.

For the 'if'-part, assume it not the case that $\mathcal{S}_1 \preceq \mathcal{S}_2$. By definition, $R(\mathcal{S}_1) \preceq R(\mathcal{S}_2)$ does not hold either. Then there exists an automaton $\mathcal{M}$ with $\mathcal{M} \models R(\mathcal{S}_1)$ but $\mathcal{M} \not\models R(\mathcal{S}_2)$. By Proposition 2, $T(\mathcal{M})$ is in normal form and $T(\mathcal{M}) \models \mathcal{S}_1$. However $T(\mathcal{M}) \not\models \mathcal{S}_2$, otherwise, $(R \circ T)(\mathcal{M})$ would be a model of $R(\mathcal{S}_2)$. This is impossible since $(R \circ T)(\mathcal{M})$ is isomorphic to $\mathcal{M}$. Hence $T(\mathcal{M})$ is a witness for $\mathsf{Mod}(\mathcal{S}_1) \not\subseteq \mathsf{Mod}(\mathcal{S}_2)$.                                  $\square$

We extend the class of TMS (over a fixed signature) with an extra object, written $\mathcal{S}_\perp$, for which Definition 4 does not apply: $\mathcal{S}_\perp$ has an empty set of states, and thus empty sets of transitions, and no initial state. By convention, $\mathsf{Mod}(\mathcal{S}_\perp) = \emptyset$ and we extend the refinement preorder by letting $\mathcal{S}_\perp \preceq \mathcal{S}$ for every TMS $\mathcal{S}$. Note that all the properties established so far remain valid for this slight extension. Intuitively, with the logic point of view, $\mathcal{S}_\perp$ is meant to denote an antilogy, while dually, $\mathcal{S}_\top$ (defined in Sect. 2) denotes a tautology.

**Lemma 3.** $\mathcal{S}_\perp \preceq \mathcal{S} \preceq \mathcal{S}_\top$, for any TMS $\mathcal{S}$.

## 5   Consistency

*Consistency* of a specification is a standard property in logic which expresses the existence of a model. Notice that TMS have the finite model property: indeed given a TMS $\mathcal{S}$, we can decide whether $R(\mathcal{S})$ has a model, and if so, effectively synthesize an automaton $\mathcal{M}$ model of $R(\mathcal{S})$. $T(\mathcal{M})$ is then a TA (with finitely many states), and by Proposition 2, $T(\mathcal{M})$ is model of $\mathcal{S}$.

Consistency *between a pair of specifications* can also be considered, with the meaning that the specifications share a common model. In a pure logical setting, consistency between two specifications reduces to the consistency of a single one, by considering their conjunction. In this regard, we equip the TMS with a conjunction operator derived from the one originally proposed by [1] for the untimed case. We recall this untimed-case construction.

A universal principle when operating a conjunction is to focus on the strongest constraint in the operands. When the operands are untimed modal specifications, constraints refer to the modal status (may or must) of transitions, and the conjunction of two untimed modal specifications amounts to combine transitions in the following manner (recall that dashed arrows are may-transitions and solid arrows are must-transitions): for example, if $p_1 \xrightarrow{a} p_1'$ in the first (untimed) modal specification and $p_2 \dashrightarrow^{a} p_2'$ in the second (untimed) modal specification, then $(p_1, p_2) \xrightarrow{a} (p_1', p_2')$ is a transition of their conjunction.

Formalizing wholly this idea leads to the three following rules:

$$\frac{p_1 \xrightarrow{a} p_1' \quad p_2 \dashrightarrow^{a} p_2'}{(p_1,p_2) \xrightarrow{a} (p_1',p_2')} \qquad \frac{p_1 \dashrightarrow^{a} p_1' \quad p_2 \xrightarrow{a} p_2'}{(p_1,p_2) \xrightarrow{a} (p_1',p_2')} \qquad \frac{p_1 \dashrightarrow^{a} p_1' \quad p_2 \dashrightarrow^{a} p_2'}{(p_1,p_2) \dashrightarrow^{a} (p_1',p_2')}$$

Remark that constraints of each operand can be inconsistent, e.g. when $p_1 \xrightarrow{a} p_1'$ ($a$ must occur) but there is no transition $p_2 \dashrightarrow^{a} p_2'$ (written $p_2 \not\dashrightarrow^{a}$ and meaning that $a$ is forbidden in $p_2$). In that case, the product state $(p_1, p_2)$ is *inconsistent*,

and it is modeled by the ability from the compound state $(p_1, p_2)$ to reach the particular state $\perp$ which precisely denotes inconsistency. Hence the two following additional rules.

$$\frac{p_1 \overset{a}{\longrightarrow} p_1' \qquad p_2 \overset{a}{\nrightarrow}}{(p_1, p_2) \longrightarrow \perp} \qquad \qquad \frac{p_2 \overset{a}{\longrightarrow} p_2' \qquad p_1 \overset{a}{\nrightarrow}}{(p_1, p_2) \longrightarrow \perp}$$

By the five rules above, we obtain a structure where the inconsistent state $\perp$ may be reachable. The *conjunction of the two untimed modal specifications* is the greatest sub-structure closed by must transitions and which does not contain the state $\perp$. Notice that this sub-structure may be empty. As proved by [1], the resulting delivers indeed the conjunction of the two specifications in the sense that it denotes the intersection of their models.

**Definition 7.** *The conjunction of the* TMS *$S_1$ and $S_2$, denoted $S_1 \wedge S_2$ is the* TMS $T(R(S_1) \wedge R(S_2))$. *If $R(S_1) \wedge R(S_2)$ is empty then $S_1 \wedge S_2$ is $S_\perp$.*

**Corollary 2.** *$S_1 \wedge S_2$ is the $\preceq$-greatest lower bound of $S_1$ and $S_2$.*

*Proof.* Write $\mathcal{R}_i := R(S_i)$. By [1], $\mathcal{R}_1 \wedge \mathcal{R}_2 \preceq \mathcal{R}_i$. Moreover, since $S_1 \wedge S_2 = T(\mathcal{R}_1 \wedge \mathcal{R}_2)$ and $T$ is monotonic, we have $S_1 \wedge S_2 \preceq T(\mathcal{R}_i)$. Finally because $T(\mathcal{R}_i) \equiv S_i$ (Lemma 1), $S_1 \wedge S_2 \preceq S_i$. We now show it is the greatest element under $S_1$ and $S_2$. Assume that there exists $S$ such that $S \preceq S_i$. Therefore, by definition of $\preceq$, $R(S) \preceq R_i$ which entails $R(S) \preceq R_1 \wedge R_2$. Now, we have $S \equiv T(R(S)) \preceq T(R_1 \wedge R_2)$ since $T$ is monotonic; as $S_1 \wedge S_2 = T(R_1 \wedge R_2)$ by definition, we conclude.     □

**Corollary 3.** $\mathsf{Mod}(S_1 \wedge S_2) = \mathsf{Mod}(S_1) \cap \mathsf{Mod}(S_2)$

*Proof.* From Corollary 2 we have $S_1 \wedge S_2 \preceq S_i$. Then Theorem 2 entails, $\mathsf{Mod}(S_1 \wedge S_2) \subseteq \mathsf{Mod}(S_i)$. Thus $\mathsf{Mod}(S_1 \wedge S_2) \subseteq [\mathsf{Mod}(S_1) \cap \mathsf{Mod}(S_2)]$. Let $\mathcal{C}$ be a TA such that $\mathcal{C} \in [\mathsf{Mod}(S_1) \cap \mathsf{Mod}(S_2)]$. By Lemma 2, $\mathcal{C}^* \preceq S_1$ and $\mathcal{C}^* \preceq S_2$. By Corollary 2, $\mathcal{C}^* \preceq S_1 \wedge S_2$ and by Lemma 2, $\mathcal{C} \models S_1 \wedge S_2$. As a result $[\mathsf{Mod}(S_1) \cap \mathsf{Mod}(S_2)] \subseteq \mathsf{Mod}(S_1 \wedge S_2)$.     □

## 6    Conclusion

We introduced timed modal specifications as a logical formalism to combine modal and timed statements, and provided them with a decidable notion of refinement and a computable conjunction operator.

Regarding related work, *timed interfaces* have been proposed in [9] as a timed extension of interface automata from [4]. This framework explicitly distinguishes between the output actions (from the components of the system) and the input actions (from the environment). A decidable test of compatibility is developed between, on the one hand, the respective assumptions made by each individual component on its environment and, on the other hand, the guarantees that each component provides. However, no refinement preorder is considered. We also mention the work from [10], in which so-called timed modal specifications are

introduced as well. Essentially, these are obtained as modal extensions of configuration graphs of timed automata, but presented instead as enriched CCS-like processes with durations and modalities. Because, the authors do not explicitly adopt a logical-based setting, their study of several types of refinement relations does not lead to addressing the corresponding lower-bound operators, thereby the work misses the crucial discussion about consistency.

In future work, we will extend product and quotient of [1] to timed modal specifications, borrowing know-hows from the untimed setting. Also, timed interfaces should be embeddable into timed modal specifications, via a construction in the line of [5] for the untimed setting. The main difficulty would be to establish that the compatibility relation of [9] is hereditary with respect to the present notion of refinement preorder; as a consequence, two compatible components could be implemented independently.

# References

1. Raclet, J.B.: Residual for component specifications. In: Proceedings of the 4th International Workshop on Formal Aspects of Component Software (FACS'07). (2007)
2. Raclet, J.B.: Quotient de spécifications pour la réutilisation de composants. PhD thesis, Université de Rennes I (december 2007) (In French).
3. Larsen, K.G.: Modal specifications. In: Proceedings of the International Workshop on Automatic Verification Methods for Finite State Systems. Volume 407 of Lecture Notes in Computer Science., Springer (1989) 232–246
4. de Alfaro, L., Henzinger, T.A.: Interface automata. In: Proceedings of the 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'01). (2001) 109–120
5. Larsen, K.G., Nyman, U., Wasowski, A.: Modal i/o automata for interface and product line theories. In: Proceedings of the 16th European Symposium on Programming (ESOP'07). Volume 4421 of Lecture Notes in Computer Science., Springer (2007) 64–79
6. Doyen, L., Henzinger, T.A., Jobstmann, B., Petrov, T.: Interface theories with component reuse. In: Proceedings of the 8th International Conference on Embedded Software (EMSOFT'08), ACM Press (2008) to appear.
7. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science **126**(2) (1994) 183–235
8. Larsen, K.G., Nyman, U., Wasowski, A.: On modal refinement and consistency. In: Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07). Volume 4703 of Lecture Notes in Computer Science., Springer (2007) 105–119
9. de Alfaro, L., Henzinger, T.A., Stoelinga, M.: Timed interfaces. In: Proceedings of the 2nd International Workshop on Embedded Software (EMSOFT'02). Volume 2491 of Lecture Notes in Computer Science., Springer (2002) 108–122
10. Čerāns, K., Godskesen, J.C., Larsen, K.G.: Timed modal specification - theory and tools. In: Proceedings of the 5th International Conference on Computer Aided Verification (CAV'93). Volume 697 of Lecture Notes in Computer Science., Springer (1993) 253–267