# Simulation and Bisimulation for Probabilistic Timed Automata*

Jeremy Sproston and Angelo Troina

Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy
{sproston,troina}@di.unito.it

**Abstract.** Probabilistic timed automata are an extension of timed automata with discrete probability distributions. Simulation and bisimulation relations are widely-studied in the context of the analysis of system models, with applications in the stepwise development of systems and in model reduction. In this paper, we study probabilistic timed simulation and bisimulation relations for probabilistic timed automata. We present an EXPTIME algorithm for deciding whether two probabilistic timed automata are probabilistically timed similar or bisimilar. Furthermore, we consider a logical characterization of probabilistic timed bisimulation.

## 1 Introduction

The increasing complexity of embedded and networked technologies has lead to a growing demand for formal techniques to reason about their safety, reliability and efficiency. In particular, formal modelling languages for describing systems have been developed, together with associated automatic verification techniques. We consider the case of real-time systems, in which timing information is associated with system behaviour, which can be reflected in system choices (for example, the system times-out if a response has not been received within 30ns) and in measures such as timeliness and efficiency (for example, a system is regarded as being timely if a leader is elected within 1s after a new node joins the network). A widespread example of a system description formalism for real-time systems is timed automata [1]. We also consider probabilistic systems, in which system behaviour is associated with a quantity representing its relative likelihood (for example, a message is lost with probability 0.01). When modelling probabilistic systems, it is often convenient, for representing interleaving between parallel components or for abstraction, to consider formalisms which include both nondeterministic and probabilistic choice, such as those based on Markov decision processes [2] or Segala's probabilistic automata [3, 4]. In certain cases, our aim is to model probabilistic real-time systems, for which it is important to model both timed and probabilistic behaviour within the same system model. An example of a formalism for such systems, based on a combination of timed automata

* Supported in part by the MIUR-PRIN project PaCo - Performability-Aware Computing: Logics, Models and Languages.

and Segala's probabilistic automata, is probabilistic timed automata [5]. Probabilistic timed automata have been used previously to model systems such as the IEEE 1394 root contention protocol, the backoff procedure in IEEE 802.11 Wireless LANs, and the IPv4 Zeroconf protocol [6, 7].

In the field of formal modelling of systems, reasoning about the same system at different levels of detail using the notions of *refinement* and *abstraction* is well-established. Both notions involve the use of a relation between two system models $\mathcal{A}$ and $\mathcal{B}$: the relation establishes that $\mathcal{A}$ refines $\mathcal{B}$, or, equivalently, that $\mathcal{B}$ is an abstraction of $\mathcal{A}$. These notions can be used in two different ways. Firstly, they offer mechanisms for the stepwise development of system models. That is, the system modeller starts from an abstract description of the system, then refines successively this description to obtain a detailed system model. Secondly, abstraction can be used in the context of system analysis: a system model may be too large to allow its analysis within the resources available, and therefore a smaller model which abstracts the original one can be constructed and analyzed.

An example of a relation for refinement and abstraction of system models is *simulation* [8]. This relation is defined on the states of the two models $\mathcal{A}$ and $\mathcal{B}$. If state $s_{\mathcal{B}}$ of $\mathcal{B}$ simulates state $s_{\mathcal{A}}$ of $\mathcal{A}$, then any single transition from $s_{\mathcal{A}}$ can be mimicked from $s_{\mathcal{B}}$, and the states reached by these transitions are also in the simulation relation. If the converse also holds (that is, also any single transition from $s_{\mathcal{B}}$ can be mimicked from $s_{\mathcal{A}}$), then the relation is a *bisimulation* [9, 10]. Simulation and bisimulation relations have been considered for real-time systems: in this paper we consider *timed* (rather than time-abstract) versions of these relations. Deciding timed simulation and bisimulation for timed systems is in EXPTIME [11, 12]. Similarly, deciding timed alternating (bi)simulation for timed games, which can be used to model real-time controller synthesis problems, is also in EXPTIME [13]. Simulation and bisimulation have also been considered for Markov decision processes or probabilistic automata models: deciding simulation and bisimulation can be done in polynomial time [14, 15]. The relations can also be accompanied by a logical characterization: in the case of bisimulation, this concerns in identifying a logic such that, whenever two states satisfy the same formulas of the logic, then the two states are bisimilar. The logical characterization of timed bisimulation for a subclass of timed systems has been considered in [16], whereas the logical characterization of timed alternating simulation for timed games has been presented in [13] (this result also provides a logical characterization for simulation and bisimulation for timed automata). In both cases a timed modal logic, based on Hennessy-Milner logic [17], or on temporal logic without the until operator, is considered. Instead, for probabilistic automata, a logical characterization of bisimulation has been presented in terms of a probabilistic extension of Hennessy-Milner logic [18].

In this paper we consider timed simulation and bisimulation relations for probabilistic timed automata, both in terms of algorithms for deciding such relations and in terms of a logical characterization of bisimulation. Such timed simulation and bisimulation relations for Segala's probabilistic automata enriched with timing durations have been presented in [4]. Given that probabilistic timed

automata are a generalization of both timed automata and Segala's probabilistic automata, our algorithm is inspired by [12, 13] for timing aspects, and by [14, 15] for probabilistic aspects. More precisely, a variant of the classical region graph is constructed from two probabilistic timed automata, on which an operator, which can determine whether sets of states are related using certain sub-procedures taken from [14, 15], is iterated. We show that, as for timed automata, deciding whether two probabilistic timed automata are related by (bi)simulation is EXPTIME-complete. The logical characterization that we present considers a logic in which the classical diamond operator is replaced with a diamond operator with a time constraint, as in [13], and which features probability thresholds, as in [18]. We also treat *probabilistic* timed (bi)simulation relations in the sense of [3, 4], which consider convex combinations of identically labelled transitions in order to represent randomized choice between nondeterministic alternatives.

We briefly discuss related work. Jensen and Gregersen [19, 20] presented a model similar to probabilistic timed automata, but which cannot have nondeterministic choice between transitions labelled with the same action. They considered a logical characterization of timed bisimulation for their formalism, and showed that timed bisimulation between acyclic versions of their models is decidable. Yamane [21] studied timed simulation on probabilistic timed automata. However, although introducing a region-graph construction, the possibility of obtaining an algorithm was mentioned only briefly. In particular, the key concept of a finite sampling of timing durations [11, 12] was missing, and the definition of how to relate probability distributions at the region-graph level was incomplete. Instead we provide a detailed description of an algorithm. Furthermore, we also establish that our algorithm matches the known lower bound, and consider also probabilistic timed (bi)simulation. Time-abstract bisimulation for probabilistic timed automata was considered in [22].

## 2 Probabilistic Timed Automata

**Notation.** We use $\mathbb{R}_{\geq 0}$ to denote the set of non-negative real numbers and $\mathbb{N}$ to denote the set of natural numbers. A discrete probability *distribution* over a countable set $Q$ is a function $\mu : Q \to [0, 1]$ such that $\sum_{q \in Q} \mu(q) = 1$. For a function $\mu : Q \to \mathbb{R}_{\geq 0}$ we define $\mathsf{support}(\mu) = \{q \in Q \mid \mu(q) > 0\}$. Then for an uncountable set $Q$ we define $\mathsf{Dist}(Q)$ to be the set of functions $\mu : Q \to [0, 1]$, such that $\mathsf{support}(\mu)$ is a countable set and $\mu$ restricted to $\mathsf{support}(\mu)$ is a distribution. For $q \in Q$, let $\{q \mapsto 1\}$ be the *point distribution at* $q$ which assigns probability 1 to $q$. Let $\{\mu_1, ..., \mu_k\}$ be a finite set of distributions over $Q$, and let $c_1, ..., c_k$ be a sequence of real numbers in $[0, 1]$ such that $\sum_{1 \leq i \leq k} c_i = 1$. Then the *convex combination* $\sum_{1 \leq i \leq k} c_i \mu_i$ is the distribution $\mu$ defined by $\mu(q) = \sum_{1 \leq i \leq k} c_i \mu_i(q)$ for each $q \in Q$.

**Probabilistic Timed Labelled Transition Systems.** A *probabilistic timed labelled transition system* (PTLTS) $\mathsf{P} = (S, \bar{S}, Act, \to)$ comprises the following components:

– A possibly uncountable set of *states* $S$ with initial states $\bar{S} \subseteq S$.
– A finite set $Act$ of *actions*.
– A possibly uncountable *timed, probabilistic, nondeterministic transition relation* $\rightarrow \subseteq S \times \mathbb{R}_{\geq 0} \times Act \times \mathsf{Dist}(S)$.

The transitions from state to state of a PTLTS are performed in two steps: given that the current state is $s$, the first step concerns a nondeterministic selection of $(s, d, a, \mu) \in \rightarrow$, where $d$ and $a$ are the duration and the action of the transition, respectively; the second step comprises a probabilistic choice, made according to the distribution $\mu$, as to which state to make the transition to (that is, we make a transition to a state $s' \in S$ with probability $\mu(s')$).

**Syntax of Probabilistic Timed Automata.** Let $\mathcal{X}$ be a finite set of real-valued variables called *clocks*, the values of which increase at the same rate as real-time. The set $CC(\mathcal{X})$ of *clock constraints* over $\mathcal{X}$ is defined as the set of conjunctions over atomic formulas of the form $x \sim c$, where $x, y \in \mathcal{X}$, $\sim \in \{<, \leq, >, \geq, =\}$, and $c \in \mathbb{N}$.

A *probabilistic timed automaton* (PTA) $\mathcal{A} = (L, \bar{L}, Act, \mathcal{X}, inv, prob)$ is a tuple consisting of the following components:

– A finite set $L$ of *locations* with the initial locations $\bar{L} \subseteq L$.
– A finite set $\mathcal{X}$ of clocks.
– A finite set $Act$ of actions.
– A function $inv : L \rightarrow CC(\mathcal{X})$ associating an *invariant condition* with each location.
– A finite set $prob \subseteq L \times CC(\mathcal{X}) \times Act \times \mathsf{Dist}(2^{\mathcal{X}} \times L)$ of *probabilistic edges*.

A probabilistic edge $(l, g, a, \mathsf{p}) \in prob$ is a tuple containing (1) a source location $l$, (2) a clock constraint $g$, called a *guard*, (3) an action $a$, and (4) a probability distribution $\mathsf{p}$ which assigns probability to pairs of the form $(X, l')$, where $X$ is a set of clocks to be reset and $l'$ is a location. The behaviour of a PTA takes a similar form to that of a timed automaton [1]: in any location time can advance as long as the invariant holds, and a probabilistic edge can be taken if its guard is satisfied by the current values of the clocks. PTA generalize timed automata in the sense that, once a probabilistic edge is nondeterministically selected, then the choice of which clocks to reset and which target location to make the transition to is *probabilistic*.

The size $|\mathcal{A}|$ of the PTA $\mathcal{A}$ is $|L| + |\mathcal{X}| + |inv| + |prob|$, where $|inv|$ represents the size of the binary encoding of the constants used in the invariant condition, and $|prob|$ includes the size of the binary encoding of the constants used in guards and the probabilities used in probabilistic edges (probabilities are expressed as a ratio between two natural numbers, each written in binary).

**Semantics of Probabilistic Timed Automata.** We refer to a mapping $v : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ as a *clock valuation*. Let $\mathbb{R}^{\mathcal{X}}_{\geq 0}$ denote the set of clock valuations. Let $\mathbf{0} \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ be the clock valuation which assigns 0 to all clocks in $\mathcal{X}$. For a clock

valuation $v \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ and a value $d \in \mathbb{R}_{\geq 0}$, we use $v+d$ to denote the clock valuation obtained by letting $(v+d)(x) = v(x) + d$ for all clocks $x \in \mathcal{X}$. For a clock set $X \subseteq \mathcal{X}$, we let $v[X := 0]$ be the clock valuation obtained from $v$ by resetting all clocks in $X$ to 0; formally, we let $v[X := 0](x) = 0$ for all $x \in X$, and let $v[X := 0](x) = v(x)$ for all $x \in \mathcal{X} \setminus X$. The clock valuation $v$ *satisfies* the clock constraint $\varphi \in CC(\mathcal{X})$, written $v \models \varphi$, if and only if $\varphi$ resolves to true after substituting each clock $x \in \mathcal{X}$ with the corresponding clock value $v(x)$.

The semantics of the PTA $\mathcal{A} = (L, \bar{L}, Act, \mathcal{X}, inv, prob)$ is the PTLTS $[\![\mathcal{A}]\!] = (S, \bar{S}, Act, \rightarrow)$ where:

- $S = \{(l, v) \mid l \in L \text{ and } v \in \mathbb{R}^{\mathcal{X}}_{\geq 0} \text{ s.t. } v \models inv(l)\}$ and $\bar{S} = \{(l, \mathbf{0}) \mid l \in \bar{L}\}$;
- $\rightarrow$ is the smallest set such that $((l, v), d, a, \mu) \in \rightarrow$ if there exist $d \in \mathbb{R}_{\geq 0}$ and a probabilistic edge $(l, g, a, \mathsf{p}) \in prob$ such that:
  1. $v + d \models g$, and $v + d' \models inv(l)$ for all $0 \leq d' \leq d$;
  2. for any $(X, l') \in 2^{\mathcal{X}} \times L$, if $\mathsf{p}(X, l') > 0$ then $(v + d)[X := 0] \models inv(l')$;
  3. for any $(l', v') \in S$, we have that $\mu(l', v') = \sum_{X \in \mathsf{Reset}(v,d,v')} \mathsf{p}(X, l')$, where $\mathsf{Reset}(v, d, v') = \{X \subseteq \mathcal{X} \mid (v + d)[X := 0] = v'\}$.

Given the state $(l, v)$ and the duration $d \in \mathbb{R}_{\geq 0}$ such that $v + d' \models inv(l)$ for all $0 \leq d' \leq d$, in the sequel we often write $(l, v) + d$ to denote the state $(l, v + d)$. By abuse of notation, we also write $((l, v), d, a, \mathsf{p}) \in \rightarrow$ to denote the existence of $((l, v), d, a, \mu) \in \rightarrow$ such that a probabilistic edge $(l, \_, \_, \mathsf{p}) \in prob$ is used to define $((l, v), d, a, \mu)$ according to the second point in the definition of the semantic PTLTS of the PTA.

**Composition of Probabilistic Timed Automata.** To aid higher-level modelling, it is often useful to define complex systems as the *parallel composition* of a number of interacting sub-components. The definition of the parallel composition operator $\|$ of PTA uses ideas from the theory of (untimed) probabilistic automata [3] and classical timed automata [1], and was presented in [6]. Let $\mathcal{A}_i = (L_i, \bar{L}_i, Act_i, \mathcal{X}_i, inv_i, prob_i)$ for $i \in \{1, 2\}$ and assume that $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$. Given $\mathsf{p}_1 \in \mathsf{Dist}(2^{\mathcal{X}_1} \times L_1)$ and $\mathsf{p}_2 \in \mathsf{Dist}(2^{\mathcal{X}_2} \times L_2)$, we define the distribution $\mathsf{p}_1 \otimes \mathsf{p}_2 \in \mathsf{Dist}(2^{\mathcal{X}_1 \cup \mathcal{X}_2} \times (L_1 \times L_2))$ in the following way: for each $X_1 \subseteq \mathcal{X}_1$, $X_2 \subseteq \mathcal{X}_2$, $l_1 \in L_1$ and $l_2 \in L_2$, let $\mathsf{p}_1 \otimes \mathsf{p}_2(X_1 \cup X_2, (l_1, l_2)) = \mathsf{p}_1(X_1, l_1) \cdot \mathsf{p}_2(X_2, l_2)$. The *parallel composition* of two PTA $\mathcal{A}_1$ and $\mathcal{A}_2$ is the PTA

$$\mathcal{A}_1 \| \mathcal{A}_2 = (L_1 \times L_2, \bar{L}_1 \times \bar{L}_2, Act_1 \cup Act_2, \mathcal{X}_1 \cup \mathcal{X}_2, inv, prob)$$

such that

- $inv(l_1, l_2) = inv_1(l_1) \wedge inv_2(l_2)$ for all $(l_1, l_2) \in L_1 \times L_2$;
- $((l_1, l_2), g, a, \mathsf{p}) \in prob$ if and only if one of the following conditions holds:
  1. $a \in Act_1 \setminus Act_2$ and there exists $(l_1, g, a, \mathsf{p}_1) \in prob_1$ such that $\mathsf{p} = \mathsf{p}_1 \otimes \{(\emptyset, l_2) \mapsto 1\}$;
  2. $a \in Act_2 \setminus Act_1$ and there exists $(l_2, g, a, \mathsf{p}_2) \in prob_2$ such that $\mathsf{p} = \{(\emptyset, l_1) \mapsto 1\} \otimes \mathsf{p}_2$;
  3. $a \in Act_1 \cap Act_2$ and there exists $(l_i, g_i, a, \mathsf{p}_i) \in prob_i$ for $i = 1, 2$ such that $g = g_1 \wedge g_2$ and $\mathsf{p} = \mathsf{p}_1 \otimes \mathsf{p}_2$.

## 3 Algorithms for Timed Simulation and Bisimulation

**Timed Simulation and Bisimulation.** We now define probabilistic timed simulation in the manner of [4, 21]. Given two sets $Q_1$ and $Q_2$, let $\mathcal{R} \subseteq Q_1 \times Q_2$ be a binary relation. Let $\mu_1 \in \mathsf{Dist}(Q_1)$ and $\mu_2 \in \mathsf{Dist}(Q_2)$ be distributions on $Q_1$ and $Q_2$, respectively. A *weight function* [23] for $(\mu_1, \mu_2)$ with respect to $\mathcal{R}$ is a function $\Delta : Q_1 \times Q_2 \to [0, 1]$ such that:

1. $\Delta(q_1, q_2) > 0$ implies $(q_1, q_2) \in \mathcal{R}$;
2. $\sum_{q_2 \in Q_2} \Delta(q_1, q_2) = \mu_1(q_1)$ for each $q_1 \in Q_1$;
3. $\sum_{q_1 \in Q_1} \Delta(q_1, q_2) = \mu_2(q_2)$ for each $q_2 \in Q_2$.

The *lifting* of $\mathcal{R}$ is a relation $\mathcal{L}(\mathcal{R}) \subseteq \mathsf{Dist}(Q_1) \times \mathsf{Dist}(Q_2)$ such that $\mu_1 \mathcal{L}(\mathcal{R}) \mu_2$ if there exists a weight function for $(\mu_1, \mu_2)$ with respect to $\mathcal{R}$. When clear from the context, we use $\mathcal{R}$ also to refer to the lifting $\mathcal{L}(\mathcal{R})$.

Let $\mathsf{P} = (S, \bar{S}, Act, \to)$ be a PTLTS. A binary relation $\mathcal{R} \subseteq S \times S$ is a *timed simulation* if $s_1 \mathcal{R} s_2$ implies that, for each $(s_1, d, a, \mu_1) \in \to$, there exists $(s_2, d, a, \mu_2) \in \to$ such that $\mu_1 \mathcal{R} \mu_2$. Given two states $s_1, s_2 \in S$, we write $s_1 \preceq s_2$ if there exists a timed simulation $\mathcal{R}$ such that $s_1 \mathcal{R} s_2$. A *timed bisimulation* is a symmetric timed simulation. Given two states $s_1, s_2 \in S$, we write $s_1 \approx s_2$ if there exists a timed bisimulation $\mathcal{R}$ such that $s_1 \mathcal{R} s_2$.

Let $s \in S$, $d \in \mathbb{R}_{\geq 0}$ and $a \in Act$. Consider the largest set $\{\mu_1, ..., \mu_k\}$ of distributions over $S$ such that $(s, d, a, \mu_i) \in \to$ for $1 \leq i \leq k$. Then the tuple $(s, d, a, \mu)$ is a *combined transition* if there exists a sequence $c_1, ..., c_k$ of real numbers in $[0, 1]$ such that $\sum_{1 \leq i \leq k} c_i = 1$ where $\mu = \sum_{1 \leq i \leq k} c_i \mu_i$. We let $\mathsf{Combined}(s, d, a)$ denote the set of combined transitions associated with $s$, $d$ and $a$. A binary relation $\mathcal{R} \subseteq S \times S$ is a *probabilistic timed simulation* if $s_1 \mathcal{R} s_2$ implies that, for each $(s_1, d, a, \mu_1) \in \to$, there exists a combined transition $(s_2, d, a, \mu_2) \in \mathsf{Combined}(s_2, d, a)$ such that $\mu_1 \mathcal{R} \mu_2$. Given two states $s_1, s_2 \in S$, we write $s_1 \preceq^p s_2$ if there exists a probabilistic timed simulation $\mathcal{R}$ such that $s_1 \mathcal{R} s_2$. A *probabilistic timed bisimulation* is a symmetric probabilistic timed simulation. Given two states $s_1, s_2 \in S$, we write $s_1 \approx^p s_2$ if there exists a probabilistic timed bisimulation $\mathcal{R}$ such that $s_1 \mathcal{R} s_2$.

Let $\mathcal{A} = (L_\mathcal{A}, \bar{L}_\mathcal{A}, Act_\mathcal{A}, \mathcal{X}_\mathcal{A}, inv_\mathcal{A}, prob_\mathcal{A})$ and $\mathcal{B} = (L_\mathcal{B}, \bar{L}_\mathcal{B}, Act_\mathcal{B}, \mathcal{X}_\mathcal{B}, inv_\mathcal{B}, prob_\mathcal{B})$ be two PTA. The disjoint composition of $\mathcal{A}$ and $\mathcal{B}$ is the PTA $\mathcal{A} \uplus \mathcal{B} = (L_\mathcal{A} \uplus L_\mathcal{B}, \bar{L}_\mathcal{A} \uplus \bar{L}_\mathcal{B}, Act_\mathcal{A} \uplus Act_\mathcal{B}, \mathcal{X}_\mathcal{A} \uplus \mathcal{X}_\mathcal{B}, inv, prob_\mathcal{A} \uplus prob_\mathcal{B})$, where $inv(l) = inv_\mathcal{A}(l)$ if $l \in L_\mathcal{A}$, and $inv(l) = inv_\mathcal{B}(l)$ if $l \in L_\mathcal{B}$. Given $\mathcal{A} \uplus \mathcal{B}$, a (probabilistic) timed simulation relation $\mathcal{R}$ on $[\![\mathcal{A} \uplus \mathcal{B}]\!]$ is *initialized* if and only if, for every $l_\mathcal{A} \in \bar{L}_\mathcal{A}$, there exists some $l_\mathcal{B} \in \bar{L}_\mathcal{B}$ such that $(l_\mathcal{A}, \mathbf{0}) \mathcal{R} (l_\mathcal{B}, \mathbf{0})$. We write $\mathcal{A} \preceq \mathcal{B}$ if there exists an initialized (probabilistic) timed simulation relation on $[\![\mathcal{A} \uplus \mathcal{B}]\!]$. It follows from [4] that $\preceq$ and $\preceq^p$ are preorders, and, together with [12], that $\preceq$ and $\preceq^p$ are compositional in the following sense: given the PTA $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, if $\mathcal{A} \preceq \mathcal{B}$ then $\mathcal{A} \| \mathcal{C} \preceq \mathcal{B} \| \mathcal{C}$, and if $\mathcal{A} \preceq^p \mathcal{B}$ then $\mathcal{A} \| \mathcal{C} \preceq^p \mathcal{B} \| \mathcal{C}$.

*Example 1.* Consider the two PTA fragments $\mathcal{A}$ (left) and $\mathcal{B}$ (right) in Figure 1. We write the invariant conditions within the locations they refer to and we omit them when they are `true`. A probabilistic edge $(l, g, a, \mathsf{p})$ is represented as an

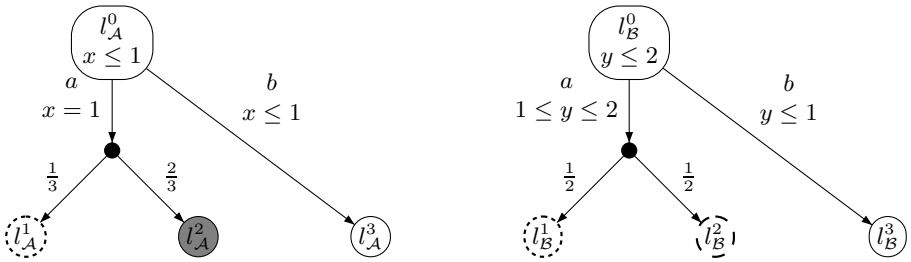


**Fig. 1.** An example of timed simulation.

arc exiting location $l$, labeled with the action $a$ and guard $g$. The distribution $\mathsf{p}$ is represented by connecting the location $l$ to a dot from which arcs, labeled with a probability, reach the locations indicated by the elements of $\mathsf{support}(\mathsf{p})$. For simplicity, when $\mathsf{p}(\emptyset, l') = 1$, we draw a direct arc from $l$ to $l'$. The initial locations of $\mathcal{A}$ and $\mathcal{B}$ are $l^0_{\mathcal{A}}$ and $l^0_{\mathcal{B}}$, respectively. Assume that there is subsequent behaviour from the bottom line of locations in the figure such that, for all clock valuations $v \in \mathbb{R}^{\{x\}}_{\ge 0}$ and $v' \in \mathbb{R}^{\{y\}}_{\ge 0}$, we have $(l^1_{\mathcal{A}}, v) \preceq (l^1_{\mathcal{B}}, v')$, $(l^2_{\mathcal{A}}, v) \preceq (l^1_{\mathcal{B}}, v')$, $(l^2_{\mathcal{A}}, v) \preceq (l^2_{\mathcal{B}}, v')$ and $(l^3_{\mathcal{A}}, v) \preceq (l^3_{\mathcal{B}}, v')$ (states in $\preceq$ are indicated by locations of the same color and shape, except for states with the gray location, which are timed simulated by the states of the short and long dashed shape). From $(l^0_{\mathcal{A}}, \mathbf{0})$, there exist transitions $((l^0_{\mathcal{A}}, \mathbf{0}), d, b, \{(l^3_{\mathcal{A}}, v_d) \mapsto 1\}) \in \rightarrow_{\mathcal{A}}$, which can be mimicked by transitions $((l^0_{\mathcal{B}}, \mathbf{0}), d, b, \{(l^3_{\mathcal{B}}, v'_d) \mapsto 1\}) \in \rightarrow_{\mathcal{B}}$ from $(l^0_{\mathcal{B}}, \mathbf{0})$, where $d \le 1$ and $v_d(x) = v'_d(y) = d$. From $(l^0_{\mathcal{A}}, \mathbf{0})$, there exists the single $a$-labelled transition $((l^0_{\mathcal{A}}, \mathbf{0}), 1, a, \mu_{\mathcal{A}}) \in \rightarrow_{\mathcal{A}}$ such that $\mu_{\mathcal{A}}(l^1_{\mathcal{A}}, v_1) = \frac{1}{3}$ and $\mu_{\mathcal{A}}(l^2_{\mathcal{A}}, v_1) = \frac{2}{3}$. This transition can be mimicked from $(l^0_{\mathcal{B}}, \mathbf{0})$ by the transition $((l^0_{\mathcal{B}}, \mathbf{0}), 1, a, \mu_{\mathcal{B}}) \in \rightarrow_{\mathcal{B}}$ such that $\mu_{\mathcal{B}}(l^1_{\mathcal{B}}, v'_1) = \frac{1}{2}$ and $\mu_{\mathcal{B}}(l^2_{\mathcal{B}}, v'_1) = \frac{1}{2}$. Furthermore, there exists a weight function $\Delta$ for $(\mu_{\mathcal{A}}, \mu_{\mathcal{B}})$ with respect to $\preceq$: we can consider $\Delta((l^1_{\mathcal{A}}, v_1), (l^1_{\mathcal{B}}, v_1)) = \frac{1}{3}$, $\Delta((l^2_{\mathcal{A}}, v_1), (l^1_{\mathcal{B}}, v_1)) = \frac{1}{6}$ and $\Delta((l^2_{\mathcal{A}}, v_1), (l^2_{\mathcal{B}}, v_1)) = \frac{1}{2}$. It can be verified that $\Delta$ satisfies the conditions of a weight function for $(\mu_{\mathcal{A}}, \mu_{\mathcal{B}})$ with respect to $\preceq$. Hence we have $(l^0_{\mathcal{A}}, \mathbf{0}) \preceq (l^0_{\mathcal{B}}, \mathbf{0})$. From this, we conclude that $\mathcal{A} \preceq \mathcal{B}$.

*Example 2.* Consider the two PTA fragments $\mathcal{A}$ (left) and $\mathcal{B}$ (right) in Figure 2. Here we suppose that, for all clock valuations $v \in \mathbb{R}^{\{x\}}_{\ge 0}$ and $v' \in \mathbb{R}^{\{y\}}_{\ge 0}$, we have $(l^1_{\mathcal{A}}, v) \preceq (l^1_{\mathcal{B}}, v')$, $(l^2_{\mathcal{A}}, v) \preceq (l^2_{\mathcal{B}}, v')$, $(l^1_{\mathcal{A}}, v) \preceq (l^3_{\mathcal{B}}, v')$ and $(l^2_{\mathcal{A}}, v) \preceq (l^4_{\mathcal{B}}, v')$. It holds that $\mathcal{A} \not\preceq \mathcal{B}$, because $\mathcal{A}$ can reach a location $l^1_{\mathcal{A}}$ in a single step with probability $\frac{1}{2}$, while $\mathcal{B}$ can reach a related location ($l^1_{\mathcal{B}}$ or $l^3_{\mathcal{B}}$) either with probability $\frac{1}{3}$ or $\frac{2}{3}$, but not with probability $\frac{1}{2}$. However, there exists a combined transition for $\mathcal{B}$ obtained by assigning $\frac{1}{2}$ to the two illustrated probabilistic edges from $l^0_{\mathcal{B}}$, and for which it is possible to reach $l^1_{\mathcal{B}}$ or $l^3_{\mathcal{B}}$ with probability $\frac{1}{2}$. Continuing this reasoning also for $l^2_{\mathcal{A}}$, we can verify that $\mathcal{A} \preceq^p \mathcal{B}$.

We now present an algorithm for deciding whether a PTA (probabilistically) timed simulates another PTA. Our approach is to extend the techniques of [12, 13], which were applied to non-probabilistic timed automata/timed games, to the case of PTA. We focus our attention on the case of timed simulation. Formally,

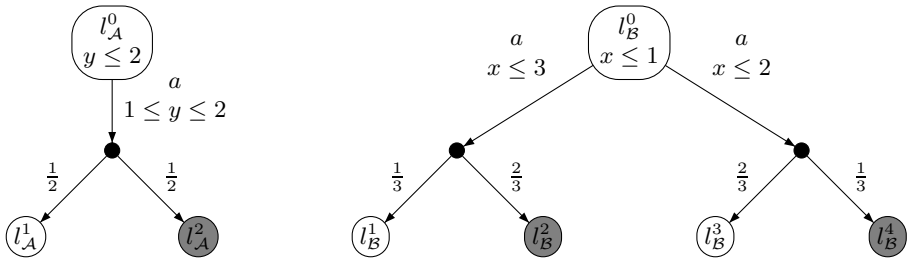


**Fig. 2.** An example of probabilistic timed simulation.

for two PTA $\mathcal{A}$ and $\mathcal{B}$, our aim is to decide whether $\mathcal{A} \preceq \mathcal{B}$. We comment briefly on how to extend the algorithm to the case of timed bisimulation, and to probabilistic timed (bi)simulation, at the end of this section.

**Region Equivalence.** We begin by recalling the standard definition of region equivalence [1]. For $r \in \mathbb{R}_{\geq 0}$, we let $\mathsf{frac}(r) = r - \lfloor r \rfloor$. Let $\mathcal{A} = (L, \bar{L}, Act, \mathcal{X}, inv, prob)$ be a PTA, and let $c_{max}$ be the maximal constant to which a clock is compared in any of the guards of probabilistic edges or invariants of $\mathcal{A}$. Two clock valuations $v, v' \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ are *clock equivalent* if the following conditions are satisfied: (1) for all clocks $x \in \mathcal{X}$, we have $v(x) \leq c_{max}$ if and only if $v'(x) \leq c_{max}$; (2) for all clocks $x \in \mathcal{X}$ with $v(x) \leq c_{max}$, we have $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$; (3) for all clocks $x, y \in \mathcal{X}$ with $v(x) \leq c_{max}$ and $v(y) \leq c_{max}$, we have $\mathsf{frac}(v(x)) \leq \mathsf{frac}(v(y))$ if and only if $\mathsf{frac}(v'(x)) \leq \mathsf{frac}(v'(y))$; and (4) for all clocks $x \in \mathcal{X}$ with $v(x) \leq c_{max}$, we have $\mathsf{frac}(v(x)) = 0$ if and only if $\mathsf{frac}(v'(x)) = 0$. Two states $(l, v), (l', v')$ of $[\![\mathcal{A}]\!]$ are *region equivalent*, written $(l, v) \equiv (l', v')$, if (1) $l = l'$, and (2) $v$ and $v'$ are clock equivalent. A *region* is an equivalence class of region equivalence, and let $\mathsf{Regions}_{\mathcal{A}}$ be the set of regions of $\mathcal{A}$. Given a state $(l, v)$, we use $[(l, v)]$ to denote the region to which $(l, v)$ belongs. The number of regions corresponding to the PTA $\mathcal{A}$ is bounded by $|L| \cdot (2c_{max} + 2)^{|\mathcal{X}|} \cdot |\mathcal{X}|! \cdot 2^{|\mathcal{X}|}$.

For deciding timed simulation on two PTA, we consider region equivalence over the state space of the parallel composition of the PTA. The subsequent algorithm for deciding whether $\mathcal{A} \preceq \mathcal{B}$ operates on the set of regions $\mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$. In the following, given two states $(l_{\mathcal{A}}, v_{\mathcal{A}})$ of $\mathcal{A}$ and $(l_{\mathcal{B}}, v_{\mathcal{B}})$ of $\mathcal{B}$, the unique state of $\mathcal{A}\|\mathcal{B}$ corresponding to these states is written $((l_{\mathcal{A}}, l_{\mathcal{B}}), v_{\mathcal{A}\|\mathcal{B}})$, where $v_{\mathcal{A}\|\mathcal{B}}(x) = v_{\mathcal{A}}(x)$ if $x \in \mathcal{X}_{\mathcal{A}}$, and $v_{\mathcal{A}\|\mathcal{B}}(x) = v_{\mathcal{B}}(x)$ if $x \in \mathcal{X}_{\mathcal{B}}$ (note that it is possible that $v_{\mathcal{A}\|\mathcal{B}} \not\models inv_{\mathcal{A}\|\mathcal{B}}(l_{\mathcal{A}}, l_{\mathcal{B}})$, where $inv_{\mathcal{A}\|\mathcal{B}}$ denotes the invariant condition of $\mathcal{A}\|\mathcal{B}$; it is trivial to decide timed simulation on such states, and henceforth we do not consider states of this form).

**Restriction to a Finite Set of Time Durations.** Let $((l_{\mathcal{A}}, v_{\mathcal{A}}), (l_{\mathcal{B}}, v_{\mathcal{B}})) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$. Let $x_1, ..., x_n \subseteq \mathcal{X}_{\mathcal{A}} \cup \mathcal{X}_{\mathcal{B}}$ be the clocks such that $v_{\mathcal{A}\|\mathcal{B}}(x_i) < c_{max}$ for each $1 \leq i \leq n$, ordered such that $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_n$, where $\tau_i = \mathsf{frac}(v_{\mathcal{A}\|\mathcal{B}}(x_i))$ for each $1 \leq i \leq n$. Furthermore, let $\tau_0 = 0$ and $\tau_{n+1} = 1$. We also define $\min(v_{\mathcal{A}}, v_{\mathcal{B}}) = \min\{\lfloor v_{\mathcal{A}\|\mathcal{B}}(x_1) \rfloor, \cdots, \lfloor v_{\mathcal{A}\|\mathcal{B}}(x_n) \rfloor\}$. We now recall the definition

of a *finite* set $Times((l_{\mathcal{A}}, v_{\mathcal{A}}), (l_{\mathcal{B}}, v_{\mathcal{B}}))$ of time durations from $[12, 13]$; it will suffice to consider only the time durations in this set in the subsequent algorithm.

$$
\begin{array}{l}
Times((l_{\mathcal{A}}, v_{\mathcal{A}}), (l_{\mathcal{B}}, v_{\mathcal{B}})) = \\
\quad \{c - \frac{1}{2}(\tau_i + \tau_{i+1}) \mid c \in \mathbb{N} \text{ and } 0 \leq i \leq n \text{ and } 1 \leq c \leq c_{max} - \min(v_{\mathcal{A}}, v_{\mathcal{B}})\} \cup \\
\quad \{c - \tau_i \mid c \in \mathbb{N} \text{ and } 1 \leq i \leq n \text{ and } 1 \leq c \leq c_{max} - \min(v_{\mathcal{A}}, v_{\mathcal{B}})\} \cup \\
\quad \{c \mid c \in \mathbb{N} \text{ and } 0 \leq c \leq c_{max} + 1 - \min(v_{\mathcal{A}}, v_{\mathcal{B}})\} \ .
\end{array}
$$

Let $I = \{c \mid c \in \mathbb{N} \text{ and } 1 \leq c \leq c_{max} - \min(v_{\mathcal{A}}, v_{\mathcal{B}})\}$. The finite set of durations $Times((l_{\mathcal{A}}, v_{\mathcal{A}}), (l_{\mathcal{B}}, v_{\mathcal{B}}))$ contains: (1) the distances between the mid-points of the intervals $(\tau_i, \tau_{i+1})$ and the integers in $I$, (2) the distances between $\tau_i$ and the integers in $I$, (3) the set of integers in $\{c \mid c \in \mathbb{N} \text{ and } 1 \leq c \leq c_{max} - \min(v_{\mathcal{A}}, v_{\mathcal{B}})\}$. Following [12, 13], the distance $d \in Times((l_{\mathcal{A}}, v_{\mathcal{A}}), (l_{\mathcal{B}}, v_{\mathcal{B}}))$ between the mid-point $\frac{1}{2}(\tau_i + \tau_{i+1})$ and an integer $c \in I$ can be used as a representative for all the time delays between $c - \tau_{i+1}$ and $c - \tau_i$.

**One-Step Goodness.** We now define two notions of "goodness", which we will use subsequently to refer to a single transition step from each of the PTA $\mathcal{A}$ and $\mathcal{B}$. This notion will be presented in two versions: a concrete version, defined on the states of $\mathcal{A}$ and $\mathcal{B}$, and a symbolic version, defined on $\mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$ and using time durations taken from $Times(\_, \_)$. Analogues of these notions, and their associated results, can be found in [12, 13].

Let $\mathcal{R} \subseteq S_{\mathcal{A}} \times S_{\mathcal{B}}$, and let $(s_{\mathcal{A}}, s_{\mathcal{B}}) \in S_{\mathcal{A}} \times S_{\mathcal{B}}$. Then $(s_{\mathcal{A}}, s_{\mathcal{B}})$ *is concretely good in* $\mathcal{R}$ if, for each $(s_{\mathcal{A}}, d, a, \mu_{\mathcal{A}}) \in \rightarrow_{\mathcal{A}}$, there exists $(s_{\mathcal{B}}, d, a, \mu_{\mathcal{B}}) \in \rightarrow_{\mathcal{B}}$ such that $\mu_{\mathcal{A}} \mathcal{R} \mu_{\mathcal{B}}$. The following lemma states that concrete goodness, with respect to a relation described as a union of regions, is invariant over regions.

**Lemma 1.** *Let* $\Gamma \subseteq \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$*, and let* $R \in \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$ *be such that there exists* $(s_{\mathcal{A}}, s_{\mathcal{B}}) \in R$ *which is concretely good in* $\bigcup_{R' \in \Gamma} R'$*. Then* each $(s'_{\mathcal{A}}, s'_{\mathcal{B}}) \in R$ *is concretely good in* $\bigcup_{R' \in \Gamma} R'$*.*

Given a relation $\mathcal{R} \subseteq S_{\mathcal{A}} \times S_{\mathcal{B}}$, we let $\Gamma_{\mathcal{R}} = \{R \in \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}} \mid R \cap \mathcal{R} \neq \emptyset\}$.

**Proposition 1.** *If* $\mathcal{R} \subseteq S_{\mathcal{A}} \times S_{\mathcal{B}}$ *is a timed simulation, then* $\bigcup_{R' \in \Gamma_{\mathcal{R}}} R'$ *is a timed simulation.*

Let $R \in \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$. Let $X_{\mathcal{A}} \subseteq \mathcal{X}_{\mathcal{A}}$, $X_{\mathcal{B}} \subseteq \mathcal{X}_{\mathcal{B}}$, $l_{\mathcal{A}} \in L_{\mathcal{A}}$ and $l_{\mathcal{B}} \in L_{\mathcal{B}}$. Then we write $R[X_{\mathcal{A}} \cup X_{\mathcal{B}} := 0, loc := l_{\mathcal{A}}, l_{\mathcal{B}}]$ to denote the region which has the location components $l_{\mathcal{A}}$ and $l_{\mathcal{B}}$, and the clock equivalence class equal to $R$ except that the clocks in $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ are reset to 0. Now let $\Gamma \subseteq \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$. Let $\mathsf{p}_{\mathcal{A}} \in \mathsf{Dist}(2^{\mathcal{X}_{\mathcal{A}}} \times L_{\mathcal{A}})$ and $\mathsf{p}_{\mathcal{B}} \in \mathsf{Dist}(2^{\mathcal{X}_{\mathcal{B}}} \times L_{\mathcal{B}})$. Then $\mathcal{R}_{R,\Gamma} \subseteq \mathsf{support}(\mathsf{p}_{\mathcal{A}}) \times \mathsf{support}(\mathsf{p}_{\mathcal{B}})$ is defined as follows: for each $(X_{\mathcal{A}}, l_{\mathcal{A}}) \in \mathsf{support}(\mathsf{p}_{\mathcal{A}})$ and $(X_{\mathcal{B}}, l_{\mathcal{B}}) \in \mathsf{support}(\mathsf{p}_{\mathcal{B}})$, we have $(X_{\mathcal{A}}, l_{\mathcal{A}}) \mathcal{R}_{R,\Gamma} (X_{\mathcal{B}}, l_{\mathcal{B}})$ if and only if $R[X_{\mathcal{A}} \cup X_{\mathcal{B}} := 0, loc := l_{\mathcal{A}}, l_{\mathcal{B}}] \in \Gamma$.

The region $R$ *is symbolically good in* $\Gamma$ if there exists $(s_{\mathcal{A}}, s_{\mathcal{B}}) \in R$ such that, for each $(s_{\mathcal{A}}, d, a, \mathsf{p}_{\mathcal{A}}) \in \rightarrow_{\mathcal{A}}$ with $d \in Times(s_{\mathcal{A}}, s_{\mathcal{B}})$, there exists a transition $(s_{\mathcal{B}}, d, a, \mathsf{p}_{\mathcal{B}}) \in \rightarrow_{\mathcal{B}}$ such that $\mathsf{p}_{\mathcal{A}} \mathcal{R}_{R',\Gamma} \mathsf{p}_{\mathcal{B}}$, where $R' = [s_{\mathcal{A}} + d, s_{\mathcal{B}} + d]$. The following lemma establishes a connection between symbolic and concrete goodness.

**Lemma 2.** *Let $\Gamma \subseteq \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$, and let $R \in \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$ be symbolically good in $\Gamma$. Then each $(s_\mathcal{A}, s_\mathcal{B}) \in R$ is concretely good in $\bigcup_{R' \in \Gamma} R'$.*

The proof of Lemma 2 relies on first showing that a state pair $(s_\mathcal{A}, s_\mathcal{B})$ in $R$ which witnesses symbolic goodness in $\Gamma$ is also concretely good in $\bigcup_{R' \in \Gamma} R'$, which then, by Lemma 1, implies concrete goodness in $\bigcup_{R' \in \Gamma} R'$ for all state pairs in $R$.

**Algorithm.** Let $\Omega : 2^{\mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}} \to 2^{\mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}}$ be the monotone operator defined by $\Omega(\Gamma) = \{R \in \Gamma \mid R \text{ is symbolically good in } \Gamma\}$. By Lemma 2, to decide whether a region $R \in \Gamma$ is such that $R \in \Omega(\Gamma)$, the choice of which representative state pair to consider for $R$ is not significant: hence, an arbitrary state pair can be considered. Note also that $|\mathit{Times}(s_\mathcal{A}, s_\mathcal{B})|$, for any $(s_\mathcal{A}, s_\mathcal{B}) \in S_\mathcal{A} \times S_\mathcal{B}$, is exponential in the sizes of $\mathcal{A}$ and $\mathcal{B}$, as is $|\mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}|$. By the results of [14], we have that, for any $\mathsf{p}_\mathcal{A} \in \mathsf{Dist}(2^{\mathcal{X}_\mathcal{A}} \times L_\mathcal{A}), \mathsf{p}_\mathcal{B} \in \mathsf{Dist}(2^{\mathcal{X}_\mathcal{B}} \times L_\mathcal{B})$, it is possible to decide $\mathsf{p}_\mathcal{A} \mathcal{R}_{R,\Gamma} \mathsf{p}_\mathcal{B}$ in polynomial time. Hence, we can compute $\Omega(\Gamma)$ in exponential time in the sizes of $\mathcal{A}$ and $\mathcal{B}$.

**Lemma 3.** *If $\mathcal{R} \subseteq S_\mathcal{A} \times S_\mathcal{B}$ is a timed simulation, then $\Gamma_\mathcal{R}$ is a fixpoint of $\Omega$.*

**Proposition 2.** *Let $\Gamma \subseteq \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$ be a set of regions. Then $\Gamma$ is a fixpoint of $\Omega$ if and only if $\bigcup_{R \in \Gamma} R$ is a timed simulation.*

The operator $\Omega$ provides the basis of the algorithm for deciding whether $\mathcal{A} \preceq \mathcal{B}$. Our aim is to compute its greatest fixpoint $\Gamma_{max}$. Let $\Gamma_0 = \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$, and let $\Gamma_{i+1} = \Omega(\Gamma_i)$ for each $i \geq 0$. From the monotonicity of $\Omega$, for some $i \leq |\mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}|$, we have $\Gamma_i = \Omega(\Gamma_i)$. Hence it suffices to apply $\Omega$ at most $|\mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}|$ times, and therefore $\Gamma_{max}$ can be computed in exponential time in the sizes of $\mathcal{A}$ and $\mathcal{B}$.

Let $\mathcal{R}_{max}$ be the maximal timed simulation relation from $\mathcal{A}$ to $\mathcal{B}$. By Proposition 1, we have that $\mathcal{R}_{max}$ is a union of regions. Given the computation of $\Gamma_{max}$, by Proposition 2 we have that $\mathcal{R}_{max} = \bigcup_{R \in \Gamma_{max}} R$. We then have that the following are equivalent:

- $\mathcal{A} \preceq \mathcal{B}$;
- for every $l_\mathcal{A} \in \bar{L}_\mathcal{A}$, there exists some $l_\mathcal{B} \in \bar{L}_\mathcal{B}$ such that $(l_\mathcal{A}, \mathbf{0}) \mathcal{R}_{max} (l_\mathcal{B}, \mathbf{0})$;
- for every $l_\mathcal{A} \in \bar{L}_\mathcal{A}$, there exists some $l_\mathcal{B} \in \bar{L}_\mathcal{B}$ such that $[((l_\mathcal{A}, l_\mathcal{B}), \mathbf{0})] \in \Gamma_{max}$.

We can adapt the algorithm above to obtain an algorithm for deciding whether two PTA $\mathcal{A}$ and $\mathcal{B}$ are timed bisimilar. First, we note that concrete and symbolic goodness are required to be redefined to obtain *symmetric* versions; furthermore, concrete goodness is defined with respect to an equivalence relation $\mathcal{R}$, and symbolic goodness is defined with respect to a set $\Gamma$ of regions which induces an equivalence relation (that is, $\bigcup_{R \in \Gamma} R$ is an equivalence relation). Then it is possible to define a version of the operator $\Omega$ which makes reference to the new, symmetric notion of symbolic goodness.

From the results of [24], we have that deciding timed simulation or timed bisimulation is EXPTIME-hard. In combination with the above, this gives us the following theorem.

**Theorem 1.** *Given two PTA $\mathcal{A}$ and $\mathcal{B}$, the following two problems are EXPTIME-complete: (1) checking whether $\mathcal{A} \preceq \mathcal{B}$; (2) checking whether $\mathcal{A} \approx \mathcal{B}$.*

**Probabilistic Timed Simulation.** The results of the previous subsection can be adapted to the case of probabilistic timed simulation and bisimulation. As in the previous section, we can obtain an algorithm for probabilistic timed bisimulation from an algorithm for probabilistic timed simulation, and hence we consider the latter. Formally, for two PTA $\mathcal{A}$ and $\mathcal{B}$, our aim is to decide whether $\mathcal{A} \preceq^p \mathcal{B}$.

Firstly, we extend the notions of concrete and symbolic goodness to accommodate the possibility of $\mathcal{B}$ choosing combined transitions. For concrete goodness, this is done simply by replacing the condition $(s_{\mathcal{B}}, d, a, \mu_{\mathcal{B}}) \in \rightarrow_{\mathcal{B}}$ with $(s_{\mathcal{B}}, d, a, \mu_{\mathcal{B}}) \in \mathsf{Combined}(s_{\mathcal{B}}, d, a)$. For symbolic goodness, we first apply the notion of combined transition to the case of distributions featured in probabilistic edges: given the largest set $\{\mathsf{p}_1, ..., \mathsf{p}_k\}$ of distributions such that $(s_{\mathcal{B}}, d, a, \mathsf{p}_i) \in \rightarrow_{\mathcal{B}}$ for $1 \leq i \leq k$, we then write $\mathsf{Combined}^p(s_{\mathcal{B}}, d, a)$ for the set of all tuples $(s_{\mathcal{B}}, d, a, \mathsf{p})$ such that there exists a sequence $c_1, ..., c_k$ of real numbers in $[0, 1]$ with $\sum_{1 \leq i \leq k} c_i = 1$ and $\mathsf{p} = \sum_{1 \leq i \leq k} c_i \mathsf{p}_i$. Then, to obtain the new notion of symbolic goodness, we replace the condition $(s_{\mathcal{B}}, d, a, \mathsf{p}_{\mathcal{B}}) \in \rightarrow_{\mathcal{B}}$ with $(s_{\mathcal{B}}, d, a, \mathsf{p}_{\mathcal{B}}) \in \mathsf{Combined}^p(s_{\mathcal{B}}, d, a)$. Then the operator $\Omega$ is adapted to take into account the new notion of symbolic goodness. Using the results of [15], for a given $\Gamma \subseteq \mathsf{Regions}_{\mathcal{A}\|\mathcal{B}}$, it is possible to compute $\Omega(\Gamma)$ in exponential time in the sizes of $\mathcal{A}$ and $\mathcal{B}$. This reasoning, combined with that concerning the exponential number of iterations of $\Omega$ given for timed simulation, then can be used to obtain the following result.

**Theorem 2.** *Given two PTA $\mathcal{A}$ and $\mathcal{B}$, the following two problems are EXPTIME-complete: (1) checking whether $\mathcal{A} \preceq^p \mathcal{B}$; (2) checking whether $\mathcal{A} \approx^p \mathcal{B}$.*

## 4 Logical Characterization of Bisimulation

In this section we give a logical characterization of our timed bisimulation and probabilistic timed bisimulation relations. Recall that [18] presents an extension of Hennessy-Milner logic [17] for probabilistic automata. The principal novelty of the logic of [18] is that its semantics is defined over distributions on states, rather than over states. Here we extend the logic of [18] with constraints on the duration of transitions, similarly to [16, 13].

We now present the syntax of the logic. The logic PTLogic is syntactically defined by the following formulas:

$$\psi ::= true \mid \neg\psi \mid \psi \wedge \psi \mid \langle a, \sim c\rangle\psi \mid [\psi]_p$$

where $a \in Act$ is an action, $c \in \mathbb{R}_{\geq 0}$ is a constant, and $p \in [0, 1]$ is a probability. Note that we will discuss the sub-logic of PTLogic in which $c \in \mathbb{Q}_{\geq 0}$ (where $\mathbb{Q}_{\geq 0}$ denotes the set of non-negative rationals) at the end of this section.

Let $\mathsf{P}$ be a PTLTS. Given a distribution $\mu \in \mathsf{Dist}(S)$ and a set $S' \subseteq S$ of states, we let $\mu(S') = \sum_{s \in S'} \mu(s)$. Let $\psi$ be a formula in PTLogic and $\mu$ be

a distribution over the set of states of a PTLTS $\mathsf{P}$. We say that $\mu$ *satisfies* $\psi$, written $\mu \models \psi$, according to the following:

$$
\begin{array}{lll}
\mu \models \mathit{true} & & \\
\mu \models \neg\psi & \text{iff} & \mu \not\models \psi \\
\mu \models \psi_1 \wedge \psi_2 & \text{iff} & \text{both } \mu \models \psi_1 \text{ and } \mu \models \psi_2 \\
\mu \models \langle a, \sim c\rangle\psi & \text{iff} & \text{for all } s \in \mathsf{support}(\mu) \text{ there exists } (s, d, a, \mu') \in \rightarrow \text{ such that} \\
 & & \quad d \sim c \text{ and } \mu' \models \psi \\
\mu \models [\psi]_p & \text{iff} & \mu(\{\!|\psi|\!\}) \geq p
\end{array}
$$

where $\{\!|\psi|\!\} = \{s \in S \mid s \models \psi\}$ denotes the set of all states of $\mathsf{P}$ that satisfy the PTLogic formula $\psi$, and where $s \models \psi$ if and only if $\{s \mapsto 1\} \models \psi$.

We now show that timed bisimilar states of the semantic PTLTS $[\![\mathcal{A}]\!] = (S, \bar{S}, \mathit{Act}, \rightarrow)$ resulting from a PTA $\mathcal{A}$ satisfy the same formulas of PTLogic and, conversely, if there exists a formula of PTLogic that is satisfied in one state and not another, then these two states are not timed bisimilar. We introduce the following notation. Let $\mathcal{F}$ be the set of all PTLogic formulas. Given a set $\mathcal{F}' \subseteq \mathcal{F}$ of PTLogic formulas, we use $\mathcal{F}'(s)$ and $\mathcal{F}'(\mu)$ to denote the subset of formulas of $\mathcal{F}'$ that are satisfied at state $s \in S$ and by distribution $\mu \in \mathsf{Dist}(S)$, respectively. The *depth* of a PTLogic formula $\psi$ is defined as the maximum number of nested $\langle a, \sim c\rangle\psi'$ operators that occur in $\psi$. Let $\mathcal{F}_n$ be the set of PTLogic formulas of depth $n$, and let $\bowtie_n \subseteq S \times S$ be the relation such that $s \bowtie_n s'$ if and only if $\mathcal{F}_n(s) = \mathcal{F}_n(s')$. Then, as in [18], we have the following results.

**Lemma 4.** *1. For each pair $s, s' \in S$ of states, if $\mathcal{F}(s) \neq \mathcal{F}(s')$ then $\mathcal{F}(s) \not\subseteq \mathcal{F}(s')$.*
*2. For each pair $s, s' \in S$ of states, $\mathcal{F}_0(s) = \mathcal{F}_0(s')$.*
*3. Let $\mathcal{R} \subseteq \bowtie_n$ for some $n \in \mathbb{N}$. Then, for each pair $\mu, \mu' \in \mathsf{Dist}(S)$, we have that $\mu \mathcal{R} \mu'$ implies $\mathcal{F}_n(\mu) = \mathcal{F}_n(\mu')$.*

The first two points of Lemma 4 follow from the definitions in a straightforward manner. The third point can be shown in a manner similar to the analogous result of [18].

Let $\approx_0 = S \times S$ (that is, the relation $\approx_0$ relates all states). For $n \in \mathbb{N}$, let $\approx_{n+1} \subseteq S \times S$ be the equivalence relation defined as follows: for each $s, s' \in S$, $s \approx_{n+1} s'$ implies that, for each $(s, d, a, \mu) \in \rightarrow$, there exists $(s', d, a, \mu') \in \rightarrow$ such that $\mu \approx_n \mu'$. On semantic PTLTS of PTAs, we have that $\approx = \bigcap_{n \in \mathbb{N}} \approx_n$.

**Theorem 3.** *Let $[\![\mathcal{A}]\!] = (S, \bar{S}, \mathit{Act}, \rightarrow)$ be the semantic PTLTS of the PTA $\mathcal{A}$. For each pair $s, s' \in S$ of states, we have $s \approx s'$ if and only if $\mathcal{F}(s) = \mathcal{F}(s')$.*

*Proof.* The proof proceeds along the same lines as that of Theorem 1 of [18]; for completeness, we present the overall structure of the proof. We proceed by induction on $n \in \mathbb{N}$, and show that $s \approx_n s'$ if and only if $\mathcal{F}_n(s) = \mathcal{F}_n(s')$. The base case follows from point 2 of Lemma 4 and the definition of $\approx_0$. We now consider both directions of the inductive step.

($\Rightarrow$) Let $s \approx_{n+1} s'$. We require that $\mathcal{F}_{n+1}(s) = \mathcal{F}_{n+1}(s')$, which requires showing that, for all $\psi \in \mathcal{F}_{n+1}$, we have $s \models \psi$ if and only if $s' \models \psi$. The

cases of the Boolean combinators and probabilistic operator $[\psi]_p$ are similar to the analogous cases in [18]. Consider the case of $\psi = \langle a, \sim c\rangle\phi$. Then, by the semantics of PTLogic, there exists $(s, d, a, \mu) \in\!\!\rightarrow$ such that $d \sim c$ and $\mu \models \phi$. From $s \approx_{n+1} s'$, there exists $(s', d, a, \mu') \in\!\!\rightarrow$ such that $\mu \approx_n \mu'$. From $\mu \approx_n \mu'$ and point 3 of Lemma 4, we have that $\mathcal{F}_n(\mu) = \mathcal{F}_n(\mu')$. Noting that $\phi \in \mathcal{F}_n$, then from $\mu \models \phi$ we have $\mu' \models \phi$. From this fact, and the observation that $d \sim c$, we have that $s' \models \langle a, \sim c\rangle\phi$.

($\Leftarrow$) We proceed by showing that $s \not\approx_{n+1} s'$ implies $\mathcal{F}_{n+1}(s) \neq \mathcal{F}_{n+1}(s')$. Let $\{[t_i]_n\}_{i\in I}$ be an enumeration of the equivalence classes of $\approx_n$ (there will be a finite number of such classes by the results of Section 3, in contrast to possibly countably infinite number in [18]). For each $i \in I$, by induction and point 1 of Lemma 4, we can construct a formula $\phi_i$ which is satisfied only by states in $[t_i]_n$. We then select some $(s, d, a, \mu) \in\!\!\rightarrow$ such that there does not exist any $(s', d, a, \mu') \in\!\!\rightarrow$ for which $\mu \approx_n \mu'$. Such $(s, d, a, \mu)$ exists because $s \not\approx_{n+1} s'$. Let $\phi = \bigwedge_{i\in I}[\phi_i]_{\mu([t_i]_n)}$. Clearly $\mu \models \phi$, and hence $s \models \langle a, = d\rangle\phi$. Aiming for a contradiction, assume that $\mathcal{F}_{n+1}(s) = \mathcal{F}_{n+1}(s')$. Then $s' \models \langle a, = d\rangle\phi$. This implies the existence of $(s', d, a, \mu'') \in\!\!\rightarrow$ such that $\mu'' \models \phi$. This is turn implies that $\mu''([t_i]_n) = \mu([t_i]_n)$ for each $i \in I$, which implies that $\mu \approx_n \mu''$, contradicting $s \not\approx_{n+1} s'$. □

**Probabilistic Timed Bisimulation.** As in [18], the above material can be adapted to the case of probabilistic timed bisimulation in the following way. First we replace the operator $\langle a, \sim c\rangle\psi$ in PTLogic with the operator $\langle a, \sim c\rangle\psi$, which has the following semantics: given a distribution $\mu$, we have $\mu \models \langle a, \sim c\rangle\psi$ if and only if for all $s \in \mathsf{support}(\mu)$ there exists $(s, d, a, \mu') \in \mathsf{Combined}(s, d, a)$ such that $d \sim c$ and $\mu' \models \psi$. Let $\mathcal{F}^\bullet$ denote the set of formulas of the resulting logic. The proof of Theorem 3 can be adapted to the new logic by changing references to transitions to references to combined transitions as necessary, because timing issues are independent of issues concerning combined transitions. This leads to the following result.

**Theorem 4.** *Let $[\![\mathcal{A}]\!] = (S, \bar{S}, Act, \rightarrow)$ be the semantic PTLTS of the PTA $\mathcal{A}$. For each pair $s, s' \in S$ of states, we have $s \approx^p s'$ if and only if $\mathcal{F}^\bullet(s) = \mathcal{F}^\bullet(s')$.*

**Restriction to Rational Timing Bounds.** The logic PTLogic features real values in constraints on timing bounds in order to provide a logical characterization of timed bisimulation for all states of a PTA. However, inspired by [13], we note that a version of PTLogic restricted to non-negative rationals $\mathbb{Q}_{\geq 0}$ provides a logical characterization of timed bisimulation for those states of a PTA with rational values of clocks. Let $\mathcal{F}_{\mathbb{Q}_{\geq 0}}$ denote the set of formulas of the logic obtained from PTLogic by restricting formulas of $\langle a, \sim c\rangle\psi$ to the case of $c \in \mathbb{Q}_{\geq 0}$.

**Theorem 5.** *Let $[\![\mathcal{A}]\!] = (S, \bar{S}, Act, \rightarrow)$ be the semantic PTLTS of the PTA $\mathcal{A}$ with the set $\mathcal{X}$ of clocks. For each pair $(l, v), (l', v') \in S$ of states such that $v(x) \in \mathbb{Q}_{\geq 0}$ and $v'(x) \in \mathbb{Q}_{\geq 0}$ for all clocks $x \in \mathcal{X}$, $\mathcal{F}_{\mathbb{Q}_{\geq 0}}(s) = \mathcal{F}_{\mathbb{Q}_{\geq 0}}(s')$ implies $(l, v) \approx (l', v')$.*

The proof of Theorem 5 follows that of direction ($\Leftarrow$) of Theorem 3, except that, as in [13], and without loss of generality, only transitions with durations taken from $Times(s, s')$ are considered.

The converse of Theorem 5 (that is, that $(l, v) \approx (l', v')$ implies $\mathcal{F}_{\mathbb{Q}_{\geq 0}}(s) = \mathcal{F}_{\mathbb{Q}_{\geq 0}}(s')$) follows trivially from Theorem 3, because $\mathcal{F}_{\mathbb{Q}_{\geq 0}} \subseteq \mathcal{F}$. Theorem 5 can also be extended to the case of probabilistic timed bisimulation.

Note that, to decide $\mathcal{A} \approx \mathcal{B}$, we consider whether the initial states of the PTA are related by $\approx$; as all clocks have to value 0 initially, clearly the above PTLogic with time constraints restricted to $\mathbb{Q}_{\geq 0}$ characterizes bisimulation between PTA. Finally, we observe that formulas of PTLogic with time constraints restricted to $\mathbb{Q}_{\geq 0}$ can be expressed in the timed modal logic of [25, 26] extended with the probabilistic operator of $[\psi]_p$. Hence, such a logic can also provide a logical characterization of states with rational clock values.

## 5 Conclusions

In this paper we have presented a framework for reasoning about simulation and bisimulation relations for PTA. On the one hand, we have presented an EXPTIME algorithm for deciding such relations, and on the other hand we have shown how a timed extension of the probabilistic model logic of [18] provides a logical characterization of bisimulation. To our knowledge a logical characterization of simulation for Segala's probabilistic automata does not yet exist: if such a characterization is found, it is likely that it can be adapted also to the case of PTA. For specifying properties of probabilistic timed automata, temporal logics such as PTCTL [5], which include constraints on time and probability, have been introduced: we note that timed bisimulation preserves PTCTL properties, and that, for a negation-free fragment of PTCTL, a state $s$ that is timed simulated by another state $s'$ satisfies at least the same properties as $s'$ [3, 4, 27].

For future work, we intend to study weak extensions of the considered relations, which abstract from non-observable computation (see [28]), and to develop quantitative versions of simulation and bisimulation for PTA, which can quantify how closely two PTA resemble each other.

## References


1. Alur, R., Dill, D.L.: A theory of timed automata. TCS **126**(2) (1994) 183–235
2. Puterman, M.L.: Markov Decision Processes. J. Wiley & Sons (1994)
3. Segala, R., Lynch, N.A.: Probabilistic simulations for probabilistic processes. Nordic Journal of Computing **2**(2) (1995) 250–273
4. Segala, R.: Modeling and Verification of Randomized Distributed Real-Time Systems. PhD thesis, Massachusetts Institute of Technology (1995)
5. Kwiatkowska, M., Norman, G., Segala, R., Sproston, J.: Automatic verification of real-time systems with discrete probability distributions. TCS **286** (2002) 101–150
6. Kwiatkowska, M., Norman, G., Sproston, J.: Probabilistic model checking of deadline properties in the IEEE 1394 FireWire root contention protocol. Formal Aspects of Computing **14**(3) (2003) 295–318

7. Kwiatkowska, M., Norman, G., Parker, D., Sproston, J.: Performance analysis of probabilistic timed automata using digital clocks. FMSD **29** (2006) 33–78
8. Milner, R.: An algebraic definition of simulation between programs. In: Proc. IJCAI'71, William Kaufmann (1971) 481–489
9. Milner, R.: A calculus of communicating systems. Volume 92 of LNCS. Springer (1980)
10. Park, D.: Concurrency and automata on infinite sequences. In: Proc. 5th GI-Conference on Theoretical Computer Science. Volume 104 of LNCS, Springer (1981) 167–183
11. Čerāns, K.: Decidability of bisimulation equivalences for parallel timer processes. In: Proc. CAV'92. Volume 663 of LNCS, Springer (1992) 302–315
12. Taşıran, S., Alur, R., Kurshan, R.P., Brayton, R.K.: Verifying abstractions of timed systems. In: Proc. CONCUR'96. Volume 1119 of LNCS, Springer (1996) 546–562
13. Bozzelli, L., Legay, A., Pinchinat, S.: On timed alternating simulation for concurrent timed games. In: Proc. FSTTCS'09. Volume 4 of LIPIcs., Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2009) 85–96
14. Baier, C., Engelen, B., Majster-Cederbaum, M.E.: Deciding bisimilarity and similarity for probabilistic processes. JCSS **60**(1) (2000) 187–231
15. Zhang, L., Hermanns, H., Eisenbrand, F., Jansen, D.N.: Flow faster: Efficient decision algorithms for probabilistic simulations. LMCS **4**(4) (2008)
16. Holmer, U., Larsen, K.G., Yi, W.: Deciding properties of regular real time processes. In: Proc. CAV'91. Volume 575 of LNCS, Springer (1991) 443–453
17. Hennessy, M., Milner, R.: Algebraic laws for nondeterminism and concurrency. JACM **32**(1) (1985) 137–161
18. Parma, A., Segala, R.: Logical characterizations of bisimulations for discrete probabilistic systems. In: Proc. FOSSACS'07. Volume 4423 of LNCS, Springer (2007) 287–301
19. Jensen, H.E., Gregersen, H.: Formal design of reliable real time systems. Master's thesis, Aalborg University (1995)
20. Jensen, H.E.: Model checking probabilistic real time systems. In: Proc. of the 7th Nordic Work. on Progr. Theory, Chalmers Institute of Technology (1996) 247–261
21. Yamane, S.: Probabilistic timed simulation verification and its application to stepwise refinement of real-time systems. In: Proc. ASIAN'03. Volume 2896 of LNCS, Springer (2003) 276–290
22. Chen, T., Han, T., Katoen, J.P.: Time-abstracting bisimulation for probabilistic timed automata. In: Proc. TASE'08, IEEE (2008) 177–184
23. Jonsson, B., Larsen, K.G.: Specification and refinement of probabilistic processes. In: Proc. LICS'91, IEEE (1991) 266–277
24. Laroussinie, F., Schnoebelen, P.: The state explosion problem from trace to bisimulation equivalence. In: Proc. FOSSACS'00. Volume 1784 of LNCS, Springer (2000) 192–207
25. Laroussinie, F., Larsen, K.G., Weise, C.: From timed automata to logic – and back. In: Proc. MFCS'95. Volume 969 of LNCS, Springer (1995) 529–539
26. Aceto, L., Laroussinie, F.: Is your model checker on time? On the complexity of model checking for timed modal logics. JLAP **52-53** (2000) 7–51
27. Sproston, J.: Model checking for probabilistic timed and hybrid systems. PhD thesis, University of Birmingham (2000)
28. Lanotte, R., Maggiolo-Schettini, A., Troina, A.: Weak bisimulation for probabilistic timed automata and applications to security. In: Proc. SEFM'03, IEEE (2003) 34–43