# Verifying Probabilistic Timed Automata Against Omega-Regular Dense-Time Properties

Hongfei Fu[1], Yi Li[2], and Jianlin Li(✉)[3,4,5]

[1] Shanghai Jiao Tong University, Shanghai, China
fuhf@cs.sjtu.edu.cn
[2] Department of Informatics, School of Mathematical Sciences, Peking University, Beijing, China
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[4] University of Chinese Academy of Sciences, Beijing, China
[5] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
ljlin@nuaa.edu.cn

**Abstract.** Probabilistic timed automata (PTAs) are timed automata (TAs) extended with discrete probability distributions. They serve as a mathematical model for a wide range of applications that involve both stochastic and timed behaviours. In this work, we consider the problem of model-checking linear *dense-time* properties over PTAs. In particular, we study linear dense-time properties that can be encoded by TAs with infinite acceptance criterion. First, we show that the problem of model-checking PTAs against deterministic-TA specifications can be solved through a product construction. Based on the product construction, we prove that the computational complexity of the problem with deterministic-TA specifications is EXPTIME-complete. Then we show that when relaxed to general (nondeterministic) TAs, the model-checking problem becomes undecidable. Our results substantially extend state of the art with both the dense-time feature and the nondeterminism in TAs.

## 1 Introduction

Stochastic timed systems are systems that exhibit both timed and stochastic behaviours. Such systems play a dominant role in many applications [1], hence addressing fundamental issues such as safety and performance over these systems are important. *Probabilistic timed automata* (PTAs) [2, 3, 4] serve as a good mathematical model for these systems. They extend the well-known model of timed automata [5] (for nonprobabilistic timed systems) with discrete probability distributions, and Markov Decision Processes (MDPs) [6] (for untimed probabilistic systems) with timing constraints.

Formal verification of PTAs has received much attention in recent years [2]. For branching-time model-checking of PTAs, the problem is reduced to computation of reachability probabilities over MDPs through well-known finite abstraction for timed automata (namely *regions* and *zones*) [7, 3, 4]. Advanced techniques for branching-time model checking of PTAs such as inverse method and symbolic method have been further explored in [8, 9, 10, 11]. Extension with *cost* or *reward*, resulting in *priced* PTAs, has also been well investigated. Jurdzinski *et al.* [12] and Kwiatkowska *et al.* [13] proved that several notions of

accumulated or discounted cost are computable over priced PTAs, while cost-bounded reachability probability over priced PTAs is shown to be undecidable by Berendsen *et al.* [14]. Most verification algorithms for PTAs have been implemented in the model checker PRISM [15]. Computational complexity of several verification problems for PTAs has been studied, for example, [16, 17, 12].

For linear-time model-checking, much less is known. As far as we know, the only relevant result is by Sproston [18] who proved that the problem of model-checking PTAs against linear *discrete-time* properties encoded by *untimed* deterministic omega-regular automata (e.g., Rabin automata) can be solved by a product construction. In his paper, Sproston first devised a production construction that produces a PTA out of the input PTA and the automaton; then he proved that the problem can be reduced to omega-regular verification of MDPs through maximal end components.

In this paper, we study the problem of model-checking linear *dense-time* properties over PTAs. Compared with discrete-time properties, dense-time properties take into account timing constraints, and therefore is more expressive and applicable to time-critical systems. Simultaneously, verification of dense-time properties is more challenging since it requires to involve timing constraints. The extra feature of timing constraints also brings more theoretical difficulty, e.g., timed automata [5] (TAs) are generally not determinizable, which is in contrast to untimed automata (such as Rabin or Muller automata).

We focus on linear dense-time properties that can be encoded by TAs. Due to the ability to model dense-time behaviours, TAs can be used to model real-time systems, while they can also act as language recognizers for timed omega-regular languages. Here we treat TAs as language recognizers for timed paths from a PTA, and study the problem of computing the minimum or maximum probability that a timed path from the PTA is accepted by the TA. The intuition is that a TA can recognize the set of "good" (or "bad") timed paths emitting from a PTA, so the problem is to compute the probability that the PTA behaves in a good (or bad) manner. The relationship between TAs and linear temporal logic (e.g., Metric Temporal Logic [19]) is studied in [20, 21].

*Our Contributions.* We distinguish between the subclass of *deterministic* TAs (DTAs) and general *nondeterministic* TAs. DTAs are the deterministic subclass of TAs. Although the class of DTAs is weaker than general timed automata, it can recognize a wide class of formal timed languages, and express interesting linear dense-time properties which cannot be expressed in branching-time logics (cf. [22]). We consider Rabin acceptance condition as the infinite acceptance criterion for TAs. We first show that the problem of model-checking PTAs against DTA specifications with Rabin acceptance condition can be solved through a nontrivial product construction which tackles the integrated feature of timing constraints and randomness. From the product construction, we further prove that the problem is EXPTIME-complete. Then we show that the problem becomes undecidable when one considers general TAs. Our results substantially extend previous ones (e.g. [18]) with both the dense-time feature and the nondeterminism in TAs.

Due to lack of space, detailed proofs of several results and some experimental results are put in the full version [23].

## 2 Preliminaries

We denote by $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Z}$, and $\mathbb{R}$ the sets of all positive integers, non-negative integers, integers and real numbers, respectively. For any infinite word $w = b_0 b_1 \dots$ over an alphabet $\Sigma$, we denote by $\mathit{inf}(w)$ the set of symbols in $\Sigma$ that occur infinitely often in $w$. A *clock* is a variable for a nonnegative real number. Below we fix a finite set $\mathcal{X}$ of clocks.

*Clock Valuations.* A *clock valuation* is a function $\nu : \mathcal{X} \to [0, \infty)$. The set of clock valuations is denoted by $\mathit{Val}(\mathcal{X})$. Given a clock valuation $\nu$, a subset $X \subseteq \mathcal{X}$ of clocks and a non-negative real number $t$, we let (i) $\nu[X := 0]$ be the clock valuation such that $\nu[X := 0](x) = 0$ for $x \in X$ and $\nu[X := 0](x) = \nu(x)$ otherwise, and (ii) $\nu{+}t$ be the clock valuation such that $(\nu{+}t)(x) = \nu(x) + t$ for all $x \in \mathcal{X}$. We denote by $\mathbf{0}$ the clock valuation such that $\mathbf{0}(x) = 0$ for $x \in \mathcal{X}$.

*Clock Constraints.* The set $CC(\mathcal{X})$ of *clock constraints* over $\mathcal{X}$ is generated by the following grammar: $\phi := \mathbf{true} \mid x \leq d \mid c \leq x \mid x{+}c \leq y{+}d \mid \neg\phi \mid \phi \wedge \phi$ where $x, y \in \mathcal{X}$ and $c, d \in \mathbb{N}_0$. We write **false** for a short hand of $\neg\mathbf{true}$. The satisfaction relation $\models$ between valuations $\nu$ and clock constraints $\phi$ is defined through substituting every $x \in \mathcal{X}$ appearing in $\phi$ by $\nu(x)$ and standard semantics for logical connectives. For a given clock constraint $\phi$, we denote by $[\![\phi]\!]$ the set of all clock valuations that satisfy $\phi$.

### 2.1 Probabilistic Timed Automata

A *discrete probability distribution* over a countable non-empty set $U$ is a function $q : U \to [0, 1]$ such that $\sum_{z \in U} q(z) = 1$. The *support* of $q$ is defined as $\mathit{supp}(q) := \{z \in U \mid q(z) > 0\}$. We denote the set of discrete probability distributions over $U$ by $\mathcal{D}(U)$.

**Definition 1 (Probabilistic Timed Automata [2]).** *A* probabilistic timed automaton *(PTA)* $\mathcal{C}$ *is a tuple*

$$\mathcal{C} = (L, \ell^*, \mathcal{X}, \mathit{Act}, \mathit{inv}, \mathit{enab}, \mathit{prob}, \mathit{AP}, \mathcal{L}) \tag{1}$$

*where :*

- $L$ *is a finite set of* locations*;*
- $\ell^* \in L$ *is the* initial *location;*
- $\mathcal{X}$ *is a finite set of* clocks*;*
- $\mathit{Act}$ *is a finite set of* actions*;*
- $\mathit{inv} : L \to CC(\mathcal{X})$ *is an* invariant condition*;*
- $\mathit{enab} : L \times \mathit{Act} \to CC(\mathcal{X})$ *is an* enabling condition*;*
- $\mathit{prob} : L \times \mathit{Act} \to \mathcal{D}\big(2^{\mathcal{X}} \times L\big)$ *is a* probabilistic transition function*;*
- $\mathit{AP}$ *is a finite set of* atomic propositions*;*
- $\mathcal{L} : L \to 2^{\mathit{AP}}$ *is a* labelling function*.*

W.l.o.g, we consider that both $\mathit{Act}$ and $\mathit{AP}$ is disjoint from $[0, \infty)$. Below we fix a PTA $\mathcal{C}$. The semantics of PTAs is as follows.

*States and Transition Relation.* A *state* of $\mathcal{C}$ is a pair $(\ell, \nu)$ in $L \times \mathit{Val}(\mathcal{X})$ such that $\nu \models \mathit{inv}(\ell)$. The set of all states is denoted by $S_{\mathcal{C}}$. The *transition relation* $\to$ consists of all triples $((\ell, \nu), a, (\ell', \nu'))$ satisfying the following conditions:

– $(\ell, \nu), (\ell', \nu')$ are states and $a \in Act \cup [0, \infty)$;
– if $a \in [0, \infty)$ then $\nu + \tau \models inv(\ell)$ for all $\tau \in [0, a]$ and $(\ell', \nu') = (\ell, \nu + a)$;
– if $a \in Act$ then $\nu \models enab(\ell, a)$ and there exists a pair $(X, \ell'') \in supp(prob(\ell, a))$ such that $(\ell', \nu') = (\ell'', \nu[X := 0])$.

By convention, we write $s \xrightarrow{a} s'$ instead of $(s, a, s') \in \rightarrow$. We omit '$\mathcal{C}$' in '$S_{\mathcal{C}}$' if the underlying context is clear.

*Probability Transition Kernel.* The *probability transition kernel* $\mathbf{P}$ is the function $\mathbf{P} : S \times Act \times S \to [0, 1]$ such that

$$\mathbf{P}((\ell, \nu), a, (\ell', \nu')) = \begin{cases} 1 & \text{if } (\ell, \nu) \xrightarrow{a} (\ell', \nu') \text{ and } a \in [0, \infty) \\ \sum_{Y \in B} prob(\ell, a)(Y, \ell') & \text{if } (\ell, \nu) \xrightarrow{a} (\ell', \nu') \text{ and } a \in Act \\ 0 & \text{otherwise} \end{cases}$$

where $B := \{X \subseteq \mathcal{X} \mid \nu' = \nu[X := 0]\}$.

*Well-formedness.* We say that $\mathcal{C}$ is *well-formed* if for every state $(\ell, \nu)$ and action $a \in Act$ such that $\nu \models enab(\ell, a)$ and every $(X, \ell') \in supp(prob(\ell, a))$, one has that $\nu[X := 0] \models inv(\ell')$. The well-formedness is to ensure that when an action is enabled, the next state after taking this action will always be legal. In the following, we always assume that the underlying PTA is well-formed. Non-well-formed PTAs can be repaired into well-formed PTAs [9].

*Paths.* A *finite path* $\rho$ (under $\mathcal{C}$) is a finite sequence $\langle s_0, a_0, s_1, \ldots, a_{n-1}, s_n \rangle$ $(n \geq 0)$ in $S \times ((Act \cup [0, \infty)) \times S)^*$ such that (i) $s_0 = (\ell^*, \mathbf{0})$, (ii) $a_{2k} \in [0, \infty)$ (resp. $a_{2k+1} \in Act$) for all integers $0 \leq k \leq \frac{n}{2}$ (resp. $0 \leq k \leq \frac{n-1}{2}$) and (iii) for all $0 \leq k \leq n - 1$, $s_k \xrightarrow{a_k} s_{k+1}$. The length $|\rho|$ of $\rho$ is defined by $|\rho| := n$. An *infinite path* (under $\mathcal{C}$) is an infinite sequence $\langle s_0, a_0, s_1, a_1, \ldots \rangle$ in $(S \times (Act \cup [0, \infty)))^\omega$ such that for all $n \in \mathbb{N}_0$, the prefix $\langle s_0, a_0, \ldots, a_{n-1}, s_n \rangle$ is a finite path. The set of finite (resp. infinite) paths under $\mathcal{C}$ is denoted by $Paths^*_{\mathcal{C}}$ (resp. $Paths^\omega_{\mathcal{C}}$).

*Schedulers.* A *(deteterministic) scheduler* is a function $\sigma$ from the set of finite paths into $Act \cup [0, \infty)$ such that for all finite paths $\rho = s_0 a_0 \ldots s_n$, (i) $\sigma(\rho) \in Act$ (resp. $\sigma(\rho) \in [0, \infty)$) if $n$ is odd (resp. even) and (ii) there exists a state $s'$ such that $s_n \xrightarrow{\sigma(\rho)} s'$.

*Paths under Schedulers.* A finite path $s_0 a_0 \ldots s_n$ *follows* a scheduler $\sigma$ if for all $0 \leq m < n$, $a_m = \sigma(s_0 a_0 \ldots s_m)$. An infinite path $s_0 a_0 s_1 a_1 \ldots$ *follows* $\sigma$ if for all $n \in \mathbb{N}_0$, $a_n = \sigma(s_0 a_0 \ldots s_n)$. The set of finite (resp. infinite) paths following a scheduler $\sigma$ is denoted by $Paths^*_{\mathcal{C},\sigma}$ (resp. $Paths^\omega_{\mathcal{C},\sigma}$). We note that the set $Paths^*_{\mathcal{C},\sigma}$ is countably infinite from definition.

*Probability Spaces under Schedulers.* Let $\sigma$ be any scheduler. The probability space w.r.t $\sigma$ is defined as $(\Omega^{\mathcal{C},\sigma}, \mathcal{F}^{\mathcal{C},\sigma}, \mathbb{P}^{\mathcal{C},\sigma})$ where (i) $\Omega^{\mathcal{C},\sigma} := Paths^\omega_{\mathcal{C},\sigma}$, (ii) $\mathcal{F}^{\mathcal{C},\sigma}$ is the smallest sigma-algebra generated by all cylinder sets induced by finite paths for which a finite path $\rho$ induces the cylinder set $Cyl(\rho)$ of all infinite paths in $Paths^\omega_{\mathcal{C},\sigma}$ with $\rho$ being their (common) prefix, and (iii) $\mathbb{P}^{\mathcal{C},\sigma}$ is the unique probability measure such that for all finite paths $\rho = s_0 a_0 \ldots a_{n-1} s_n$ in $Paths^*_{\mathcal{C},\sigma}$,

$$\mathbb{P}^{\mathcal{C},\sigma}(Cyl(\rho)) = \prod_{k=0}^{n-1} \mathbf{P}(s_k, \sigma(s_0 a_0 \ldots a_{k-1} s_k), s_{k+1}).$$

For details see [4].

*Zenoness and Time-Divergent Schedulers.* An infinite path $\pi = s_0 a_0 s_1 a_1 \ldots$ is *zeno* if $\sum_{n=0}^{\infty} d_n < \infty$, where $d_n := a_n$ if $a_n \in [0, \infty)$ and $d_n := 0$ otherwise. Then a scheduler $\sigma$ is *time divergent* if $\mathbb{P}^{\mathcal{C},\sigma}(\{\pi \mid \pi \text{ is zeno}\}) = 0$. In the following, we only consider time-divergent schedulers. The purpose is to eliminate non-realistic zeno behaviours (i.e., performing infinitely many actions within a finite amount of time).

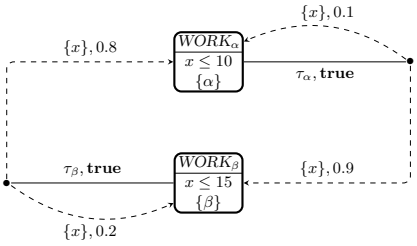


**Fig. 1.** A Simple Task-Processing Example

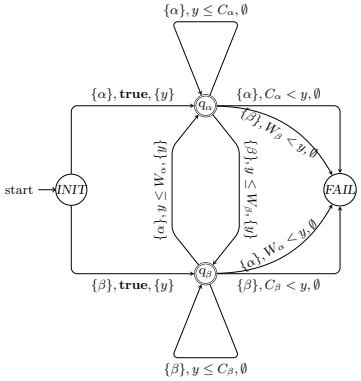


**Fig. 2.** A DTRA Specification

In the following example, we illustrate a PTA which models a simple task-processing example.

*Example 1.* Consider the PTA depicted in Figure 1. $WORK_\alpha, WORK_\beta$ are locations and $x$ is the only clock. Below each location first comes (vertically) its invariant condition and then the set of labels assigned to the location. For example, $inv(WORK_\alpha) = x \leq 10$ and $\mathcal{L}(WORK_\alpha) = \{\alpha\}$. The two dots together with their corresponding solid line and dashed arrows refer to two actions $\tau_\alpha, \tau_\beta$ with their enabling conditions and transition probabilities given by the probabilistic transition function. For example, the upper dot at the right of $WORK_\alpha$ refers to the action $\tau_\alpha$ for which $enab(WORK_\alpha, \tau_\alpha) = \mathbf{true}$, $prob(WORK_\alpha, \tau_\alpha)(\{x\}, WORK_\alpha) = 0.1$, and $prob(WORK_\alpha, \tau_\alpha)(\{x\}, WORK_\beta) = 0.9$. The PTA models a faulty machine which processes two different kinds of jobs (i.e., $\alpha, \beta$) in an alternating fashion. If the machine fails to complete the current job, then it will repeat processing the job until it completes the job. For job $\alpha$, the machine always processes the job within 10 time units (cf. the invariant condition $x \leq 10$), but may fail to complete the job with probability 0.1; Analogously, the machine always processes the job $\beta$ within 15 time units (cf. the invariant condition $x \leq 15$), but may fail to complete the job with probability 0.2. Note that we omit the initial location in this example.

### 2.2 Timed Automata

**Definition 2 (Timed Automata [22, 24, 25]).** *A* timed automaton *(TA) $\mathcal{A}$ is a tuple*

$$\mathcal{A} = (Q, \Sigma, \mathcal{Y}, \Delta) \tag{2}$$

*where*

– $Q$ *is a finite set of* modes*;*
– $\Sigma$ *is a finite* alphabet *of* symbols *disjoint from* $[0,\infty)$*;*
– $\mathcal{Y}$ *is a finite set of* clocks*;*
– $\Delta \subseteq Q \times \Sigma \times CC(\mathcal{Y}) \times 2^{\mathcal{Y}} \times Q$ *is a finite set of* rules.

$\mathcal{A}$ *is a* deterministic *TA (DTA) if the following holds:*

1. *(*determinism*) for* $(q_i, b_i, \phi_i, X_i, q_i') \in \Delta$ *(*$i \in \{1,2\}$*), if* $(q_1, b_1) = (q_2, b_2)$ *and* $[\![\phi_1]\!] \cap [\![\phi_2]\!] \neq \emptyset$ *then* $(\phi_1, X_1, q_1') = (\phi_2, X_2, q_2')$*;*
2. *(*totality*) for all* $(q, b) \in Q \times \Sigma$ *and* $\nu \in Val(\mathcal{X})$*, there exists* $(q, b, \phi, X, q') \in \Delta$ *such that* $\nu \models \phi$*.*

Informally, A TA is deterministic if there is always exactly one rule applicable for the timed transition. We do not incorporate invariants in TAs as we use TAs as language acceptors.

Below we illustrate the semantics of TAs. We fix a TA $\mathcal{A}$ in the form (2).

*Configurations and One-Step Transition Relation.* A *configuration* is a pair $(q, \nu)$, where $q \in Q$ and $\nu \in Val(\mathcal{Y})$. The *one-step transition relation*

$$\Rightarrow \;\subseteq\; (Q \times Val(\mathcal{Y})) \times (\Sigma \cup [0,\infty)) \times (Q \times Val(\mathcal{Y}))$$

is defined by: $((q,\nu), a, (q',\nu')) \in \Rightarrow$ iff either (i) $a \in [0,\infty)$ and $(q',\nu') = (q, \nu+a)$ or (ii) $a \in \Sigma$ and there exists a rule $(q, a, \phi, X, q') \in \Delta$ such that $\nu \models \phi$ and $\nu' = \nu[X := 0]$. For the sake of convenience, we write $(q,\nu) \overset{a}{\Rightarrow} (q',\nu')$ instead of $((q,\nu), a, (q',\nu')) \in \Rightarrow$. Note that if $\mathcal{A}$ is deterministic, then there is a unique $(q',\nu')$ such that $(q,\nu) \overset{a}{\Rightarrow} (q',\nu')$ given any $(q,\nu), a$.

*Infinite Timed Words and Runs.* An *infinite timed word* is an infinite sequence $w = \{a_n\}_{n \in \mathbb{N}_0}$ such that $a_{2n} \in [0,\infty)$ and $a_{2n+1} \in \Sigma$ for all $n$; the infinite timed word $w$ is *time-divergent* if $\sum_{n \in \mathbb{N}_0} a_{2n} = \infty$. A *run* of $\mathcal{A}$ on an infinite timed word $w = \{a_n\}_{n \in \mathbb{N}_0}$ with *initial configuration* $(q,\nu)$, is an infinite sequence $\xi = \{(q_n, \nu_n, a_n)\}_{n \in \mathbb{N}_0}$ satisfying that $(q_0,\nu_0) = (q,\nu)$ and $(q_n,\nu_n) \overset{a_n}{\Rightarrow} (q_{n+1},\nu_{n+1})$ for all $n \in \mathbb{N}_0$; the *trajectory* $traj(\xi)$ of the run $\xi$ is defined as an infinite word over $Q$ such that $traj(\xi) := q_0 q_1 \dots$ . Note that if $\mathcal{A}$ is deterministic, then there is a unique run on every infinite timed word.

Below we illustrate the acceptance condition for TAs. We consider Rabin acceptance condition as the infinite acceptance condition.

**Definition 3 (Rabin Acceptance Condition [1]).** *A TA with* Rabin acceptance condition *(TRA) is a tuple*

$$\mathcal{A} = (Q, \Sigma, \mathcal{Y}, \Delta, \mathcal{F}) \tag{3}$$

*where* $(Q, \Sigma, \mathcal{Y}, \Delta)$ *is a TA and* $\mathcal{F}$ *is a finite set of pairs* $\mathcal{F} = \{(H_1, K_1), \dots, (H_n, K_n)\}$ *representing a Rabin condition for which* $H_i$ *and* $K_i$ *are subsets of* $Q$ *for all* $i \leq n$*.* $\mathcal{A}$ *is a* deterministic *TRA (DTRA) if* $(Q, \Sigma, \mathcal{Y}, \Delta)$ *is a DTA. A set* $Q' \subseteq Q$ *is* Rabin-accepting *by* $\mathcal{F}$*, written as the predicate* $\mathbf{ACC}(Q', \mathcal{F})$*, if there is* $1 \leq i \leq n$ *such that* $Q' \cap H_i = \emptyset$ *and* $Q' \cap K_i \neq \emptyset$*. An infinite timed word* $w$ *is* Rabin-accepted *by* $\mathcal{A}$ *with* initial configuration $(q,\nu)$ *iff there exists a run* $\xi$ *of* $(Q, \Sigma, \mathcal{Y}, \Delta)$ *on* $w$ *with* $(q,\nu)$ *such that* $inf(traj(\xi))$ *is Rabin-accepting by* $\mathcal{F}$*.*

*Example 2.* Consider the DTRA depicted in Figure 2. The alphabet of this DTRA is the powerset of atomic propositions in Figure 1. In the figure, *INIT*, $q_\alpha, q_\beta$ and *FAIL* are modes with the Rabin condition $\mathcal{F} = \{(\{FAIL\}, \{q_\alpha, q_\beta\})\}$, $y$ is a clock and arrows between modes are rules. $C_\gamma, W_\gamma$ ($\gamma \in \{\alpha, \beta\}$) are undetermined integer constants. For example, there are four rules emitting from $q_\alpha$:

$$(q_\alpha, \{\alpha\}, y \leq C_\alpha, \emptyset, q_\alpha),\ (q_\alpha, \{\beta\}, y \leq W_\beta, \{y\}, q_\beta),$$
$$(q_\alpha, \{\alpha\}, C_\alpha < y, \emptyset, FAIL),\ (q_\alpha, \{\beta\}, W_\beta < y, \emptyset, FAIL).$$

*INIT* is the initial mode to read the first symbol upon which transiting to either $q_\alpha$ or $q_\beta$. *FAIL* is a deadlock mode from which all rules go to itself. Note that the rules of the DTRA does not satisfy the totality condition. However, we assume that all missing rules lead to the mode *FAIL* and does not affect the Rabin acceptance condition. The mode $q_\alpha$ does not reset the clock $y$ until it reads $\beta$. Moreover, $q_\alpha$ does not transit to *FAIL* only if the time spent within a maximal consecutive segment of $\alpha$'s (in an infinite timed word) is no greater than $C_\alpha$ time units (cf. the rule $(q_\alpha, \{\alpha\}, y \leq C_\alpha, \emptyset, q_\alpha)$) and the total time from the start of the segment until $\beta$ is read (the time within a maximal consecutive segment of $\alpha$'s plus the time spent on the last $\alpha$ in the segment) is no greater than $W_\beta$ (cf. the rule $(q_\alpha, \{\beta\}, y \leq W_\beta, \{y\}, q_\beta)$). The behaviour of the mode $q_\beta$ can be argued similar to that of $q_\alpha$ where the only difference is to flip $\alpha$ and $\beta$. From the Rabin acceptance condition, the DTRA specifies a property on infinite timed words that the time spent within a maximal consecutive segment of $\alpha$'s (resp. $\beta$'s) and the total time until $\beta$ (resp. $\alpha$) is read always satisfy the conditions specified by $q_\alpha$ (resp. $q_\beta$).

## 3 Problem Statement

In this part, we define the PTA-TRA problem of model-checking PTAs against TA-specifications. The problem takes a PTA and a TRA as input, and computes the minimum and the maximum probability that infinite paths under the PTA are accepted by the TRA. Informally, the TRA encodes the linear dense-time property by judging whether an infinite path is accepted or not through its external behaviour, then the problem is to compute the probability that an infinite path is accepted by the TRA. In practice, the TRA can be used to capture all good (or bad) behaviours, so the problem can be treated as a task to evaluate to what extent the PTA behaves in a good (or bad) way.

Below we fix a well-formed PTA $\mathcal{C}$ taking the form (1) and a TRA $\mathcal{A}$ taking the form (3). W.l.o.g., we assume that $\mathcal{X} \cap \mathcal{Y} = \emptyset$ and $\Sigma = 2^{AP}$. We first show how an infinite path in $Paths^\omega_\mathcal{C}$ can be interpreted as an infinite timed word.

**Definition 4 (Infinite Paths as Infinite Timed Words).** *Given an infinite path $\pi = (\ell_0, \nu_0)a_0(\ell_1, \nu_1)a_1(\ell_2, \nu_2)a_2 \dots$ under $\mathcal{C}$, the infinite timed word $\mathcal{L}(\pi)$ is defined as $\mathcal{L}(\pi) := a_0\mathcal{L}(\ell_2)a_2\mathcal{L}(\ell_4)\dots a_{2n}\mathcal{L}(\ell_{2n+2})\dots$ . Recall that $\nu_0 = \mathbf{0}$, $a_{2n} \in [0, \infty)$ and $a_{2n+1} \in Act$ for $n \in \mathbb{N}_0$.*

*Remark 1.* Informally, the interpretation in Definition 4 works by (i) dropping (a) the initial location $\ell_0$, (b) all clock valuations $\nu_n$'s, (c) all locations $\ell_{2n+1}$'s

following a time-elapse, (d) all internal actions $a_{2n+1}$'s of $\mathcal{C}$ and (ii) replacing every $\ell_{2n}$ $(n \geq 1)$ by $\mathcal{L}(\ell_{2n})$. The interpretation captures only external behaviours including time-elapses and labels of locations upon state-change, and discards internal behaviours such as the concrete locations, clock valuations and actions. Although the interpretation ignores the initial location, we deal with it in our acceptance condition where the initial location is preprocessed by the TRA.

**Definition 5 (Path Acceptance).** *An infinite path $\pi$ of $\mathcal{C}$ is* accepted *by $\mathcal{A}$ w.r.t initial configuration $(q, \nu)$, written as the single predicate* $\mathbf{ACC}\left(\mathcal{A}, (q, \nu), \pi\right)$, *if there is a configuration $(q', \nu')$ such that $(q, \nu) \xrightarrow{\mathcal{L}(\ell^*)} (q', \nu')$ and the infinite word $\mathcal{L}(\pi)$ is Rabin-accepted by $\mathcal{A}$ with $(q', \nu')$.*

The initial location omitted in Definition 4 is preprocessed by specifying explicitly that the first label $\mathcal{L}(\ell^*)$ is read by the initial configuration $(q, \nu)$. Below we define acceptance probabilities over infinite paths under $\mathcal{C}$.

**Definition 6 (Acceptance Probabilities).** *The probability that $\mathcal{C}$* observes *$\mathcal{A}$ under scheduler $\sigma$ and initial mode $q \in Q$, denoted by $\mathfrak{p}_q^\sigma$, is defined by:*

$$\mathfrak{p}_q^\sigma := \mathbb{P}^{\mathcal{C},\sigma}(AccPaths_{\mathcal{C},\sigma}^{\mathcal{A},q})$$

*where $AccPaths_{\mathcal{C},\sigma}^{\mathcal{A},q}$ is the set of infinite paths under $\mathcal{C}$ that are accepted by the TRA $\mathcal{A}$ w.r.t $(q, \mathbf{0})$ i.e. $AccPaths_{\mathcal{C},\sigma}^{\mathcal{A},q} = \left\{\pi \in Paths_{\mathcal{C},\sigma}^{\omega} \mid \mathbf{ACC}\left(\mathcal{A}, (q, \mathbf{0}), \pi\right)\right\}$.*

Since the set $Paths_{\mathcal{C},\sigma}^{*}$ is countably-infinite, $AccPaths_{\mathcal{C},\sigma}^{\mathcal{A},q}$ is measurable since it can be represented as a countable intersection of certain countable unions of some cylinder sets (cf. [1, Remark 10.24] for details).

Now we introduce the PTA-TRA problem.

- **Input:** a well-formed PTA $\mathcal{C}$, a TRA $\mathcal{A}$ and an initial mode $q$ in $\mathcal{A}$;
- **Output:** $\inf_\sigma \mathfrak{p}_q^\sigma$ and $\sup_\sigma \mathfrak{p}_q^\sigma$, where $\sigma$ ranges over all time-divergent schedulers for $\mathcal{C}$.

We refer to the problem as PTA-DTRA if $\mathcal{A}$ is deterministic.

## 4 The PTA-DTRA Problem

In this section, we solve the PTA-DTRA problem through a product construction. Based on the product construction, we also settle the complexity of the problem. Below we fix a well-formed PTA $\mathcal{C}$ in the form (1) and a DTRA $\mathcal{A}$ in the form (3). W.l.o.g, we consider that $\mathcal{X} \cap \mathcal{Y} = \emptyset$ and $\Sigma = 2^{AP}$.

**The Main Idea.** The core part of the product construction is a PTA which preserves the probability of the set of infinite paths accepted by $\mathcal{A}$. The intuition is to let $\mathcal{A}$ reads external actions of $\mathcal{C}$ while $\mathcal{C}$ evolves along the time axis. The major difficulty is that when $\mathcal{C}$ performs actions in $Act$, there is a probabilistic choice between the target locations. Then $\mathcal{A}$ needs to know the labelling of the target location and the rule (in $\Delta$) used for the transition. A naive solution is to integrate each single rule in $\Delta$ into the enabling condition *enab* in $\mathcal{C}$. However, this simple solution does not work since a single rule fixes the labelling of a location in $\mathcal{C}$, while the probability distribution (given by *prob*) can jump to

locations with different labels. We solve this difficulty by integrating into the enabling condition enough information on clock valuations under $\mathcal{A}$ so that the rule used for the transition is clear.

**The Product Construction.** For each $q \in Q$, we let

$$\mathcal{T}_q := \{h : \Sigma \to CC(\mathcal{Y}) \mid \forall b \in \Sigma.\, (q, b, h(b), X, q') \in \Delta \text{ for some } X, q')\} \ .$$

The totality of $\Delta$ ensures that $\mathcal{T}_q$ is non-empty. Intuitively, every element of $\mathcal{T}_q$ is a tuple of clock constraints $\{\phi_b\}_{b\in\Sigma}$, where each clock constraint $\phi_b$ is chosen from the rules emitting from $q$ and $b$. The *product PTA* $\mathcal{C}\otimes\mathcal{A}_q$ (between $\mathcal{C}$ and $\mathcal{A}$ with initial mode $q$) is defined as $\left(L_\otimes, \ell^*_\otimes, \mathcal{X}_\otimes, Act_\otimes, inv_\otimes, enab_\otimes, prob_\otimes, Q, \mathcal{L}_\otimes\right)$ where :

- $L_\otimes := L \times Q$;
- $\ell^*_\otimes := (\ell^*, q^\star)$ where $q^\star$ is the unique mode such that $(q, \mathbf{0}) \xrightarrow{\mathcal{L}(\ell^*)} (q^\star, \mathbf{0})$;
- $\mathcal{X}_\otimes := \mathcal{X} \cup \mathcal{Y}$;
- $Act_\otimes := Act \times \bigcup_q \mathcal{T}_q$;
- $inv_\otimes(\ell, q) := inv(\ell)$ for all $(\ell, q) \in L_\otimes$;
- $enab_\otimes((\ell, q), (a, h)) := enab(\ell, a) \wedge \bigwedge_{b\in\Sigma} h(b)$ if $h \in \mathcal{T}_q$,and $enab_\otimes((\ell, q), (a, h)) :=$ **false** otherwise, for all $(\ell, q) \in L_\otimes$, $(a, h) \in Act_\otimes$.
- $\mathcal{L}_\otimes(\ell, q) := \{q\}$ for all $(\ell, q) \in L_\otimes$;
- $prob_\otimes$ is given by

$$prob_\otimes((\ell, q), (a, h))(Y, (\ell', q')) := \begin{cases} prob(\ell, a)(Y \cap \mathcal{X}, \ell') & \text{if } (q, \mathcal{L}(\ell'), h(\mathcal{L}(\ell')), Y \cap \mathcal{Y}, q') \\ & \quad \text{is a (unique) rule in } \Delta \\ 0 & \text{otherwise} \end{cases}$$

  for all $(\ell, q), (\ell', q') \in L_\otimes$, $(a, h) \in Act_\otimes$ and $Y \in \mathcal{X}_\otimes$.

Besides standard constructions (e.g., the Cartesian product between $L$ and $Q$), the product construction also has Cartesian product between $Act$ and $\bigcup_q \mathcal{T}_q$. For each extended action $(a, h)$, the enabling condition for this action is the conjunction between $enab(\ell, a)$ and all clock constraints from $h$. This is to ensure that when the action $(a, h)$ is taken, the clock valuation under $\mathcal{A}$ satisfies every clock constraint in $h$. Then in the definition for $prob_\otimes$, upon the action $(a, h)$, the product PTA first perform probabilistic jump from $\mathcal{C}$ with the target location $\ell'$, then chooses the unique rule $(q, \mathcal{L}(\ell'), h(\mathcal{L}(\ell')), Y \cap \mathcal{Y}, q')$ from the emitting mode $q$ and the label $\mathcal{L}(\ell')$ for which the uniqueness comes from the determinism of $\Delta$, then perform the discrete transition from $\mathcal{A}$. Finally, we label each $(\ell, q)$ by $q$ to meet the Rabin acceptance condition. □

It is easy to see that the PTA $\mathcal{C}\otimes\mathcal{A}_q$ is well-formed as $\mathcal{C}$ is well-formed and $\mathcal{A}$ does not introduce extra invariant conditions.

*Example 3.* The product PTA between the PTA in Example 1 and the DTRA in Example 2 is depicted in Figure 3. In the figure, $(WORK_\alpha, q_\alpha), (WORK_\beta, q_\beta)$ and $(WORK_\alpha, FAIL), (WORK_\beta, FAIL)$ are product locations. We omit the initial location and unreachable locations in the product construction. From the

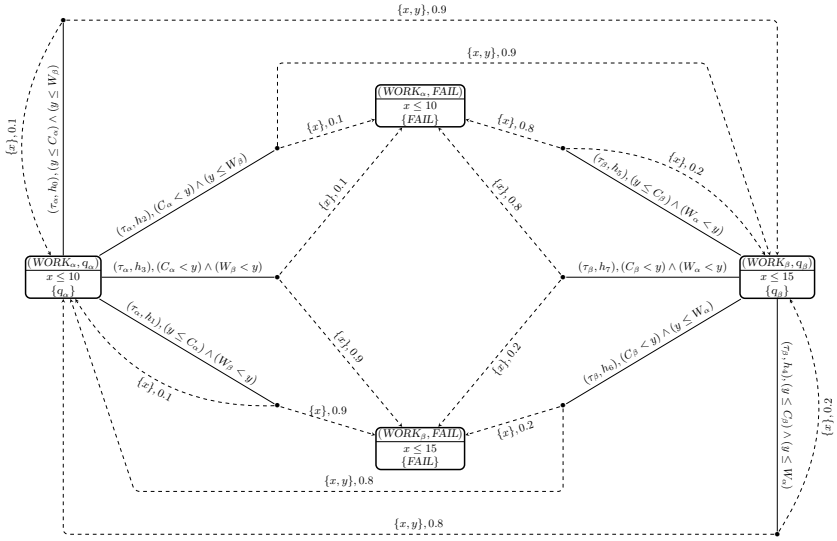


**Fig. 3.** The Product PTA for Our Running Example

construction of $\mathcal{T}_q$'s, the functions $h_i$'s are as follows (we omit redundant labels such as $\emptyset$ and $\{\alpha,\beta\}$ which never appear in the PTA):

- $h_0 = \{\{\alpha\} \mapsto y \le C_\alpha, \{\beta\} \mapsto y \le W_\beta\}$;
- $h_1 = \{\{\alpha\} \mapsto y \le C_\alpha, \{\beta\} \mapsto W_\beta < y\}$;
- $h_2 = \{\{\alpha\} \mapsto C_\alpha < y, \{\beta\} \mapsto y \le W_\beta\}$;
- $h_3 = \{\{\alpha\} \mapsto C_\alpha < y, \{\beta\} \mapsto W_\beta < y\}$;
- $h_4 = \{\{\beta\} \mapsto y \le C_\beta, \{\alpha\} \mapsto y \le W_\alpha\}$;
- $h_5 = \{\{\beta\} \mapsto y \le C_\beta, \{\alpha\} \mapsto W_\alpha < y\}$;
- $h_6 = \{\{\beta\} \mapsto C_\beta < y, \{\alpha\} \mapsto y \le W_\alpha\}$;
- $h_7 = \{\{\beta\} \mapsto C_\beta < y, \{\alpha\} \mapsto W_\alpha < y\}$.

The intuition is that the DTA accepts all infinite paths under the PTA such that the failing time for job $\gamma$ ($\gamma \in \{\alpha,\beta\}$) (the time within the consecutive $\gamma$'s) should be no greater than $C_\gamma$ and the waiting time for job $\gamma$ (the failing time plus the time spent on the last $\gamma$) should be no greater than $W_\gamma$.

Below we clarify the correspondence between $\mathcal{C}, \mathcal{A}$ and $\mathcal{C}\otimes\mathcal{A}_q$. We first show the relationship between paths under $\mathcal{C}$ and those under $\mathcal{C}\otimes\mathcal{A}_q$. Informally, paths under $\mathcal{C}\otimes\mathcal{A}_q$ are just paths under $\mathcal{C}$ extended with runs of $\mathcal{A}$.

**Transformation $\mathcal{T}$ for Paths from $\mathcal{C}$ into $\mathcal{C}\otimes\mathcal{A}_q$.** The transformation is defined as the function $\mathcal{T} : \mathit{Paths}^*_{\mathcal{C}} \cup \mathit{Paths}^\omega_{\mathcal{C}} \to \mathit{Paths}^*_{\mathcal{C}\otimes\mathcal{A}_q} \cup \mathit{Paths}^\omega_{\mathcal{C}\otimes\mathcal{A}_q}$ which transform a finite or infinite path under $\mathcal{C}$ into one under $\mathcal{C}\otimes\mathcal{A}_q$. For a finite path $\rho = (\ell_0,\nu_0)a_0 \dots a_{n-1}(\ell_n,\nu_n)$ under $\mathcal{C}$ (note that $(\ell_0,\nu_0) = (\ell^*,\mathbf{0})$ by definition), we define $\mathcal{T}(\rho)$ to be the unique finite path

$$\mathcal{T}(\rho) := ((\ell_0,q_0),\nu_0\cup\mu_0)a'_0 \dots a'_{n-1}((\ell_n,q_n),\nu_n\cup\mu_n) \tag{4}$$

under $\mathcal{C}\otimes\mathcal{A}_q$ such that the following conditions (†) hold:

– $(q, \mathbf{0}) \xRightarrow{\mathcal{L}(\ell^*)} (q_0, \mu_0)$ (note that $\mu_0 = \mathbf{0}$);
– for all $0 \leq k < n$, if $a_k \in [0, \infty)$ then $a'_k = a_k$ and $(q_k, \mu_k) \xRightarrow{a_k} (q_{k+1}, \mu_{k+1})$;
– for all $0 \leq k < n$, if $a_k \in Act$ then $a'_k = (a_k, \xi_k)$ and $(q_k, \mu_k) \xRightarrow{\mathcal{L}(\ell_{k+1})} (q_{k+1}, \mu_{k+1})$ where $\xi_k$ is the unique function such that for each symbol $b \in \Sigma$, $\xi_k(b)$ is the unique clock constraint appearing in a rule emitting from $q_k$ and with symbol $b$ such that $\mu_k \models \xi_k(b)$.

Likewise, for an infinite path $\pi = (\ell_0, \nu_0)a_0(\ell_1, \nu_1)a_1 \dots$ under $\mathcal{C}$, we define $\mathcal{T}(\pi)$ to be the unique infinite path

$$\mathcal{T}(\pi) := ((\ell_0, q_0), \nu_0 \cup \mu_0)a'_0((\ell_1, q_1), \nu_1 \cup \mu_1)a'_1 \dots \tag{5}$$

under $\mathcal{C} \otimes \mathcal{A}_q$ such that the three conditions in (†) hold for all $k \in \mathbb{N}_0$ instead of all $0 \leq k < n$. From the determinism and totality of $\mathcal{A}$, it is straightforward to prove the following result.

**Lemma 1.** *The function $\mathcal{T}$ is a bijection. Moreover, for any infinite path $\pi$ under $\mathcal{C}$, $\pi$ is non-zeno iff $\mathcal{T}(\pi)$ is non-zeno.*

Below we also show the correspondence on schedulers before and after the product construction.

**Transformation $\theta$ for Schedulers from $\mathcal{C}$ into $\mathcal{C} \otimes \mathcal{A}_q$.** We define the function $\theta$ from the set of schedulers under $\mathcal{C}$ into the set of schedulers under $\mathcal{C} \otimes \mathcal{A}_q$ as follows: for any scheduler $\sigma$ for $\mathcal{C}$, $\theta(\sigma)$ (for $\mathcal{C} \otimes \mathcal{A}_q$) is defined such that for any finite path $\rho$ under $\mathcal{C}$ where $\rho = (\ell_0, \nu_0)a_0 \dots a_{n-1}(\ell_n, \nu_n)$ and $\mathcal{T}(\rho)$ given as in (4),

$$\theta(\sigma)(\mathcal{T}(\rho)) := \begin{cases} \sigma(\rho) & \text{if } n \text{ is even} \\ (\sigma(\rho), \lambda(\rho)) & \text{if } n \text{ is odd} \end{cases}$$

where $\lambda(\rho)$ is the unique function such that for each symbol $b \in \Sigma$, $\lambda(\rho)(b)$ is the clock constraint in the unique rule emitting from $q_n$ and with symbol $b$ such that $\mu_n \models \lambda(\rho)(b)$. Note that the well-definedness of $\theta$ follows from Lemma 1.

From Lemma 1, the product construction, the determinism and totality of $\Delta$, one can prove directly the following lemma.

**Lemma 2.** *The function $\theta$ is a bijection.*

Now we prove the relationship between infinite paths accepted by a DTRA before product construction and infinite paths satisfying certain Rabin condition.

We introduce more notations. First, we lift the function $\mathcal{T}$ to all subsets of paths in the standard fashion: for all subsets $A \subseteq Paths^*_{\mathcal{C}} \cup Paths^{\omega}_{\mathcal{C}}$, $\mathcal{T}(A) := \{\mathcal{T}(\omega) \mid \omega \in A\}$. Then for an infinite path $\pi$ under $\mathcal{C} \otimes \mathcal{A}_q$ in the form (5), we define the *trace* of $\pi$ as an infinite word over $Q$ by $trace(\pi) := q_0 q_1 \dots$ . Finally, for any scheduler $\sigma$ for $\mathcal{C} \otimes \mathcal{A}_q$, we define the set $RPaths_\sigma$ by

$$RPaths_\sigma := \left\{ \pi \in Paths^{\omega}_{\mathcal{C} \otimes \mathcal{A}_q, \sigma} \mid \mathbf{ACC}\left(inf(trace(\pi)), \mathcal{F}\right) \right\}.$$

Intuitively, $RPaths_\sigma$ is the set of infinite paths under $\mathcal{C} \otimes \mathcal{A}_q$ that meet the Rabin condition $\mathcal{F}$ from $\mathcal{A}$. The following proposition clarifies the role of $RPaths_\sigma$.

**Proposition 1.** *For any scheduler $\sigma$ for $\mathcal{C}$ and any initial mode $q$ for $\mathcal{A}$, we have* $\mathcal{T}\left(AccPaths_{\mathcal{C},\sigma}^{\mathcal{A},q}\right) = RPaths_{\theta(\sigma)}$.

Finally, we demonstrate the relationship between acceptance probabilities before product construction and Rabin(-accepting) probabilities after product construction. We also clarify the probability of zenoness before and after the product construction. The proof follows standard argument from measure theory.

**Proposition 2.** *For any scheduler $\sigma$ for $\mathcal{C}$ and mode $q$, the followings hold:*
- $\mathfrak{p}_q^\sigma = \mathbb{P}^{\mathcal{C},\sigma}\left(AccPaths_{\mathcal{C},\sigma}^{\mathcal{A},q}\right) = \mathbb{P}^{\mathcal{C}\otimes\mathcal{A}_q,\theta(\sigma)}\left(RPaths_{\theta(\sigma)}\right)$;
- $\mathbb{P}^{\mathcal{C},\sigma}\left(\{\pi \mid \pi \textit{ is zeno}\}\right) = \mathbb{P}^{\mathcal{C}\otimes\mathcal{A}_q,\theta(\sigma)}\left(\{\pi' \mid \pi' \textit{ is zeno}\}\right)$.

A side result from Proposition 2 says that $\theta$ preserves time-divergence for schedulers before and after product construction. From Proposition 2 and Lemma 2, one immediately obtains the following result which transforms the PTA-DTRA problem into Rabin(-accepting) probabilities under the product PTA.

**Corollary 1.** *For any initial mode $q$, $opt_\sigma \mathfrak{p}_q^\sigma = opt_{\sigma'} \mathbb{P}^{\mathcal{C}\otimes\mathcal{A}_q,\sigma'}(RPaths_{\sigma'})$ where opt refers to either* inf *(infimum) or* sup *(supremum), $\sigma$ (resp. $\sigma'$) range over all time-divergent schedulers for $\mathcal{C}$ (resp. $\mathcal{C}\otimes\mathcal{A}_q$).*

**Solving Rabin Probabilities.** We follow the approach in [18] to solve Rabin probabilities over PTAs. Below we briefly describe the approach. The approach can be divided into two steps. The first step is to ensure time-divergence. This is achieved by (i) making a copy for every location in the PTA, (ii) enforcing a transition from every location to its copy to happen after 1 time-unit elapses, (iii) enforcing a transition from every copy location back to the original one immediately with no time-delay, and (iv) putting a special label *tick* in every copy. Then time-divergence is guaranteed by adding the label *tick* to the Rabin condition. The second step is to transform the problem into limit Rabin properties over MDPs [1, Theorem 10.127]. This step constructs an MDP $\mathrm{Reg}[\mathcal{C}\otimes\mathcal{A}_q]$ from the PTA $\mathcal{C}\otimes\mathcal{A}_q$ through a *region-graph* construction so that the problem is reduced to solving limit Rabin properties over $\mathrm{Reg}[\mathcal{C}\otimes\mathcal{A}_q]$. *Regions* are finitely-many equivalence classes of clock valuations that serve as a finite abstraction which capture exactly reachability behaviours over timed transitions (cf. [5]). Then standard methods based on *maximal end components* (MECs) are applied to $\mathrm{Reg}[\mathcal{C}\otimes\mathcal{A}_q]$. In detail, the algorithm computes the reachability probability to MECs that satisfy the Rabin acceptance condition. In order to guarantee time-divergence, the algorithm only picks up MECs with at least one location that has a *tick* label. Based on this approach, our result leads to an algorithm for solving the problem PTA-DTRA.

Note that in $\mathcal{C}\otimes\mathcal{A}_q$, although the size of $Act_\otimes$ may be exponential due to possible exponential blow-up from $\mathcal{T}_q$, one easily sees that $|L_\otimes|$ is $|L|\cdot|Q|$ and $|\mathcal{X}_\otimes| = |\mathcal{X}| + |\mathcal{Y}|$. Hence, the size of $\mathrm{Reg}[\mathcal{C}\otimes\mathcal{A}_q]$ is still exponential in the sizes of $\mathcal{C}$ and $\mathcal{A}$. It follows that $opt_\sigma \mathfrak{p}_q^\sigma$ can be calculated in exponential time from the MEC-based algorithm illustrated in [1, Theorem 10.127], as is demonstrated by the following proposition.

**Proposition 3.** *The problem* PTA-DTRA *is in EXPTIME in the size of the input PTA and DTRA.*

It is proved in [26] that the reachablity-probability problem for arbitrary PTAs is *EXPTIME*-complete. Since Rabin acceptance condition subsumes reachability, one obtains that the problem PTA-DTRA is EXPTIME-hard . Thus we obtain the main result of this section which settles the computational complexity of the problem PTA-DTRA.

**Theorem 1.** *The PTA-DTRA problem is EXPTIME-complete.*

*Remark 2.* The main novelty for our product construction is that by adopting extended actions (i.e. $\mathcal{T}_q$) and integrating them into enabling condition and probabilistic transition function, the product PTA can know which rule to use from the DTA upon any symbol to be read. This solves the problem that probabilistic jumps can lead to different locations, causing the usage of different rules from the DTA. Moreover, our product construction ensures EXPTIME-completeness of the problem.

## 5 The PTA-TRA problem

In this section, we study the PTA-TRA problem where the input timed automaton needs not to be deterministic. In contrast to the deterministic case (which is shown to be decidable and EXPTIME-complete in the previous section), we show that the problem is undecidable.

**The Main Idea.** The main idea for the undecidability result is to reduce the universality problem of timed automata to the PTA-TRA problem. The universality problem over timed automata is well-known to be undecidable, as follows.

**Lemma 3.** *([5, Theorem 5.2]) Given a timed automaton over an alphabet $\Sigma$ and an initial mode, the problem of deciding whether it accepts all time-divergent timed words w.r.t Büchi acceptance condition over $\Sigma$ is undecidable.*

Although Lemma 3 is on Büchi acceptance condition, it holds also for Rabin acceptance condition since Rabin acceptance condition extends Büchi acceptance condition. Actually the two acceptance conditions are equivalent over timed automata (cf. [5, Theorem 3.20]). We also remark that Lemma 3 was originally for multiple initial modes, which can be mimicked by a single initial mode through aggregating all rules emitting from some initial mode as rules emitting from one initial mode.

Now we prove the undecidability result as follows. The proof idea is that we construct a PTA that can generate every time-divergent timed words with probability 1 by some time-divergent scheduler. Then the TRA accepts all time-divergent timed words iff the minimal probability that the PTA observes the TRA equals 1.

**Theorem 2.** *Given a PTA $\mathcal{C}$ and a TRA $\mathcal{A}$, the problem to decide whether the minimal probability that $\mathcal{C}$* observes *$\mathcal{A}$ (under a given initial mode) is equal to* 1 *is undecidable.*

*Proof (Proof Sketch).* Let $\mathcal{A} = (Q, \Sigma, \mathcal{Y}, \Delta, \mathcal{F})$ be any TRA where the alphabet $\Sigma = \{b_1, b_2, \cdots, b_k\}$ and the initial mode is $q_{start}$. W.l.o.g, we consider that $\Sigma \subseteq 2^{AP}$ for some finite set $AP$. This assumption is not restrictive since what $b_i$'s concretely are is irrelevant, while the only thing that matters is that $\Sigma$ has $k$ different symbols. We first construct the TRA $\mathcal{A}' = (Q', \Sigma', \mathcal{Y}, \Delta', \mathcal{F})$ where $Q' = Q \cup \{q_{init}\}$ for which $q_{init}$ is a fresh mode, $\Sigma' = \Sigma \cup \{b_0\}$ for which $b_0$ is a fresh symbol and $\Delta' = \Delta \cup \{\langle q_{init}, b_0, \mathbf{true}, \mathcal{Y}, q_{start}\rangle\}$. Then we construct the PTA :

- $L := \Sigma'$, $\ell^* := b_0$, $\mathcal{X} := \emptyset$ and $Act := \Sigma$;
- $inv(b_i) := \mathbf{true}$ for $b_i \in L$;
- $enab(b_i, b_j) := \mathbf{true}$ for $b_i \in L$ and $b_j \in Act$;
- $prob(b_i, b_j)$ is the Dirac distribution at $(\emptyset, b_j)$ (i.e., $prob(b_i, b_j)(\emptyset, b_j) = 1$ and $prob(b_i, b_j)(X, b) = 0$ whenever $(X, b) \neq (\emptyset, b_j)$), for $b_i \in L$ and $b_j \in Act$;
- $\mathcal{L}(b_i) := b_i$ for $b_i \in L$.

Note that we allow no clocks in the construction since clocks are irrelevant for our result. Since we omit clocks, we also treat states (of $\mathcal{C}'$) as single locations. One can prove that $\mathcal{A}$ accepts all time-divergent timed words over $\Sigma$ with initial mode $q_{start}$ iff the minimal probability that $\mathcal{C}'$ observes $\mathcal{A}'$ with initial mode $q_{init}$ equals 1. □

*Remark 3.* Theorem 2 shows that the problem to qualitatively decide the minimal probability is undecidable. On the other hand, the decidability of the problem to decide maximum acceptance probabilities is left open.

## 6 Conclusion

In this paper, we studied the problem of model-checking PTAs against timed-automata specifications. We considered Rabin acceptance condition as the acceptance criterion. We first solved the problem with deterministic-timed-automata specifications through a product construction and proved that its computational complexity is EXPTIME-complete. Then we proved that the problem with general timed-automata specifications is undecidable through a reduction from the universality problem of timed automata.

A future direction is zone-based algorithms for Rabin acceptance condition. Another direction is to investigate timed-automata specifications with cost or reward. Besides, it is also interesting to explore model-checking PTAs against Metric Temporal Logic [19].

## 7 Related Works

Model-checking TAs or MDPs against omega-regular (dense-time) properties is well-studied (cf. [1, 20, 27], etc.). PTAs extend both TAs and MDPs with either probability or timing constraints, hence require new techniques for verification problems. On one hand, our technique extends techniques for MDPs (e.g. [27]) with timing constraints. On the other hand, our technique is incomparable to techniques for TAs since linear-time model checking of TAs focus mostly on proving decidability of temporal logic formulas (e.g. Metric Temporal logic [19, 21, 20]), while we prove that model-checking PTAs against TA-specifications is undecidable.

Model-checking probabilistic timed models against linear dense-time properties are mostly considered for continuous-time Markov processes (CTMPs). First, Donatelli *et al.* [22] proved an expressibility result that the class of linear dense-time properties encoded by DTAs is not subsumed by branching-time properties. They also demonstrated an efficient algorithm for verifying continuous-time Markov chains [22] against one-clock DTAs. Then various results on verifying CTMPs are obtained for specifications through DTAs and general timed automata (cf. e.g. [22, 24, 25, 28, 29, 30]). The fundamental difference between CTMPs and PTAs is that the former assign probability distributions to time elapses, while the latter treat time-elapses as pure nondeterminism. As a consequence, the techniques for CTMPs cannot be applied to PTAs.

For PTAs, the only relevant result is by Sproston [18] who developed an approach for verifying PTAs against deterministic discrete-time omega-regular automata by a similar product construction. Our results extend his result in two ways. First, our product construction has the extra ability to tackle timing constraints from the DTA. The extension is nontrivial since it needs to resolve the integration between randomness (from the PTA) and timing constraints (from the DTA), and still ensures the EXPTIME-completeness of the problem, matching the computational complexity in the discrete-time case [18]. Second, we have proved an undecidability result in the case of general nondeterministic timed automata, thus extending [18] with nondeterminism.

## Acknowledgements

This work has been supported by by the National Natural Science Foundation of China (Grants 61761136011, 61532019, 61472473). We also thank anonymous reviewers for detailed comments.

## References

[1] Baier, C., Katoen, J.: Principles of Model Checking. MIT Press (2008)
[2] Norman, G., Parker, D., Sproston, J.: Model checking for probabilistic timed automata. Formal Methods in System Design **43**(2) (2013) 164–190
[3] Beauquier, D.: On probabilistic timed automata. Theor. Comput. Sci. **292**(1) (2003) 65–84
[4] Kwiatkowska, M.Z., Norman, G., Segala, R., Sproston, J.: Automatic verification of real-time systems with discrete probability distributions. Theor. Comput. Sci. **282**(1) (2002) 101–150
[5] Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**(2) (1994) 183–235
[6] Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc. (1994)
[7] Jensen, H.E.: Model checking probabilistic real time systems. In: 7th Nordic Workshop on Programming Theory, Chalmers University of Technology (1996) 247–261
[8] André, É., Fribourg, L., Sproston, J.: An extension of the inverse method to probabilistic timed automata. Formal Methods in System Design **42**(2) (2013) 119–145
[9] Kwiatkowska, M.Z., Norman, G., Sproston, J., Wang, F.: Symbolic model checking for probabilistic timed automata. Inf. Comput. **205**(7) (2007) 1027–1077

[10] Kwiatkowska, M.Z., Norman, G., Parker, D.: Stochastic games for verification of probabilistic timed automata. In: FORMATS. (2009) 212–227
[11] Jovanovic, A., Kwiatkowska, M.Z., Norman, G.: Symbolic minimum expected time controller synthesis for probabilistic timed automata. In: FORMATS. (2015) 140–155
[12] Jurdzinski, M., Kwiatkowska, M.Z., Norman, G., Trivedi, A.: Concavely-priced probabilistic timed automata. In: CONCUR. (2009) 415–430
[13] Kwiatkowska, M.Z., Norman, G., Parker, D., Sproston, J.: Performance analysis of probabilistic timed automata using digital clocks. Formal Methods in System Design **29**(1) (2006) 33–78
[14] Berendsen, J., Chen, T., Jansen, D.N.: Undecidability of cost-bounded reachability in priced probabilistic timed automata. In: TAMC. (2009) 128–137
[15] Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: CAV. (2011) 585–591
[16] Laroussinie, F., Sproston, J.: State explosion in almost-sure probabilistic reachability. Inf. Process. Lett. **102**(6) (2007) 236–241
[17] Jurdzinski, M., Sproston, J., Laroussinie, F.: Model checking probabilistic timed automata with one or two clocks. LMCS **4**(3) (2008)
[18] Sproston, J.: Discrete-time verification and control for probabilistic rectangular hybrid automata. In: QEST. (2011) 79–88
[19] Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Systems **2**(4) (1990) 255–299
[20] Ouaknine, J., Worrell, J.: On the decidability of metric temporal logic. In: LICS. (2005) 188–197
[21] Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. J. ACM **43**(1) (1996) 116–146
[22] Donatelli, S., Haddad, S., Sproston, J.: Model checking timed and stochastic properties with CSL^{TA}. IEEE Trans. Software Eng. **35**(2) (2009) 224–240
[23] Fu, H., Li, Y., Li, J.: Verifying probabilistic timed automata against omega-regular dense-time properties. CoRR **abs/1712.00275** (2017)
[24] Chen, T., Han, T., Katoen, J., Mereacre, A.: Model checking of continuous-time Markov chains against timed automata specifications. Logical Methods in Computer Science **7**(1) (2011)
[25] Fu, H.: Approximating acceptance probabilities of CTMC-paths on multi-clock deterministic timed automata. In: HSCC. (2013) 323–332
[26] Laroussinie, F., Sproston, J.: State explosion in almost-sure probabilistic reachability. Inf. Process. Lett. **102**(6) (2007) 236–241
[27] Vardi, M.Y.: Probabilistic linear-time model checking: An overview of the automata-theoretic approach. In: AMAST. (1999) 265–276
[28] Brázdil, T., Krcál, J., Kretínský, J., Kucera, A., Rehák, V.: Measuring performance of continuous-time stochastic processes using timed automata. In: HSCC. (2011) 33–42
[29] Barbot, B., Chen, T., Han, T., Katoen, J., Mereacre, A.: Efficient CTMC model checking of linear real-time objectives. In: TACAS. (2011) 128–142
[30] Bortolussi, L., Lanciani, R.: Fluid model checking of timed properties. In: FORMATS. (2015) 172–188