Verifying Probabilistic Timed Automata Against Deterministic-Timed-Automata Specifications*

Extended Abstract†

Ben Trovato‡
Institute for Clarity in Documentation
Dublin, Ohio
trovato@corporation.com

G.K.M. Tobin§
Institute for Clarity in Documentation
Dublin, Ohio
webmaster@marysville-ohio.com

Lars Thørväld¶
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Valerie Béranger
Inria Paris-Rocquencourt
Rocquencourt, France

Aparna Patel
Rajiv Gandhi University
Doimukh, Arunachal Pradesh, India

Huifen Chan
Tsinghua University
Haidian Qu, Beijing Shi, China

Charles Palmer
Palmer Research Laboratories
San Antonio, Texas
cpalmer@prl.com

John Smith
The Thørväld Group
jsmith@affiliation.org

Julius P. Kumquat
The Kumquat Consortium
jpkumquat@consortium.net

ABSTRACT

Probabilistic timed automata (PTAs) are timed automata 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 paper, we study model checking of linear-time temporal properties over PTAs. In particular, we consider linear-time properties that can be encoded through deterministic timed automata (DTAs) with finite acceptance criterion. DTAs are a deterministic subclass of timed automata that can recognize a wide range of timed formal languages, thus can be effectively used to specify timed behaviours of systems. We show that through a product construction, model checking of PTAs against DTA-specifications can be reduced to solving reachability probabilities over PTAs, thus can be effectively solved by known algorithms on PTA-reachability. Our experimental results demonstrate the efficiency of our approach. As far as we know, we are the first to consider linear-time model checking of PTAs.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference Format:
Ben Trovato, G.K.M. Tobin, Lars Thørväld, Valerie Béranger, Aparna Patel, Huifen Chan, Charles Palmer, John Smith, and Julius P. Kumquat. 1997. Verifying Probabilistic Timed Automata Against Deterministic-Timed-Automata Specifications. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997 (WOODSTOCK'97),* 8 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Stochastic timed systems are systems that exhibit both timed and stochastic behaviours. Such systems play a dominant role in many real-world applications (cf. [3]), hence addressing fundamental issues such as safety and performance over these systems are important. *Probabilistic timed automata* (PTAs) [5, 20, 23] serve as a good mathematical model for these systems. They extend the well-known model of timed automata [1] (for nonprobabilistic timed systems) with discrete probability distributions, and Markov Decision Processes (MDPs) [24] (for untimed probabilistic systems) with timing constraints.

Formal verification of PTAs has received much attention in recent years [23]. 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*) [5, 13, 20]. Advanced techniques for branching-time model checking of PTAs such as inverse method and symbolic method have been explored in [2, 14, 17, 21]. Extension with *cost* or *reward*, resulting in *priced* PTAs, has also been investigated. On one hand, Berendsen *et al.* [6] proved that cost-bounded reachability probability is undecidable over priced PTAs. On the other hand, Jurdzinski *et al.* [15] and Kwiatkowska *et al.* [19] proved that several notions of accumulated (discounted) cost are computable over priced PTAs. Most verification algorithms for PTAs

*Produces the permission block, and copyright information
†The full version of the author's guide is available as `acmart.pdf` document
‡Dr. Trovato insisted his name be first.
§The secretary disavows any knowledge of this author's actions.
¶This author is the one who did all the really hard work.

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

have been implemented in the model checker PRISM [18]. Computational complexity of several verification problems for PTAs is studied in [15, 16, 22].

A shortcoming of existing verification approaches for PTAs is that they all considered branching-time properties. As far as we know, no results have ever considered linear-time model checking for PTAs. Linear-time temporal properties are however important as they can specify complex timed behaviours induced by e.g. finite sequences of timed events. In particular, we focus on linear-time temporal properties that can be encoded by deterministic timed automata (DTAs). DTA is the deterministic version of timed automata. Although DTA is weaker than general timed automata, it can recognize a wide class of formal timed languages, and express interesting properties which cannot be expressed in branching-time logics [11]. The problem to verify DTA-specifications over stochastic timed models has only been investigated for continuous-time Markov processes (cf. [4, 8–12]), a completely different formalism for stochastic timed systems which assigns probability distributions to time elapses. In this paper, we study verification of DTA-specifications over PTAs. As far as we know, we are the first to conduct this line of research.

Our Contributions. We show that through a product construction, the optimal probability of PTA-paths accepted by a DTA w.r.t the finite acceptance criterion can be computed exactly by known algorithms for reachability probabilities over PTAs. The novelty of our product construction is that to enable the DTA to keep track of the next location after a probabilistic jump in the PTA, one needs to integrate either the set of regions of the DTA or a local conjunction over the rules of the DTA. We demonstrate experimental results on several case studies and show that our approach is effective to analyse linear-time properties over PTAs.

2 PRELIMINARIES

In the whole paper, 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 an infinite word w, $\inf(w)$ is the set of symbols that occur infinitely many times in w.

2.1 Clock Valuations, Clock Constraints and Clock Equivalences

In this part, we fix a finite set $\mathcal{X}$ of *clocks*.

Clock Valuations. Let $\mathcal{X}$ be a finite set of *clocks*. A *clock valuation* is a function $\nu : \mathcal{X} \to [0, \infty)$. The set of clock valuations is denoted by $Val(\mathcal{X})$. Given a clock valuation ν, 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 $x \in \mathcal{X}$. Moreover, we denote by $\mathbf{0}$ the clock valuation such that $\mathbf{0}(x) = 0$ for all $x \in \mathcal{X}$. *Clock Constraints.* The set of *clock constraints* $CC(\mathcal{X})$ over $\mathcal{X}$ is generated by the following grammar:

$$\phi := \textbf{true} \mid x \le d \mid c \le x \mid x + c \le y + d \mid \neg\phi \mid \phi \wedge \phi$$

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

Clock Equivalence. Consider a nonnegative integer N which acts a threshold for relevant clock values: values held by clocks are treated the same if they exceed N. With such a fixed N, the standard notion of clock equivalence (see [1]) is an equivalence relation $\sim_N$ over $Val(\mathcal{X})$ as follows: for any two clock valuations ν, ν', $\nu \sim_N \nu'$ iff the following conditions hold:

- for all $x \in \mathcal{X}$, $\nu(x) > N$ iff $\nu'(x) > N$;
- for all $x \in \mathcal{X}$, if $\nu(x) \le N$ then (i) $\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$ and (ii) $frac(\nu(x)) > 0$ iff $frac(\nu'(x)) > 0$;
- for all $x, y \in \mathcal{X}$, if $\nu(x), \nu(y) \le N$ then $frac(\nu(x)) \bowtie frac(\nu(y))$ iff $frac(\nu'(x)) \bowtie frac(\nu'(y))$ for all $\bowtie \in \{<, =, >\}$.

Equivalence classes of $\sim_N$ are conventionally called *regions*. The equivalence class that contains a given clock valuation ν is conventionally denoted by $[\nu]_\sim$.

2.2 Probabilistic Timed Automata

To introduce the notion of probabilistic timed automata (PTAs), we first define the notion of discrete probability distributions.

Discrete Probability Distributions. 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 $supp(q) := \{z \in U \mid q(z) > 0\}$. The set of discrete probability distributions over U is denoted by $\mathcal{D}(U)$. For $u \in U$, let μ_u be the *point distribution at u* which assigns probability 1 to u.

Definition 2.1 (Probabilistic Timed Automata (PTAs) [23]). A *probabilistic timed automaton* (PTA) $\mathcal{C}$ is a tuple

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

where

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

W.l.o.g, we assume that both Act and AP is disjoint from $[0, \infty)$. Below we fix a PTA $\mathcal{C}$ in the form (1). The semantics of PTAs is as follows.

States and Transition Relation. A *state* of $\mathcal{C}$ is a pair (ℓ, ν) in $L \times Val(\mathcal{X})$ such that $\nu \models inv(\ell)$. 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 that

- $(\ell, \nu), (\ell', \nu')$ 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 $(s, a, s') \in \to$. We omit the subscript '$\mathcal{C}$' in '$S_{\mathcal{C}}$' if the underlying context is clear. 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 C is *well-formed* if for every state (ℓ, ν) and action $a \in Act$ such that $\nu \models enab(\ell, a)$ and for 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 rest of the paper, we always assume that the underlying PTA is well-formed. PTAs that are not well-formed can be repaired to satisfy the well-formedness condition [21].

Paths. A *finite path* ρ (under 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) $a_{2k} \in [0, \infty)$ for all integers $0 \leq k \leq \frac{n}{2}$, (iii) $a_{2k+1} \in Act$ for all integers $0 \leq k \leq \frac{n-1}{2}$ and (iv) for all $0 \leq k \leq n-1$, $s_k \xrightarrow{a_k} s_{k+1}$. The length of ρ is n, denoted by len(ρ) An *infinite path* (under 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 $n \in \mathbb{N}_0$, $\langle s_0, a_0, \ldots, a_{n-1}, s_n \rangle$ is a finite path. The set of finite (resp. infinite) paths under C is denoted by $Paths^*_C$ (resp. $Paths^\omega_C$).

Schedulers. A *scheduler* (or *adversary*) is a function σ 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) $\sigma(\rho) \in Act$ if n is odd, (ii) $\sigma(\rho) \in [0, \infty)$ if n is even, and (iii) there exists a state s' such that $s_n \xrightarrow{\sigma(\rho)} s'$. A finite path $\rho = s_0 a_0 \ldots s_n$ is said to *follow* a scheduler σ if for all $0 \leq m \leq n$, $a_m = \sigma(s_0 a_0 \ldots s_m)$. Likewise, an infinite path $s_0 a_0 s_1 a_1 \ldots$ *follows* a scheduler σ 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 σ is denoted by $Paths^*_{C,\sigma}$ (resp. $Paths^\omega_{C,\sigma}$). We note that the set $Paths^*_{C,\sigma}$ is countably-infinite from definition.

Probability Spaces under Schedulers. Let σ be any scheduler for C. The probability space for C w.r.t σ is defined as $(\Omega^{C,\sigma}, \mathcal{F}^{C,\sigma}, \mathbb{P}^{C,\sigma})$ where $\Omega^{C,\sigma} := Paths^\omega_{C,\sigma}$, $\mathcal{F}^{C,\sigma}$ is the smallest σ-algebra generated by all cylinder sets induced by finite paths (a finite path ρ induces the cylinder set $Cyl(\rho)$ of all infinite paths in $Paths^\omega_{C,\sigma}$ with ρ being their (common) prefix) and $\mathbb{P}^{C,\sigma}$ is the unique probability measure such that for all finite paths $\rho = s_0 a_0 \ldots a_{n-1} s_n$ in $Paths^*_{C,\sigma}$, $\mathbb{P}^{C,\sigma}(Cyl(\rho)) = \prod_{k=0}^{n-1} \mathbf{P}(s_k, \sigma(s_0 a_0 \ldots a_{k-1} s_k), s_{k+1})$. Intuitively, the probability space under σ is induced by a Markov chain where the state space is $Paths^*_{C,\sigma}$ and the one-step probability transition matrix is determined by $\mathbf{P}$ and σ.

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

Reachability. An infinite path $\pi = (\ell_0, \nu_0) a_0 (\ell_1, \nu_1) a_1 \ldots$ is said to *visit* a subset $U \subseteq L$ of locations *eventually* if there exists $n \in \mathbb{N}_0$ such that $\ell_n \in U$. The set of infinite paths in $Paths^\omega_{C,\sigma}$ that visit U eventually is denoted by $Reach^U_{C,\sigma}$. From the fact that the set $Paths^*_{C,\sigma}$ is countably-infinite, $Reach^U_{C,\sigma}$ is measurable since it is a countable union of cylinder sets.

2.3 Deterministic Timed Automata

Definition 2.2 (Timed Automata (TAs) [10–12]). A *timed automaton* (TA) $\mathcal{A}$ is a tuple

$$\mathcal{A} = (Q, \Sigma, \mathcal{X}, \Delta) \quad (2)$$

where

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

Definition 2.3 (Deterministic Timed Automata (DTAs) [10–12]). A TA $\mathcal{A} = (Q, \Sigma, \mathcal{X}, \Delta)$ is called *deterministic* iff

(1) (*determinism*): whenever $(q_1, b_1, \phi_1, X_1, q_1'), (q_2, b_2, \phi_2, X_2, q_2') \in \Delta$, 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 $\nu \models \phi$.

Below we fix a DTA $\mathcal{A}$ in the form (2). Given $q \in Q$, $\nu \in Val(\mathcal{X})$ and $b \in \Sigma$, the triple $(\Phi^\nu_{q,b}, \mathbf{X}^\nu_{q,b}, \mathbf{q}^\nu_{q,b}) \in CC(\mathcal{X}) \times 2^{\mathcal{X}} \times Q$ are determined such that $\left(q, b, \Phi^\nu_{q,b}, \mathbf{X}^\nu_{q,b}, \mathbf{q}^\nu_{q,b}\right) \in \Delta$ is the unique rule satisfying $\nu \models \Phi^\nu_{q,b}$. We illustrate the semantics of DTAs as follows.

Configurations and One-Step Transition Function. A *configuration* of $\mathcal{A}$ is a pair (q, ν), where $q \in Q$ and $\nu \in Val(\mathcal{X})$. The *one-step transition function*

$$\kappa : (Q \times Val(\mathcal{X})) \times (\Sigma \cup [0, \infty)) \to Q \times Val(\mathcal{X})$$

is defined by: $\kappa((q, \nu), a) := \left(\mathbf{q}^\nu_{q,a}, \nu\left[\mathbf{X}^\nu_{q,a} := 0\right]\right)$ for $a \in \Sigma$; $\kappa((q, \nu), a) := (q, \nu + a)$ for $a \in [0, \infty)$. For the sake of convenience, we write $(q, \nu) \stackrel{a}{\Rightarrow} (q', \nu')$ instead of $\kappa((q, \nu), a) = (q', \nu')$.

Infinite Time Words and Runs. An *infinite time word* is an infinite sequence $\{a_n\}_{n \in \mathbb{N}_0}$ such that $a_{2n} \in [0, \infty)$ and $a_{2n+1} \in \Sigma$ for all n.

The *run* of $\mathcal{A}$ on an infinite word $w = \{a_n\}_{n \in \mathbb{N}_0}$ with *initial configuration* (q, ν), denoted by $\mathcal{A}_{q,\nu}(w)$, is the unique infinite sequence $\{(q_n, \nu_n, a_n)\}_{n \in \mathbb{N}_0}$ which satisfies that $(q_0, \nu_0) = (q, \nu)$ and $(q_n, \nu_n) \stackrel{a_n}{\Longrightarrow} (q_{n+1}, \nu_{n+1})$ for all $n \in \mathbb{N}_0$. The trajectory of $\mathcal{A}_{q,\nu}(w)$, an infinite string over Q, is define as follow traj$(\mathcal{A}_{q,\nu}(w)) := q_0 q_1 \cdots$

Now we illustrate the acceptance condition for DTAs. In this paper, we focus on infinite acceptance condition. Finite case is trivial and we also support it in our tool.

Definition 2.4 (Rabin Acceptance Criterion). An Rabin acceptance condition is a a finite set of pairs $\mathcal{F} = \{(H_1, K_1), \ldots, (H_n, K_n)\}$, where H_i and K_i are subset of Q for all $i < n$. A set $Q' \subseteq Q$ is called Rabin accepting by $\mathcal{F}$ if there exists $1 \leq i \leq n$ such that $Q' \cap H_i = \emptyset$ and $Q' \cap K_i \neq \emptyset$. An infinite word w is accepted $\mathcal{A}$ with *initial* *configuration* (q, v) and acceptance $\mathcal{F}$ iff $\inf(\mathrm{traj}(\mathcal{A}_{(q,v)}(w)))$ is Rabin accepting by $\mathcal{F}$.

3 THE PTA-DTA PROBLEM

In this section, we define the problem of model-checking PTAs against DTA-specifications. The problem takes a PTA and a DTA as input, and computes the probability that infinite paths under the PTA are accepted by the DTA. Informally, the DTA encodes the linear-time property by judging whether an infinite path is accepted or not through the external behaviour of the path, thus the problem is to compute the probability that the external behaviour of PTA meets the criterion specified by the DTA. In practice, the DTA is often used to capture all good (or bad) behaviours, so the problem can be treated as a task to evaluated to what extent the PTA behaves in a good (or bad) way.

Below we fix a well-formed PTA C taking the form (1) and a DTA $\mathcal{A}$ taking the form (2) with the difference that the set of clocks for C (resp. for $\mathcal{A}$) is denoted by $\mathcal{X}_1$ (resp. $\mathcal{X}_2$). W.l.o.g., we assume that $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$ and $\Sigma = 2^{AP}$. We first show how an infinite path in $Paths_C^{\omega}$ can be interpreted as an infinite word.

Definition 3.1 (Infinite Paths as Infinite Words). Given an infinite path

$$\pi = (\ell_0, v_0)a_0(\ell_1, v_1)a_1(\ell_2, v_2)a_2 \dots a_{2n}(\ell_{2n+1}, v_{2n+1})a_{2n+1}(\ell_{2n+2}, v_{2n+2}) \dots$$

under C (note that $v_0 = \mathbf{0}$), the infinite word $\mathcal{L}(\pi)$ over $2^{AP} \cup [0, \infty)$ is defined as

$$\mathcal{L}(\pi) := a_0\mathcal{L}(\ell_2)a_2\mathcal{L}(\ell_4) \dots a_{2n}\mathcal{L}(\ell_{2n+2}) \dots \quad .$$

Recall that $a_{2n} \in [0, \infty)$ and $a_{2n+1} \in Act$.

Remark 1. *Informally, the interpretation in Definition 3.1 works by (i) dropping (a) the initial location ℓ_0, (b) all clock valuations v_n's, (c) all locations ℓ_{2n+1}'s following a time-elapse, (d) all internal actions a_{2n+1}'s of C and (ii) replacing every ℓ_{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 DTA.*

Remark 2. *Our interpretation is different from [10–12]. In the style from [10–12], an infinite path $(\ell_0, v_0)a_0(\ell_1, v_1)a_1 \dots$ is interpreted as $a_0\mathcal{L}(\ell_0)a_2\mathcal{L}(\ell_2) \dots$, reversing the locations and actions/time-elapses. In contrast, our interpretation follows a natural way that preserves the order of external events in an infinite path. This advantage allows one to specify DTAs (for linear-time properties) in a straightforward way.*

Based on Definition 3.1, we define the finite acceptance condition as follows. For an infinite path $\pi = (\ell_0, v_0)a_0(\ell_1, v_1)a_1 \dots$ under C, we denote by $init(\pi)$ the initial location ℓ_0.

Definition 3.2 (Path Acceptance). An infinite path π under C is *infinitely accepted* by $\mathcal{A}$ w.r.t initial configuration (q, v) and a Rabin acceptance condition $\mathcal{F}$ if the infinite word $\mathcal{L}(\pi)$ is accepted by $\mathcal{A}$ w.r.t $(\kappa((q, v), \mathcal{L}(init(\pi))), \mathbf{0})$ and $\mathcal{F}$.

In the definitions above, the initial location omitted in Definition 3.1 is preprocessed by specifying explicitly that the initial configuration is $(\kappa((q, v), \mathcal{L}(init(\pi))), \mathbf{0})$.

Now we define the notion of acceptance probabilities over infinite paths under C.

Definition 3.3 (Acceptance Probabilities). Let F be a Rain acceptance condition. The probability that C *observes* $\mathcal{A}$ under scheduler σ, initial mode $q \in Q$ and F, denoted by $\mathfrak{p}_{q,F}^{\sigma}$, is defined by:

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

where $AccPaths_{C,\sigma}^{\mathcal{A},q,\mathcal{F}}$ is paths in C that falls into the Rabin-accepted language of $\mathcal{A}$

$$AccPaths_{C,\sigma}^{\mathcal{A},q,\mathcal{F}} = \left\{\pi \in Paths_{C,\sigma}^{\omega} \mid \pi \text{ is accepted by } \mathcal{A} \text{ w.r.t. } (q, \mathbf{0}) \text{ and } \mathcal{F}\right\}$$

Again, from the fact that the set $Paths_{C,\sigma}^{*}$ is countably-infinite, $AccPaths_{C,\sigma}^{\mathcal{A},q,F}$ is measurable since it can be represent int the form of a countable intersect and countable union of some cylinder sets.

Now the PTA-DTA problem is as follows.

- **Input:** a well-formed PTA C, a DTA $\mathcal{A}$, an initial mode q and a subset $\mathcal{F}$ of modes;
- **Output:** $\inf_{\sigma} \mathfrak{p}_{q,\mathcal{F}}^{\sigma}$ and $\sup_{\sigma} \mathfrak{p}_{q,\mathcal{F}}^{\sigma}$, where σ ranges over all time-divergent schedulers.

4 THE PRODUCT CONSTRUCTION

In this section, we introduce the core part of our algorithms to solve the PTA-DTA problem. The core part is a product construction which given a PTA C and a DTA $\mathcal{A}$, output a PTA which preserves the probability of the set of infinite paths of C accepted by $\mathcal{A}$. Below we fix a well-formed PTA C in the form (1) and a DTA $\mathcal{A}$ in the form (2) with the difference that the set of clocks for C (resp. for $\mathcal{A}$) is denoted by $\mathcal{X}_1$ (resp. $\mathcal{X}_2$). W.l.o.g., we assume that $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$ and $\Sigma = 2^{AP}$. We let $\mathcal{G}$ be the set of regions w.r.t $\sim_N$, where N is the maximal integer appearing in the clock constraints of $\mathcal{A}$.

The Main Idea. The intuition of the product construction is to let $\mathcal{A}$ reads external actions of C while C evolves along the time axis. The major difficulty is that when 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 Δ) used for the transition. A naive solution is to integrate each single rule Δ into the enabling condition *enab* in C. However, this simple solution does not work since a single rule in Δ fixes the labelling of a location in C, while the probabilistic distribution given by *prob* can jump to locations with different labels. We solve this difficulty by integrating into the enabling condition *enab* enough information on clock valuations under $\mathcal{A}$ so that the rule used for the transition (in $\mathcal{A}$) is clear. In detail, we introduce two versions of the product construction, each having a computational advantage against the other.

Product Construction (First Version). The *product PTA* $C \otimes \mathcal{A}_q$ between C and $\mathcal{A}$ with initial mode q is defined as the PTA $\left(L_{\otimes}, \ell_{\otimes}^{*}, \mathcal{X}_{\otimes}, Act_{\otimes}, inv_{\otimes}\right.$ where:

- $L_{\otimes} := L \times Q$;
- $\ell_{\otimes}^{*} := (\ell^{*}, q^{\star})$ where $q^{\star}$ is the unique mode such that $\kappa((q, \mathbf{0}), \mathcal{L}(\ell^{*})) = (q^{\star}, \mathbf{0})$;
- $\mathcal{X}_{\otimes} := \mathcal{X}_1 \cup \mathcal{X}_2$;

- $Act_\otimes := Act \times \mathcal{G}$;
- $inv_\otimes(\ell, q) := inv(\ell)$ for all $(\ell, q) \in L_\otimes$;
- $enab_\otimes((\ell, q), (a, R)) := enab(\ell, a) \wedge \phi_R$ for all $(\ell, q) \in L_\otimes$, where ϕ_R is any clock constraint such that $[\![\phi_R]\!] = R$;
- $\mathcal{L}_\otimes(\ell, q) := \{q\}$ for all $(\ell, q) \in L_\otimes$
- $prob_\otimes$ is given by

$$prob_\otimes((\ell, q), (a, R))(Y, (\ell', q')) :=$$
$$\begin{cases} prob(\ell, a)(Y \cap \mathcal{X}_1, \ell') & \text{if } (q, \mathcal{L}(\ell'), \phi_R^{q, \mathcal{L}(\ell')}, Y \cap \mathcal{X}_2, q') \in \Delta \\ 0 & \text{otherwise} \end{cases} .$$

where $(q, \mathcal{L}(\ell'), \phi_R^{q, \mathcal{L}(\ell')}, Y \cap \mathcal{X}_2, q')$ is the unique rule such that for all $\nu \in R$, $\nu \in [\![\phi_R^{q, \mathcal{L}(\ell')}]\!]$. The uniqueness follows from determinism and totality of DTAs.

Apart from standard constructions (e.g., the Cartesian product between L and Q), the product construction also has Cartesian product between Act and $\mathcal{G}$. Then for each extended action (a, R), the enabling condition for this action is just the conjunction between $enab(\ell, a)$ and R. This is to ensure that when the action (a, R) is taken, the clock valuation under $\mathcal{A}$ lies in R. Finally in the definition for $prob_\otimes$, upon the action (a, R) and the target location ℓ', the DTA $\mathcal{A}$ chooses the unique rule $(q, \mathcal{L}(\ell'), \phi_R^{q, \mathcal{L}(\ell')}, Y \cap \mathcal{X}_2, q')$ and then jump to q' with reset set $Y \cap \mathcal{X}_2$. By integrating regions into the enabling condition, the DTA $\mathcal{A}$ can know the status of the clock valuation under $\mathcal{A}$ through its region, hence can decide which rule to use for the transition. This version of product construction works well if the number of regions is not large. We note that the number of regions only depends on N, not on the size of $\mathcal{A}$. In the following, we introduce another version which depends directly on the size of $\mathcal{A}$. The second version has an advantage when the number of regions is large.

Product Construction (Second Version). For each $q \in Q$, we let

$$\mathcal{T}_q := \{h : \Sigma \to CC(\mathcal{X}_2) \mid \forall b \in \Sigma.\ (q, b, h(b), X, q') \in \Delta \text{ for some } X, q'\} .$$

Intuitively, every element of $\mathcal{T}_q$ is a tuple of clock constraints $\{\phi_b\}_{b \in \Sigma}$, where each clock constraint ϕ_b is chosen from the rules emitting from q and b. The *product PTA* $C \otimes \mathcal{A}_q$ between C and $\mathcal{A}$ with initial mode q is defined almost the same as in the first version of the product construction, with the following differences:

- $Act_\otimes := Act \times \bigcup_q \mathcal{T}_q$;
- $enab_\otimes((\ell, q), (a, h)) := enab(\ell, a) \wedge \bigwedge_{b \in \Sigma} h(b)$ for all $(\ell, q) \in L_\otimes$ and $h \in \mathcal{T}_q$, and $enab_\otimes((\ell, q), (a, h)) :=$ **false** otherwise;
- $prob_\otimes$ is given by

$$prob_\otimes((\ell, q), (a, h))(Y, (\ell', q')) :=$$
$$\begin{cases} prob(\ell, a)(Y \cap \mathcal{X}_1, \ell') & \text{if } (q, \mathcal{L}(\ell'), h(\mathcal{L}(\ell')), Y \cap \mathcal{X}_2, q') \in \Delta \\ 0 & \text{otherwise} \end{cases} .$$

The intuition for the second version is that it is also possible to specify the information needed to identify the rule to be chosen by the DTA through a local conjunction of the rules emitting from a mode. For each mode, the local conjunction chooses one clock constraint from rules with the same symbol, and group them together through conjunction. From determinism and totality of DTAs, each conjunction constructed in this way determines which rule to use in the DTA for every symbol in a unique way. The advantage of the second version against the first one is that it is more suitable for DTAs with small size and large N (leading to a large number of reigons), as the size of the product PTA relies only the size of the DTA.

Remark 3. *It is easy to see that the PTA $C \otimes \mathcal{A}_q$ (in both versions) is well-formed as C is well-formed and the DTA $\mathcal{A}$ does not introduce extra invariant conditions.*

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

Transformation $\mathcal{T}$ From Paths under C into Paths under $C \otimes \mathcal{A}_q$. Since the two versions of product construction shares similarities, we illustrate the transformation in a unified fashion. The transformation is defined as the function $\mathcal{T} : Paths_C^* \cup Paths_C^\omega \to Paths_{C \otimes \mathcal{A}_q}^* \cup Paths_{C \otimes \mathcal{A}_q}^\omega$ which transform a finite or infinite path under C into one under $C \otimes \mathcal{A}_q$ as follows. For a finite path

$$\rho = (\ell_0, \nu_0) a_0 \dots a_{n-1} (\ell_n, \nu_n)$$

under C (note that $(\ell_0, \nu_0) = (\ell^*, \mathbf{0})$ by definition), we define $\mathcal{T}(\rho)$ to be the unique finite path

$$\mathcal{T}(\rho) := ((\ell_0, q_0), \nu_0 \cup \mu_0) a'_0 \dots a'_{n-1} ((\ell_n, q_n), \nu_n \cup \mu_n) \quad (3)$$

under $C \otimes \mathcal{A}_q$ such that (†)

- $\kappa((q, \mathbf{0}), \mathcal{L}(\ell^*)) = (q_0, \mu_0)$ (note that $\mu_0 = \mathbf{0}$), and
- for all $0 \le k < n$, if $a_k \in [0, \infty)$ then $a'_k = a_k$ and $(q_k, \mu_k) \xRightarrow{a_k} (q_{k+1}, \mu_{k+1})$, and
- for all $0 \le k < n$, 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 either (i) the first version of the product construction is taken and ξ_k is the region $[\mu_k]_\sim$ or (ii) the second version is taken and ξ_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 C, we define $\mathcal{T}(\pi)$ to be the unique infinite path

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

under $C \otimes \mathcal{A}_q$ such that the three conditions below (†) hold for all $k \in \mathbb{N}_0$ instead of all $0 \le k < n$. □

The following lemma shows that $\mathcal{T}$ is a bijection and preserves zenoness.

Lemma 4.1. *The function $\mathcal{T}$ is a bijection. Moreover, for any infinite path π, π is non-zeno iff $\mathcal{T}(\pi)$ is non-zeno.*

Proof. The first claim follows straightforwardly from the determinism and totality of DTAs. The second claim follows from the fact that $\mathcal{T}$ preserves time elapses in the transformation. □ □

We also show the relationship on schedulers before and after product construction.

Transformation θ From Schedulers under C into Schedulers under $C \otimes \mathcal{A}_q$. We define the function θ from the set of schedulers under C into the set of schedulers under $C \otimes \mathcal{A}_q$ as follows: for any scheduler σ for C, $\theta(\sigma)$ (for $C \otimes \mathcal{A}_q$) is defined such that for any finite path ρ under C where $\rho = (\ell_0, \nu_0)a_0 \dots a_{n-1}(\ell_n, \nu_n)$ and $\mathcal{T}(\rho)$ is given as in (3),

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

where $\lambda(\rho)$ is either $[\mu_n]_\sim$ if the first version of the product construction is taken, or the unique function such that for each symbol $b \in \Sigma$, $\lambda(\rho)(b)$ is the unique clock constraint appearing in a rule emitting from q_k and with symbol b such that $\mu_n \models \lambda(\rho)(b)$. Note that the well-definedness of θ follows from Lemma 4.1. □

By Lemma 4.1, the product construction and the determinism and totality of DTAs, one can prove straightforwardly the following lemma.

Lemma 4.2. *The function θ is a bijection.*

Now we show the relationship between infinite paths accepted by a DTA before product construction and infinite paths visiting certain target locations after product construction. Below we lift the function $\mathcal{T}$ to all subsets of paths in the standard fashion: for all subsets $A \subseteq Paths^*_C \cup Paths^\omega_C$, $\mathcal{T}(A) := \{\mathcal{T}(\omega) \mid \omega \in A\}$.

Definition 4.3 (Traces). Let $\mathcal{T}(\pi) = ((\ell_0, q_0), \nu_0 \cup \mu_0)a'_0((\ell_1, q_1), \nu_1 \cup \mu_1)a'_1 \dots$ the trace of $\mathcal{T}(\pi)$ is defined by $\text{trace}(\mathcal{T}(\pi)) := q_0 q_1 \dots$.

Verifying Limit Rabin Properties. Paths in $C \otimes \mathcal{A}_q$ that C is accepted by $\mathcal{A}$ with $\mathcal{F}$ is

$$RabinPaths^{\mathcal{F}}_{C\otimes\mathcal{A}_q,\sigma} = \left\{\pi \in Paths^\omega_{C\otimes\mathcal{A}_q,\sigma} \mid \inf(\text{trace}(\pi)) \text{ is Rabin accepting by } \mathcal{F}\right\}$$

and $RabinPaths^{\mathcal{F}}_{C\otimes\mathcal{A}_q,\sigma}$ is an limit LT Property.

Proposition 4.4. *For any scheduler σ, any initial mode q and any Rabin acceptance condition $\mathcal{F}$ on DTA $\mathcal{A}$, $\mathcal{T}\left(AccPaths^{\mathcal{A},q,\mathcal{F}}_{C,\sigma}\right) = RabinPaths^{\mathcal{F}}_{C\otimes\mathcal{A}_q,\theta(\sigma)}$.*

Proof. By definition we have

$$AccPaths^{\mathcal{A},q,\mathcal{F}}_{C,\sigma} = \left\{\pi \in Paths^\omega_{C,\sigma} \mid \inf(\text{traj}(\mathcal{A}_{(q^*,\mathbf{0})}(\mathcal{L}(\pi)))) \text{ is Rabin accepting by } \mathcal{F}\right\},$$

where $q^* = \kappa((q,\mathbf{0}), \mathcal{L}(init(\pi))$. Let $\pi = (\ell_0, \nu_0)a_0(\ell_1, \nu_1)a_1 \dots$ be any infinite path. And by definition of $\mathcal{T}$ we have

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

$$\mathcal{A}_{(q^*,\mathbf{0})}(\mathcal{L}(\pi)) = \{(q_n, \mu_n, \mathcal{L}(\pi)_n)\}_{n\in\mathbb{N}_0}.$$

Then it's obvious that

$$\text{trace}(\mathcal{T}(\pi)) = q_0 q_1 \cdots = \text{traj}(\mathcal{A}_{(q^*,\mathbf{0})}(\mathcal{L}(\pi))).$$

Then we conclude that $\inf(\text{trace}(\mathcal{T}(\pi)))$ is Rabin accepting by $\mathcal{F}$ iff $\inf(\text{traj}(\mathcal{A}_{(q^*,\mathbf{0})}(\mathcal{L}(\pi))))$ is Rabin accepting by $\mathcal{F}$. □

Finally, we demonstrate the relationship between acceptance probabilities before product construction and reachability probabilities after product construction. We also clarify the probability of zenoness before and after the product construction.

Theorem 4.5. *For any scheduler σ, initial mode q and Rabin acceptance $\mathcal{F}$,*

$$\mathfrak{p}^\sigma_{q,\mathcal{F}} = \mathbb{P}^{C,\sigma}\left(AccPaths^{\mathcal{A},q,\mathcal{F}}_{C,\sigma}\right) = \mathbb{P}^{C\otimes\mathcal{A}_q,\theta(\sigma)}\left(RabinPaths^{\mathcal{F}}_{C\otimes\mathcal{A}_q,\theta(\sigma)}\right).$$

Moreover, $\mathbb{P}^{C,\sigma}(\{\pi \mid \pi \text{ is zeno}\}) = \mathbb{P}^{C\otimes\mathcal{A}_q,\theta(\sigma)}(\{\pi' \mid \pi' \text{ is zeno}\})$
.

Proof. Define the probability measure $\mathbb{P}'$ by: $\mathbb{P}'(A) = \mathbb{P}^{C\otimes\mathcal{A}_q,\theta(\sigma)}(\mathcal{T}(A))$ for $A \in \mathcal{F}^{C,\sigma}$. We show that $\mathbb{P}' = \mathbb{P}^{C,\sigma}$. By [7, Theorem 3.3], it suffices to consider cylinder sets as they form a pi-system (cf. [7, Page 43]). Let $\rho = (\ell_0, \nu_0)a_0 \dots a_{n-1}(\ell_n, \nu_n)$ be any finite path under C. By definition, we have that

$$\mathbb{P}^{C,\sigma}(Cyl(\rho)) = \mathbb{P}^{C\otimes\mathcal{A}_q,\theta(\sigma)}(Cyl(\mathcal{T}(\rho))) = \mathbb{P}^{C\otimes\mathcal{A}_q,\theta(\sigma)}(\mathcal{T}(Cyl(\rho))) = \mathbb{P}'(Cyl(\rho$$

The first equality comes from the fact that both versions of product construction preserves transition probabilities. The second equality is due to $Cyl(\mathcal{T}(\rho)) = \mathcal{T}(Cyl(\rho))$. The final equality follows from the definition. Hence $\mathbb{P}^{C,\sigma} = \mathbb{P}'$. Then the first claim follows from Proposition 4.4 and the second claim follows from Lemma 4.1. □
□

Note that a side result from Theorem 4.5 says that θ preserves time-divergence for schedulers before and after product construction. From Theorem 4.5 and Lemma 4.2, one immediately obtains the following result which transforms the PTA-DTA problem into computing reachability probabilities under the product PTA.

Corollary 4.6. *([26]) For any initial mode q and any Rabin acceptance condition $\mathcal{F}$, there exists an $\mathcal{F}_* \subseteq L_\otimes$ s.t. $\mathcal{F}_*$ is a union of several maximal end components that satisfy $\mathcal{F}$.*

$$\begin{aligned} opt_\sigma \mathfrak{p}^\sigma_{q,\mathcal{F}} &= opt_{\sigma'} \mathbb{P}^{C\otimes\mathcal{A}_q,\sigma'}\left(RabinPaths^{\mathcal{F}}_{C\otimes\mathcal{A}_q,\sigma'}\right) \\ &= opt_{\sigma''} \mathbb{P}^{C\otimes\mathcal{A}_q,\sigma''}\left(Reach^{\mathcal{F}_*}_{C\otimes\mathcal{A}_q,\sigma''}\right) \\ &= opt_{\sigma'''} \mathbb{P}^{C\otimes\mathcal{A}_q,\sigma'''}\left(Reach^{\mathcal{F}_*}_{C\otimes\mathcal{A}_q,\sigma'''}\right), \end{aligned}$$

where opt refers to either inf *(infimum) or* sup *(supremum), σ (resp. σ') range over all time-divergent schedulers for C (resp. $C\otimes\mathcal{A}_q$) and σ'' (resp. σ''') range over all time-divergent (resp. all time-divergent and time-convergent) schedulers for $C\otimes\mathcal{A}_q$. $\mathcal{F}_*$ can be resolved by an MEC algorithm and $opt_\sigma \mathfrak{p}^\sigma_{q,\mathcal{F}}$ can be calculated by a reachability analysis.*

The way [26] discards time-convergent path is making a copy of every location in PTA model and enforcing a transition from the original one to the copy happen when 1 time unit is passed. After transiting to the copy, A transition back to the original one will immediately happend with no delay. And we put a label *tick* in copy. We only deal with paths that satisfy $\square\lozenge tick$ (i.e. *tick* is satisfied infinitely many times).

Using a standard MEC algorithm, we can find all MECs satisfy the corresponding property of an Rabin acceptance condition. In order to guarantee time-divergence, we only pick up MECs with at least one location that has an *tick* label and let $\mathcal{F}_*$ be the union of those MECs.

5 UNDECIDABILITY OF PTA-NTA PROBLEM

Lemma 5.1. *([27]) Given a timed automaton over an alphabet Σ, the problem of deciding whether it accepts all timed words over Σ is undecidable.*

Proposition 5.2. *Given a Nonedeterministic timed automaton over an alphabet Σ, the problem of deciding the minimal probability that it accepts a PTA w.r.t. $(q_{start}, \mathbf{0})$ is undecidable.*

Proof. We reach our goal by reducing the NTA universality problem to this problem. For any NTA $\mathcal{A} = (Q, \Sigma, \mathcal{X}, \Delta)$, let $\Sigma = \{b_1, b_2, \cdots, b_k\}$.

we construct an $\mathcal{A}' = (Q', \Sigma', \mathcal{X}, \Delta')$ where

$$Q' = Q \cup \{q_{init}\}, \Sigma' = \Sigma \cup \{b_0\}, \Delta' = \Delta \cup \{\langle q_{init}, b_0, \mathbf{true}, \mathcal{X}, q_{start} \rangle\}.$$

Let PTA $C = (L, \ell^*, \mathcal{X}, Act, inv, enab, prob, \mathcal{L})$ where

$$L := \Sigma',$$
$$\ell^* := b_0,$$
$$\mathcal{X} := \emptyset,$$
$$Act := \Sigma,$$
$$inv(b_i) := \mathbf{true}, \text{ for all } b_i \in L,$$
$$enab(b_i, b_j) := \mathbf{true}, \text{ for all } b_i \in L \text{ and all } b_j \in Act,$$
$$prob(b_i, b_j) := \mu_{(\emptyset, b_j)}, \text{ for all } b_i \in L \text{ and all } b_j \in Act,$$
$$\mathcal{L}(b_i) := b_i, \text{ for all } b_i \in L.$$

It is natural to see, for any time word $w = \alpha_0\alpha_1\alpha_2\cdots$ there is a scheduler $\sigma_w(\rho) := a_{len(\rho)}$ such that $\mathbb{P}^{C,\sigma_w}\left(AccPaths^{\mathcal{A}', q_{init}, \mathcal{F}}_{C, \sigma_w}\right) = 1$ iff $\mathcal{A}$ accepts w w.r.t. $(q_{start}, \mathbf{0})$ and $\mathbb{P}^{C,\sigma_w}\left(AccPaths^{\mathcal{A}', q_{init}, \mathcal{F}}_{C, \sigma_w}\right) = 0$ iff $\mathcal{A}$ rejects w .

Then we have $\inf_\sigma \mathbb{P}^{C,\sigma}\left(AccPaths^{\mathcal{A}', q_{init}, \mathcal{F}}_{C, \sigma}\right) = 1$ iff $\mathcal{A}$ accepts all timewords w.r.t. $(q_{start}, \mathbf{0})$. □

6 CONCLUSION AND FUTURE WORK

In this paper, we studied the linear-time model-checking problem PTA-DTA of DTA-specifications over PTAs. To solve the problem, we gave two versions of product construction (between a PTA and a DTA) which both reduce the problem to computing reachability probabilities over PTAs, for which efficient algorithms exist [20, 23]. Both the product constructions are nontrivial and are elaborated so that one caters for DTAs with a small number of regions and the other for DTAs with small size. Then we demonstrated two case studies clarifying that the problem PTA-DTA can be applied to real-world applications. Experimental results show that our product construction is efficient to solve the problem. A challenging future work is to study the PTA-DTA problem with general timed-automata specifications. Another future work is to integrate costs or rewards into this problem.

Acknowledgements. We thank Arnd Hartmanns for valuable suggestions on probabilistic timed automata.

REFERENCES

[1] Rajeev Alur and David L. Dill. 1994. A Theory of Timed Automata. *Theor. Comput. Sci.* 126, 2 (1994), 183–235. https://doi.org/10.1016/0304-3975(94)90010-8

[2] Étienne André, Laurent Fribourg, and Jeremy Sproston. 2013. An extension of the inverse method to probabilistic timed automata. *Formal Methods in System Design* 42, 2 (2013), 119–145. https://doi.org/10.1007/s10703-012-0169-x

[3] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking.* MIT Press.

[4] Benoît Barbot, Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. 2011. Efficient CTMC Model Checking of Linear Real-Time Objectives. In *Tools and Algorithms for the Construction and Analysis of Systems - 17th International Conference, TACAS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings (Lecture Notes in Computer Science),* Parosh Aziz Abdulla and K. Rustan M. Leino (Eds.), Vol. 6605. Springer, 128–142. https://doi.org/10.1007/978-3-642-19835-9_12

[5] Danièle Beauquier. 2003. On probabilistic timed automata. *Theor. Comput. Sci.* 292, 1 (2003), 65–84. https://doi.org/10.1016/S0304-3975(01)00215-8

[6] Jasper Berendsen, Taolue Chen, and David N. Jansen. 2009. Undecidability of Cost-Bounded Reachability in Priced Probabilistic Timed Automata. In *Theory and Applications of Models of Computation, 6th Annual Conference, TAMC 2009, Changsha, China, May 18-22, 2009. Proceedings (Lecture Notes in Computer Science),* Jianer Chen and S. Barry Cooper (Eds.), Vol. 5532. Springer, 128–137. https://doi.org/10.1007/978-3-642-02017-9_16

[7] Patrick Billingsley. 2012. *Probability and Measure* (anniversary edition ed.). Wiley.

[8] Luca Bortolussi and Roberta Lanciani. 2015. Fluid Model Checking of Timed Properties, See [25], 172–188. https://doi.org/10.1007/978-3-319-22975-1_12

[9] Tomás Brázdil, Jan Krcál, Jan Kretínský, Antonín Kucera, and Vojtech Rehák. 2011. Measuring performance of continuous-time stochastic processes using timed automata. In *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011,* Marco Caccamo, Emilio Frazzoli, and Radu Grosu (Eds.). ACM, 33–42. https://doi.org/10.1145/1967701.1967709

[10] Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. 2011. Model Checking of Continuous-Time Markov Chains Against Timed Automata Specifications. *Logical Methods in Computer Science* 7, 1 (2011). https://doi.org/10.2168/LMCS-7(1:12)2011

[11] Susanna Donatelli, Serge Haddad, and Jeremy Sproston. 2009. Model Checking Timed and Stochastic Properties with CSL^{TA}. *IEEE Trans. Software Eng.* 35, 2 (2009), 224–240. https://doi.org/10.1109/TSE.2008.108

[12] Hongfei Fu. 2013. Approximating acceptance probabilities of CTMC-paths on multi-clock deterministic timed automata. In *Proceedings of the 16th international conference on Hybrid systems: computation and control, HSCC 2013, April 8-11, 2013, Philadelphia, PA, USA,* Calin Belta and Franjo Ivancic (Eds.). ACM, 323–332. https://doi.org/10.1145/2461328.2461376

[13] Henrik Ejersbo Jensen. 1996. Model Checking Probabilistic Real Time Systems. In *7th Nordic Workshop on Programming Theory.* Chalmers University of Technology, 247–261. Report 86.

[14] Aleksandra Jovanovic, Marta Z. Kwiatkowska, and Gethin Norman. 2015. Symbolic Minimum Expected Time Controller Synthesis for Probabilistic Timed Automata, See [25], 140–155. https://doi.org/10.1007/978-3-319-22975-1_10

[15] Marcin Jurdzinski, Marta Z. Kwiatkowska, Gethin Norman, and Ashutosh Trivedi. 2009. Concavely-Priced Probabilistic Timed Automata. In *CONCUR 2009 - Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings (Lecture Notes in Computer Science),* Mario Bravetti and Gianluigi Zavattaro (Eds.), Vol. 5710. Springer, 415–430. https://doi.org/10.1007/978-3-642-04081-8_28

[16] Marcin Jurdzinski, Jeremy Sproston, and François Laroussinie. 2008. Model Checking Probabilistic Timed Automata with One or Two Clocks. *Logical Methods in Computer Science* 4, 3 (2008). https://doi.org/10.2168/LMCS-4(3:12)2008

[17] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. 2009. Stochastic Games for Verification of Probabilistic Timed Automata. In *Formal Modeling and Analysis of Timed Systems, 7th International Conference, FORMATS 2009, Budapest, Hungary, September 14-16, 2009. Proceedings (Lecture Notes in Computer Science),* Joël Ouaknine and Frits W. Vaandrager (Eds.), Vol. 5813. Springer, 212–227. https://doi.org/10.1007/978-3-642-04368-0_17

[18] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. 2011. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings (Lecture Notes in Computer Science),* Ganesh Gopalakrishnan and Shaz Qadeer (Eds.), Vol. 6806. Springer, 585–591. https://doi.org/10.1007/978-3-642-22110-1_47

[19] Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Jeremy Sproston. 2006. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design* 29, 1 (2006), 33–78. https://doi.org/10.1007/s10703-006-0005-2

[20] Marta Z. Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. 2002. Automatic verification of real-time systems with discrete probability distributions. *Theor. Comput. Sci.* 282, 1 (2002), 101–150. https://doi.org/10.1016/S0304-3975(01)00046-9

[21] Marta Z. Kwiatkowska, Gethin Norman, Jeremy Sproston, and Fuzhi Wang. 2007. Symbolic model checking for probabilistic timed automata. *Inf. Comput.* 205, 7 (2007), 1027–1077. https://doi.org/10.1016/j.ic.2007.01.004

[22] François Laroussinie and Jeremy Sproston. 2007. State explosion in almost-sure probabilistic reachability. *Inf. Process. Lett.* 102, 6 (2007), 236–241. https://doi.org/10.1016/j.ipl.2007.01.003

[23] Gethin Norman, David Parker, and Jeremy Sproston. 2013. Model checking for probabilistic timed automata. *Formal Methods in System Design* 43, 2 (2013), 164–190. https://doi.org/10.1007/s10703-012-0177-x

[24] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc.

[25] Sriram Sankaranarayanan and Enrico Vicario (Eds.). 2015. *Formal Modeling and Analysis of Timed Systems - 13th International Conference, FORMATS 2015, Madrid, Spain, September 2-4, 2015, Proceedings.* Lecture Notes in Computer Science,

Vol. 9268. Springer. https://doi.org/10.1007/978-3-319-22975-1

[26] Jeremy Sproston. 2011. Discrete-Time Verification and Control for Probabilistic Rectangular Hybrid Automata. In *Eighth International Conference on Quantitative Evaluation of Systems, QEST 2011, Aachen, Germany, 5-8 September, 2011*. 79–88. https://doi.org/10.1109/QEST.2011.18

[27] Frits W. Vaandrager. 1997. A Theory of Testing for Timed Automata (Abstract). In *TAPSOFT'97: Theory and Practice of Software Development, 7th International Joint Conference CAAP/FASE, Lille, France, April 14-18, 1997, Proceedings*. 39. https://doi.org/10.1007/BFb0030587