MODEL CHECKING PROBABILISTIC TIMED AUTOMATA WITH ONE OR TWO CLOCKS

MARCIN JURDZIŃSKI[a], FRANÇOIS LAROUSSINIE[b], AND JEREMY SPROSTON[c]

[a] Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK
   e-mail address: mju@dcs.warwick.ac.uk

[b] LIAFA, Université Paris 7 & CNRS, France
   e-mail address: francoisl@liafa.jussieu.fr

[c] Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy
   e-mail address: sproston@di.unito.it

ABSTRACT. Probabilistic timed automata are an extension of timed automata with discrete probability distributions. We consider model-checking algorithms for the subclasses of probabilistic timed automata which have one or two clocks. Firstly, we show that PCTL probabilistic model-checking problems (such as determining whether a set of target states can be reached with probability at least 0.99 regardless of how nondeterminism is resolved) are PTIME-complete for one-clock probabilistic timed automata, and are EXPTIME-complete for probabilistic timed automata with two clocks. Secondly, we show that, for one-clock probabilistic timed automata, the model-checking problem for the probabilistic timed temporal logic PTCTL is EXPTIME-complete. However, the model-checking problem for the subclass of PTCTL which does not permit both punctual timing bounds, which require the occurrence of an event at an exact time point, and comparisons with probability bounds other than 0 or 1, is PTIME-complete for one-clock probabilistic timed automata.

1. INTRODUCTION

Model checking is an automatic method for guaranteeing that a mathematical model of a system satisfies a formally-described property [CGP99]. Many real-life systems, such as multimedia equipment, communication protocols, networks and fault-tolerant systems, exhibit *probabilistic* behaviour. This leads to the study of model checking of probabilistic models based on Markov chains or Markov decision processes [Var85, HJ94, CY95, BdA95,

Table 1: Complexity results for model checking probabilistic timed automata

	One clock	Two clocks
Reachability, PCTL	P-complete	EXPTIME-complete
PTCTL$^{0/1}[\leq, \geq]$	P-complete	EXPTIME-complete
PTCTL$^{0/1}$	EXPTIME-complete	EXPTIME-complete
PTCTL$[\leq, \geq]$	P-hard, in EXPTIME	EXPTIME-complete
PTCTL	EXPTIME-complete	EXPTIME-complete

dA97a, BK98]. Similarly, it is common to observe complex *real-time* behaviour in systems. Model checking of (non-probabilistic) continuous-time systems against properties of timed temporal logics, which can refer to the time elapsed along system behaviours, has been studied extensively in, for example, the context of *timed automata* [ACD93, AD94], which are automata extended with *clocks* that progress synchronously with time. Finally, certain systems exhibit both probabilistic *and* timed behaviour, leading to the development of model-checking algorithms for such systems [ACD91, HJ94, dA97a, KNSS02, BHHK03, LS05, AB06, BCH$^+$07, DHS07].

In this paper, we aim to study model-checking algorithms for *probabilistic timed automata* [Jen96, KNSS02], which can be regarded as a variant of timed automata extended with discrete probability distributions, or (equivalently) Markov decision processes extended with clocks. Probabilistic timed automata have been used to model systems such as the IEEE 1394 root contention protocol, the backoff procedure in the IEEE 802.11 Wireless LANs, and the IPv4 link local address resolution protocol [KNPS06]. The temporal logic that we use to describe properties of probabilistic timed automata is PTCTL (Probabilistic Timed Computation Tree Logic) [KNSS02]. The logic PTCTL includes operators that can refer to bounds on exact time and on the probability of the occurrence of events. For example, the property "a request is followed by a response within 5 time units with probability 0.99 or greater" can be expressed by the PTCTL property $request \Rightarrow \mathbb{P}_{\geq 0.99}(\mathsf{F}_{\leq 5} response)$. The logic PTCTL extends the probabilistic temporal logic PCTL [HJ94, BdA95], and the real-time temporal logic TCTL [ACD93].

In the non-probabilistic setting, timed automata with one clock have recently been studied extensively [LMS04, LW05, ADOW05]. In this paper we consider the subclasses of probabilistic timed automata with one or two clocks. While probabilistic timed automata with a restricted number of clocks are less expressive than their counterparts with an arbitrary number of clocks, they can be used to model systems with simple timing constraints, such as probabilistic systems in which the time of a transition depends only on the time elapsed since the last transition. Conversely, one-clock probabilistic timed automata are more natural and expressive than Markov decision processes in which durations are associated with transitions (for example, in [dA97b, LS05]). We note that the IEEE 802.11 Wireless LAN case study has two clocks [KNPS06], and that an abstract model of the IEEE 1394 root contention protocol can be obtained with one clock [Sto02].

After introducing probabilistic timed automata and PTCTL in Section 2 and Section 3, respectively, in Section 4 we show that model-checking properties of PCTL, such as the property $\mathbb{P}_{\geq 0.99}(\mathsf{F}\, target)$ ("a set of target states is reached with probability at least 0.99 regardless of how nondeterminism is resolved"), is PTIME-complete for one clock probabilistic timed automata, which is the same complexity as for probabilistic reachability

properties on (untimed) Markov decision processes [PT87]. We also show that, in general, model checking of PTCTL on one clock probabilistic timed automata is EXPTIME-complete. However, inspired by the efficient algorithms obtained for non-probabilistic one clock timed automata [LMS04], we also show that, restricting the syntax of PTCTL to the sub-logic in which (1) punctual timing bounds and (2) comparisons with probability bounds other than 0 or 1, are disallowed, results in a PTIME-complete model-checking problem. In Section 5, we show that reachability properties with probability bounds of 0 or 1 are EXPTIME-complete for probabilistic timed automata with two or more clocks, implying EXPTIME-completeness of all the model-checking problems that we consider for this class of models. Our complexity results are summarized in Table 1, where 0/1 denotes the sub-logics of PTCTL with probability bounds of 0 and 1 only, and $[\leq, \geq]$ denotes the sub-logics of PTCTL in which punctual timing bounds are disallowed. The EXPTIME-hardness results are based on the concept of *countdown games*, which are two-player games operating in discrete time in which one player wins if it is able to make a state transition after *exactly* c time units have elapsed, regardless of the strategy of the other player. We show that the problem of deciding the winning player in countdown games is EXPTIME-complete. We believe that countdown games are of independent interest, and note that they have been used to show EXPTIME-hardness of model checking punctual timing properties of timed concurrent game structures [LMO06]. Finally, in Section 6, we consider the application of the forward reachability algorithm of Kwiatkowska et al. [KNSS02] to one-clock probabilistic timed automata, and show that the algorithm computes the exact probability of reaching a certain state set. This result is in contrast to the case of probabilistic timed automata with an arbitrary number of clocks, for which the application of the forward reachability algorithm results in an upper bound on the maximal probability of reaching a state set, rather than in the exact maximal probability. Note that, throughout the paper, we restrict our attention to probabilistic timed automata in which positive durations elapse in all loops of the system.

2. Probabilistic Timed Automata

2.1. **Preliminaries.** We use $\mathbb{R}_{\geq 0}$ to denote the set of non-negative real numbers, $\mathbb{Q}$ to denote the set of rational numbers, $\mathbb{N}$ to denote the set of natural numbers, and AP to denote a set of atomic propositions. A (discrete) probability *distribution* over a countable set Q is a function $\mu : Q \to [0, 1]$ such that $\sum_{q \in Q} \mu(q) = 1$. For a function $\mu : Q \to \mathbb{R}_{\geq 0}$ we define $\mathsf{support}(\mu) = \{q \in Q \mid \mu(q) > 0\}$. Then for an uncountable set Q we define $\mathsf{Dist}(Q)$ to be the set of functions $\mu : Q \to [0, 1]$, such that $\mathsf{support}(\mu)$ is a countable set and μ restricted to $\mathsf{support}(\mu)$ is a (discrete) probability distribution. In this paper, we make the additional assumption that distributions assign rational probabilities only; that is, for each $\mu \in \mathsf{Dist}(Q)$ and $q \in Q$, we have $\mu(q) \in [0, 1] \cap \mathbb{Q}$.

We now introduce *timed Markov decision processes*, which are Markov decision processes in which rewards associated with transitions are interpreted as time durations.

Definition 2.1. A *timed Markov decision process* (TMDP) $\mathsf{T} = (S, \bar{s}, \to, lab)$ comprises the following components:
- A (possibly uncountable) set of *states* S with an initial state $\bar{s} \in S$.

- A (possibly uncountable) *timed probabilistic, nondeterministic transition relation* $\rightarrow$ $\subseteq S \times \mathbb{R}_{\geq 0} \times \mathsf{Dist}(S)$ such that, for each state $s \in S$, there exists at least one tuple $(s, \_, \_) \in \rightarrow$.
- A *labelling function* $lab : S \rightarrow 2^{AP}$.

The transitions from state to state of a TMDP are performed in two steps: given that the current state is s, the first step concerns a nondeterministic selection of $(s, d, \nu) \in \rightarrow$, where d corresponds to the duration of the transition; the second step comprises a probabilistic choice, made according to the distribution ν, as to which state to make the transition to (that is, we make a transition to a state $s' \in S$ with probability $\nu(s')$). We often denote such a completed transition by $s \xrightarrow{d,\nu} s'$.

An *infinite path* of the TMDP T is an infinite sequence of transitions $\omega = s_0 \xrightarrow{d_0,\nu_0} s_1 \xrightarrow{d_1,\nu_1} \cdots$ such that the target state of one transition is the source state of the next. Similarly, a *finite path* of T is a finite sequence of consecutive transitions $\omega = s_0 \xrightarrow{d_0,\nu_0} s_1 \xrightarrow{d_1,\nu_1} \cdots \xrightarrow{d_{n-1},\nu_{n-1}} s_n$. The length of ω, denoted by $|\omega|$, is n (the number of transitions along ω). We use $Path_{ful}$ to denote the set of infinite paths of T, and $Path_{fin}$ the set of finite paths of T. If ω is a finite path, we denote by $last(\omega)$ the last state of ω. For any path ω and $i \leq |\omega|$, let $\omega(i) = s_i$ be the $(i+1)$th state along ω. Let $Path_{ful}(s)$ and $Path_{fin}(s)$ refer to the sets of infinite and finite paths, respectively, commencing in state $s \in S$.

In contrast to a path, which corresponds to a resolution of nondeterministic and probabilistic choice, an *adversary* represents a resolution of nondeterminism *only*. Formally, an adversary of a TMDP T is a function A mapping every finite path $\omega \in Path_{fin}$ to a transition $(last(\omega), d, \nu) \in \rightarrow$. Let Adv_{T} be the set of adversaries of T (when the context is clear, we write simply Adv). For any adversary $A \in Adv$, let $Path_{ful}^A$ and $Path_{fin}^A$ denote the sets of infinite and finite paths, respectively, resulting from the choices of distributions of A, and, for a state $s \in S$, let $Path_{ful}^A(s) = Path_{ful}^A \cap Path_{ful}(s)$ and $Path_{fin}^A(s) = Path_{fin}^A \cap Path_{fin}(s)$. Note that, by defining adversaries as functions from finite paths, we permit adversaries to be dependent on the history of the system. Hence, the choice made by an adversary at a certain point in system execution can depend on the sequence of states visited, the nondeterministic choices taken, and the time elapsed from each state, up to that point.

Given an adversary $A \in Adv$ and a state $s \in S$, we define the probability measure $Prob_s^A$ over $Path_{ful}^A(s)$ in the following way. We first define the function $\mathbb{A} : Path_{fin}^A(s) \times Path_{fin}^A(s) \rightarrow [0, 1]$. For two finite paths $\omega_{fin}, \omega'_{fin} \in Path_{fin}^A(s)$, let:

$$\mathbb{A}(\omega_{fin}, \omega'_{fin}) = \begin{cases} \mu(s') & \text{if } \omega'_{fin} \text{ is of the form } \omega_{fin} \xrightarrow{d,\mu} s' \text{ and } A(\omega_{fin}) = (d, \mu) \\ 0 & \text{otherwise.} \end{cases}$$

Next, for any finite path $\omega_{fin} \in Path_{fin}^A(s)$ such that $|\omega_{fin}| = n$, we define the probability $\mathbf{P}_s^A(\omega_{fin})$ as follows:

$$\mathbf{P}_s^A(\omega_{fin}) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } n = 0 \\ \mathbb{A}(\omega_{fin}(0), \omega_{fin}(1)) \cdot \ldots \cdot \mathbb{A}(\omega_{fin}(n-1), \omega_{fin}(n)) & \text{otherwise.} \end{cases}$$

Then we define the *cylinder* of a finite path ω_{fin} as:

$$cyl^A(\omega_{fin}) \stackrel{\text{def}}{=} \{\omega \in Path_{ful}^A(s) \mid \omega_{fin} \text{ is a prefix of } \omega\},$$

and let Σ_s^A be the smallest sigma-algebra on $Path_{ful}^A(s)$ which contains the cylinders $cyl^A(\omega_{fin})$ for $\omega_{fin} \in Path_{fin}^A(s)$. Finally, we define $Prob_s^A$ on Σ_s^A as the unique measure such that $Prob_s^A(cyl(\omega_{fin})) = \mathbf{P}_s^A(\omega_{fin})$ for all $\omega_{fin} \in Path_{fin}^A(s)$.

An *untimed Markov decision process* (MDP) $(S, \bar{s}, \rightarrow, lab)$ is defined as a finite-state TMDP, but for which $\rightarrow \subseteq S \times \mathsf{Dist}(S)$ (that is, the transition relation $\rightarrow$ does not contain timing information). Paths, adversaries and probability measures can be defined for untimed MDPs in the standard way (see, for example, [BK98]).

In the remainder of the paper, we distinguish between the following classes of TMDP.

- *Discrete TMDPs* are TMDPs in which (1) the state space S is finite, and (2) the transition relation $\rightarrow$ is finite and of the form $\rightarrow \subseteq S \times \mathbb{N} \times \mathsf{Dist}(S)$. In discrete TMDPs, the delays are interpreted as discrete jumps, with no notion of a continuously changing state as time elapses. The size $|\mathsf{T}|$ of a discrete TMDP T is $|S| + |\rightarrow|$, where $|\rightarrow|$ includes the size of the encoding of the timing constants and probabilities used in $\rightarrow$: the timing constants are written in binary, and, for any $s, s' \in S$ and (s, d, ν), the probability $\nu(s')$ is expressed as a ratio between two natural numbers, each written in binary. We let T^u be the untimed Markov decision process (MDP) corresponding to the discrete TMDP T, in which each transition $(s, d, \nu) \in \rightarrow$ is represented by a transition (s, ν). A discrete TMDP T is *structurally non-Zeno* when any finite path of T of the form $s_0 \xrightarrow{d_0, \nu_0} s_1 \cdots \xrightarrow{d_n, \nu_n} s_{n+1}$, such that $s_{n+1} = s_0$, satisfies $\sum_{0 \leq i \leq n} d_i > 0$.

- *Continuous TMDPs* are infinite-state TMDPs in which any transition $s \xrightarrow{d, \nu} s'$ describes the continuous passage of time, and thus a path $\omega = s_0 \xrightarrow{d_0, \nu_0} s_1 \xrightarrow{d_1, \nu_1} \cdots$ describes implicitly an infinite set of visited states. In the sequel, we use continuous TMDPs to give the semantics of probabilistic timed automata.

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

Definition 2.2. A *probabilistic timed automaton* (PTA) $\mathsf{P} = (L, \bar{l}, \mathcal{X}, inv, prob, \mathcal{L})$ is a tuple consisting of the following components:
- A finite set L of *locations* with the initial location $\bar{l} \in L$.
- A finite set $\mathcal{X}$ of clocks.
- A function $inv : L \rightarrow CC(\mathcal{X})$ associating an *invariant condition* with each location.
- A finite set $prob \subseteq L \times CC(\mathcal{X}) \times \mathsf{Dist}(2^{\mathcal{X}} \times L)$ of *probabilistic edges*.
- A *labelling function* $\mathcal{L} : L \rightarrow 2^{AP}$.

A probabilistic edge $(l, g, p) \in prob$ is a triple containing (1) a source location l, (2) a clock constraint g, called a *guard*, and (3) a probability distribution p which assigns probabilities to pairs of the form (X, l') for some clock set $X \subseteq \mathcal{X}$ and target location l'. The behaviour of a probabilistic timed automaton takes a similar form to that of a timed automaton [AD94]: in any location time can advance as long as the invariant holds, and a probabilistic edge can be taken if its guard is satisfied by the current values of the clocks. However, probabilistic timed automata generalize timed automata in the sense that, once a probabilistic edge is nondeterministically selected, then the choice of which clocks to reset and which target location to make the transition to is *probabilistic*. We require that the

values of the clocks after taking a probabilistic edge satisfy the invariant conditions of the target locations.
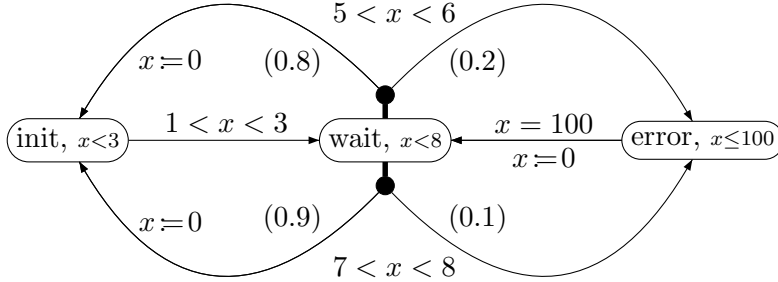

Figure 1: A probabilistic timed automaton P

Example 2.3. A PTA P is illustrated in Figure 1. The PTA represents a simple communication protocol, in which the sender can wait for between 5 and 6 time units before sending the message, at which point the message is delivered successfully with probability 0.8, or can wait for between 7 and 8 time units before sending the message, which corresponds to the message being sent successfully with probability 0.9. From location *wait*, there are two probabilistic edges: the upper one has the guard $5 < x < 6$, and assigns probability 0.8 to $(\{x\}, \textit{init})$ and 0.2 to $(\emptyset, \textit{error})$, whereas the lower one has the guard $7 < x < 8$, and assigns probability 0.9 to $(\{x\}, \textit{init})$ and 0.1 to $(\emptyset, \textit{error})$.

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

In the sequel, we assume that at least 1 time unit elapses in all structural loops within a PTA. Formally, a PTA is *structurally non-Zeno* [TYB05] if, for every sequence X_0, $(l_0, g_0, p_0), X_1, (l_1, g_1, p_1), \cdots, X_n, (l_n, g_n, p_n)$, such that $p_i(X_{i+1}, l_{i+1}) > 0$ for $0 \leq i < n$, and $p_n(X_0, l_0) > 0$, there exists a clock $x \in \mathcal{X}$ and $0 \leq i, j \leq n$ such that $x \in X_i$ and $g_j \Rightarrow x \geq 1$ (that is, g_j contains a conjunct of the form $x \geq c$ for some $c \geq 1$).

We also assume that there are no deadlock states in a PTA. This can be guaranteed by assuming that, in any state of a PTA, it is always possible to take a probabilistic edge, possibly after letting time elapse, a sufficient syntactic condition for which has been presented in [Spr01]. First, for a set $X \subseteq \mathcal{X}$ of clocks, and clock constraint $\psi \in CC(\mathcal{X})$, let $[X := 0]\psi$ be the clock constraint obtained from ψ by letting, for each $x \in X$, each conjunct of the form $x > c$ or $x \geq c'$ where $c' \geq 1$ be equal to `false`. For a clock constraint $\psi \in CC(\mathcal{X})$, let $\mathsf{upper}(\psi)$ be the clock constraint obtained from ψ by substituting constraints of the form $x < c$ with $x > c - 1 \wedge x < c$, and constraints of the form $x \leq c$ with $x \geq c \wedge x \leq c$. Then, for an invariant condition $inv(l)$ of a PTA location, the clock constraint $\mathsf{upper}(inv(l))$ represents the set of clock valuations for which a guard of a probabilistic edge must be enabled, otherwise the clock valuations correspond to deadlock

states from which it is not possible to let time pass and then take a probabilistic edge. Then a PTA has *non-deadlocking invariants* if, for each location $l \in L$, we have $\mathsf{upper}(inv(l)) \Rightarrow \bigvee_{(l,g,p) \in prob}(g \wedge \bigwedge_{(X,l') \in \mathsf{support}(p)}[X := 0]inv(l'))$. The condition of non-deadlocking invariants usually holds for PTA models in practice [KNPS06].

We use 1C-PTA (respectively, 2C-PTA) to denote the set of structurally non-Zeno PTA with non-deadlocking invariants, and with only one (respectively, two) clock(s).

2.3. **Semantics of probabilistic timed automata.** We refer to a mapping $v : \mathcal{X} \to \mathbb{R}_{\geq 0}$ as a *clock valuation*. Let $\mathbb{R}_{\geq 0}^{\mathcal{X}}$ denote the set of clock valuations. Let $\mathbf{0} \in \mathbb{R}_{\geq 0}^{\mathcal{X}}$ be the clock valuation which assigns 0 to all clocks in $\mathcal{X}$. For a clock valuation $v \in \mathbb{R}_{\geq 0}^{\mathcal{X}}$ and a value $d \in \mathbb{R}_{\geq 0}$, we use $v + d$ to denote the clock valuation obtained by letting $(v + d)(x) = v(x) + d$ for all clocks $x \in \mathcal{X}$. For a clock set $X \subseteq \mathcal{X}$, we let $v[X := 0]$ be the clock valuation obtained from v by resetting all clocks within X to 0; formally, we let $v[X := 0](x) = 0$ for all $x \in X$, and let $v[X := 0](x) = v(x)$ for all $x \in \mathcal{X} \setminus X$. The clock valuation v *satisfies* the clock constraint $\psi \in CC(\mathcal{X})$, written $v \models \psi$, if and only if ψ resolves to true after substituting each clock $x \in \mathcal{X}$ with the corresponding clock value $v(x)$.

We now present formally the semantics of PTA in terms of continuous TMDPs. The semantics has a similar form to that of non-probabilistic timed automata [AD94], but with the addition of rules for the definition of a timed, probabilistic transition relation from the probabilistic edges of the PTA.

Definition 2.4. The semantics of the probabilistic timed automaton $\mathsf{P} = (L, \bar{l}, \mathcal{X}, inv, prob, \mathcal{L})$ is the continuous TMDP $\mathsf{T}[\mathsf{P}] = (S, \bar{s}, \to, lab)$ where:

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

Given a path $\omega = (l_0, v_0) \xrightarrow{d_0, \nu_0} (l_1, v_1) \xrightarrow{d_1, \nu_1} \cdots$ of $\mathsf{T}[\mathsf{P}]$, for every $i \in \mathbb{N}$, we use $\omega(i, d)$, with $0 \leq d \leq d_i$, to denote the state $(l_i, v_i + d)$ reached from (l_i, v_i) after delaying d time units. Such a pair (i, d) is called a *position* of ω. We define a total order on positions of ω: given two positions $(i, d), (j, d')$ of ω, the position (i, d) precedes (j, d') — denoted $(i, d) \prec_\omega (j, d')$ — if and only if either $i < j$, or $i = j$ and $d < d'$.

3. Probabilistic timed temporal logic

We now proceed to describe a *probabilistic, timed* temporal logic which can be used to specify properties of probabilistic timed automata [KNSS02].

Definition 3.1. The formulae of Ptctl (Probabilistic Timed Computation Tree Logic) are given by the following grammar:

$$\Phi ::= a \mid \Phi \wedge \Phi \mid \neg \Phi \mid \mathbb{P}_{\bowtie \zeta}(\Phi \mathsf{U}_{\sim c} \Phi)$$

where $a \in AP$ is an atomic proposition, $\bowtie \in \{<, \leq, \geq, >\}$, $\sim \in \{\leq, =, \geq\}$, $\zeta \in [0, 1]$ is a probability, and $c \in \mathbb{N}$ is a natural number.

We use standard abbreviations such as $\texttt{true}$, $\texttt{false}$, $\Phi_1 \vee \Phi_2$, $\Phi_1 \Rightarrow \Phi_2$, and $\mathbb{P}_{\bowtie\zeta}(\mathsf{F}_{\sim c}\Phi)$ (for $\mathbb{P}_{\bowtie\zeta}(\texttt{true}\mathsf{U}_{\sim c}\Phi)$). Formulae with "always" temporal operators $\mathsf{G}_{\sim c}$ can also be written; for example $\mathbb{P}_{\geq\zeta}(\mathsf{G}_{\sim c}\Phi)$ can be expressed by $\mathbb{P}_{\leq 1-\zeta}(\mathsf{F}_{\sim c}\neg\Phi)$. The modalities U, F and G without subscripts abbreviate $\mathsf{U}_{\geq 0}$, $\mathsf{F}_{\geq 0}$ and $\mathsf{G}_{\geq 0}$, respectively.

We identify the following sub-logics of PTCTL.

- PTCTL$[\leq, \geq]$ is defined as the sub-logic of PTCTL in which subscripts of the form $= c$ are not allowed in modalities $\mathsf{U}_{\sim c}, \mathsf{F}_{\sim c}, \mathsf{G}_{\sim c}$.
- PCTL is defined as the sub-logic of PTCTL (and PTCTL$[\leq, \geq]$) in which there is no timing subscript $\sim c$ associated with the modalities $\mathsf{U}, \mathsf{F}, \mathsf{G}$.
- PTCTL$^{0/1}$ and PTCTL$^{0/1}[\leq, \geq]$ are the sub-logics of PTCTL and PTCTL$[\leq, \geq]$, respectively, in which probability thresholds ζ belong to $\{0, 1\}$. We refer to PTCTL$^{0/1}$ and PTCTL$^{0/1}[\leq, \geq]$ as the *qualitative* restrictions of PTCTL and PTCTL$[\leq, \geq]$.
- *Reachability properties* are those PCTL properties of the form $\mathbb{P}_{\bowtie\zeta}(\mathsf{F}a)$ or $\neg\mathbb{P}_{\bowtie\zeta}(\mathsf{F}a)$. Qualitative reachability properties are those reachability properties for which $\zeta \in \{0, 1\}$.

The size $|\Phi|$ of a PTCTL formula Φ is defined in the standard way as the number of symbols in Φ, with each occurrence of the same subformula of Φ as a single symbol.

We now define the satisfaction relation of PTCTL for discrete TMDPs. Given the infinite path $\omega = s_0 \xrightarrow{d_0,\nu_0} s_1 \xrightarrow{d_1,\nu_1} \cdots$ of the discrete TMDP T, let $\mathsf{DiscDur}(\omega, i) = \sum_{0 \leq k < i} d_k$ be the accumulated duration along ω until $(i + 1)$-th state.

Definition 3.2. Given a discrete TMDP $\mathsf{T} = (S, \bar{s}, \rightarrow, lab)$ and a PTCTL formula Φ, we define the satisfaction relation $\models_\mathsf{T}$ of PTCTL as follows:

$$
\begin{array}{llll}
s & \models_\mathsf{T} & a & \text{iff} \quad a \in lab(s) \\
s & \models_\mathsf{T} & \Phi_1 \wedge \Phi_2 & \text{iff} \quad s \models_\mathsf{T} \Phi_1 \text{ and } s \models_\mathsf{T} \Phi_2 \\
s & \models_\mathsf{T} & \neg\Phi & \text{iff} \quad s \not\models_\mathsf{T} \Phi \\
s & \models_\mathsf{T} & \mathbb{P}_{\bowtie\zeta}(\varphi) & \text{iff} \quad Prob_s^A\{\omega \in Path_{ful}^A(s) \mid \omega \models_\mathsf{T} \varphi\} \bowtie \zeta, \; \forall A \in Adv \\
\omega & \models_\mathsf{T} & \Phi_1 \mathsf{U}_{\sim c} \Phi_2 & \text{iff} \quad \exists i \in \mathbb{N} \text{ s.t. } \omega(i) \models_\mathsf{T} \phi_2, \; \mathsf{DiscDur}(\omega, i) \sim c, \\
& & & \quad\quad \text{and } \omega(j) \models_\mathsf{T} \phi_1, \; \forall j < i \; .
\end{array}
$$

We proceed to define the satisfaction relation of PTCTL for continuous TMDPs. Given the infinite path $\omega = s_0 \xrightarrow{d_0,\nu_0} s_1 \xrightarrow{d_1,\nu_1} \cdots$ of the continuous TMDP T, let $\mathsf{CtsDur}(\omega, i, d) = d + \sum_{0 \leq k < i} d_k$ be the accumulated duration along ω until position (i, d).

Definition 3.3. Given a continuous TMDP $\mathsf{T} = (S, \bar{s}, \rightarrow, lab)$ and a PTCTL formula Φ, we define the satisfaction relation $\models_\mathsf{T}$ of PTCTL as in Definition 3.2, except for the following rule for $\Phi_1 \mathsf{U}_{\sim c} \Phi_2$:

$$
\begin{array}{llll}
\omega & \models_\mathsf{T} & \Phi_1 \mathsf{U}_{\sim c} \Phi_2 & \text{iff} \quad \exists \text{ position } (i, \delta) \text{ of } \omega \text{ s.t. } \omega(i, \delta) \models_\mathsf{T} \phi_2, \; \mathsf{CtsDur}(\omega, i, \delta) \sim c, \\
& & & \quad\quad \text{and } \omega(j, \delta') \models_\mathsf{T} \phi_1, \; \forall \text{ positions } (j, \delta') \text{ of } \omega \text{ s.t. } (j, \delta') \prec_\omega (i, \delta) \; .
\end{array}
$$

When clear from the context, we omit the T subscript from $\models_\mathsf{T}$. We say that the TMDP $\mathsf{T} = (S, \bar{s}, \rightarrow, lab)$ satisfies the PTCTL formula Φ, denoted by $\mathsf{T} \models \Phi$, if and only if $\bar{s} \models \Phi$. Furthermore, the PTA P satisfies Φ, denoted by $\mathsf{P} \models \Phi$, if and only if $\mathsf{T}[\mathsf{P}] \models \Phi$.

Complexity of PTCTL *model checking for PTA.* Given an arbitrary structurally non-Zeno PTA P, model checking PTCTL formulae is in EXPTIME [KNSS02] (the algorithm consists of executing a standard polynomial-time model-checking algorithm for finite-state probabilistic systems [BdA95, BK98] on the exponential-size region graph of P). The problem of model checking qualitative reachability formulae of the form $\neg\mathbb{P}_{<1}(\mathsf{F}a)$ is EXPTIME-hard for PTA with an arbitrary number of clocks [LS07]. Hence PTCTL model checking for structurally non-Zeno PTA with an arbitrary number of clocks is EXPTIME-complete.

Example 3.4. Consider the PTA P of Figure 1. The formula $\mathbb{P}_{>0}(\mathsf{F}_{\leq 9}error)$ holds for the configuration $(init, 0)$: for every non-deterministic choice, the probability to reach *error* within 9 time units is strictly positive. The formula $\mathbb{P}_{<0.1}(\mathsf{F}_{\leq 6}error)$ does not hold for $(init, 0)$: if the adversary chooses to delay until $x = 5.4$ in *wait*, and then performs the probabilistic edge with the guard $5 < x < 6$, then the probability to reach *error* is 0.2. Note also that the formula $\mathbb{P}_{\geq 0.1}(\mathsf{F}_{\leq 6}error)$ is not true either in $(init, 0)$: the adversary can choose to delay in *wait* until $x = 7.8$ and then perform the second probabilistic edge, in which case the probability to reach *error* within 6 time units is zero.

4. Model Checking One-Clock Probabilistic Timed Automata

In this section we consider the case of 1C-PTA. We will see that model checking PCTL and PTCTL$^{0/1}[\leq, \geq]$ for 1C-PTA is P-complete, but remains EXPTIME-complete for the logic PTCTL$^{0/1}$.

4.1. **Model Checking** PCTL **on 1C-PTA.** First we present the following result about the model checking of PCTL formulae.

Proposition 4.1. *The* PCTL *model-checking problem for 1C-PTA is P-complete.*

Proof. The problem is P-hard because model checking formulae of the form $\neg\mathbb{P}_{<1}(\mathsf{F}a)$ in finite MDPs is P-hard [PT87]. Here we show P-membership. For this we adapt the encoding for showing NLOGSPACE-membership of reachability in one-clock timed automata [LMS04] in order to obtain an untimed MDP which is polynomial in the size of the 1C-PTA. This untimed MDP is then subject to the established polynomial-time PCTL model-checking algorithm [BdA95].

Let $\mathsf{P} = (L, \bar{l}, \{x\}, inv, prob, \mathcal{L})$ be a 1C-PTA. A state of P is a control location and a value v for x. The exact value of x is not important to solve the problem: we just need to know in which interval (with respect to the constants occurring in the guards and invariants of P) is x. Let $\mathsf{Cst}(\mathsf{P})$ be the set of integer values used in the guards and invariants of P, and let $\mathbb{B} = \mathsf{Cst}(\mathsf{P}) \cup \{0\}$. We use $b_0, b_1, \ldots, b_k$ to range over $\mathbb{B}$, where $0 = b_0 < b_1 < \cdots < b_k$ and $|\mathbb{B}| = k+1$. The set $\mathbb{B}$ defines a set $\mathcal{I}_{\mathbb{B}}$ of $2(k+1)$ intervals $[b_0; b_0], (b_0; b_1), [b_1; b_1], \cdots, (b_k, \infty)$. We also define a total order on the set $\mathcal{I}_{\mathbb{B}}$, where $[b_0; b_0] < (b_0; b_1) < [b_1; b_1] < \cdots < (b_k, \infty)$. The configuration (l, v) is then encoded by the pair $(l, n(v))$ such that v belongs to the $n(v)$-th interval in $\mathcal{I}_{\mathbb{B}}$: note that the length of the binary representation of the number of an interval is $\log(2(k+1))$. We then build an untimed MDP $\mathsf{M}[\mathsf{P}]$ whose states are the pairs $(l, n(v))$ and the transitions simulate those of P. Note that we can easily decide whether a guard is satisfied by the clock values of the $n(v)$-th interval. A step of P from (l, v) consists in choosing a duration d and a distribution μ (as represented by the transition $((l, v), d, \mu)$), and finally making a probabilistic choice. Such a step is simulated in $\mathsf{M}[\mathsf{P}]$ by a transition

$((l, n(v)), \nu)$, which corresponds to choosing the appropriate interval $n(v + d)$ in the future (i.e., $n(v + d) \geq n(v)$), then making a probabilistic choice according to the distribution ν from $(l, n(v + d))$, where $\nu(l', n(v')) = \mu(l', v')$ for each state (l', v') of $\mathsf{T[P]}$.

For a clock constraint $\psi \in CC(\{x\})$, let $[\![\psi]\!] = \{v \in \mathbb{R}_{\geq 0} \mid v \models \psi\}$. For an interval $I \subseteq \mathbb{R}_{\geq 0}$, let $I[\{x\} := 0] = [0; 0]$ and $I[\emptyset := 0] = I$. The *MDP for* PCTL *of the PTA* P is the untimed MDP $\mathsf{M[P]} = (S_\mathsf{M}, \bar{s}_\mathsf{M}, \rightarrow_\mathsf{M}, lab_\mathsf{M})$ where:

- $S_\mathsf{M} = \{(l, B) \mid l \in L, B \in \mathcal{I}_\mathbb{B} \text{ and } B \subseteq [\![inv(l)]\!]\}$ and $\bar{s}_\mathsf{M} = (\bar{l}, [0, 0])$;
- $\rightarrow_\mathsf{M}$ is the least set such that $((l, B), \nu) \in \rightarrow_\mathsf{M}$ if there exists an interval $B' \in \mathcal{I}_\mathbb{B}$ and a probabilistic edge $(l, g, p) \in prob$ such that:
  (1) $B' \geq B$, $B' \subseteq [\![g]\!]$, and $B'' \subseteq [\![inv(l)]\!]$ for all $B \leq B'' \leq B'$;
  (2) for any $(X, l') \in \{\{x\}, \emptyset\} \times L$, we have that $p(X, l') > 0$ implies $(B' \cap [\![g]\!])[X := 0] \subseteq [\![inv(l')]\!]$;
  (3) for any $(l', B'') \in S_\mathsf{M}$, we have that $\nu(l', B'') = \nu_0(l', B'') + \nu_{B'}(l', B'')$, where $\nu_0(l', B'') = p(\{x\}, l')$ if $B'' = [0, 0]$ and $\nu_0(l', B'') = 0$ otherwise, and where $\nu_{B'}(l', B'') = p(\emptyset, l')$ if $B' = B''$ and $\nu_{B'}(l', B'') = 0$ otherwise.
- lab_M is such that $lab_\mathsf{M}(l, B) = \mathcal{L}(l)$ for each state $(l, B) \in S_\mathsf{M}$.

Given a PCTL formula Φ and a state (l, v) of $\mathsf{T[P]}$, we then have that $(l, v) \models_{\mathsf{T[P]}} \Phi$ if and only if $(l, n(v)) \models_{\mathsf{M[P]}} \Phi$, which can be shown by induction on the length of the formula. The cases of atomic propositions and boolean combinators are straightforward, and therefore we concentrate on the case of a formula $\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U} \Phi_2)$. We can show that, for each adversary A of $\mathsf{T[P]}$, it is possible to construct an adversary A' of $\mathsf{M[P]}$ such that, for each state (l, v) of $\mathsf{T[P]}$, we have $Prob_{(l,v)}^A \{\omega \in Path_{ful}^A(l, v) \mid \omega \models_{\mathsf{T[P]}} \Phi_1 \mathsf{U} \Phi_2\} = Prob_{(l,n(v))}^{A'} \{\omega \in Path_{ful}^{A'}(l, n(v)) \mid \omega \models_{\mathsf{M[P]}} \Phi_1 \mathsf{U} \Phi_2\}$. Conversely, we can show that, for each adversary A of $\mathsf{M[P]}$, it is possible to construct an adversary A' of $\mathsf{T[P]}$ such that, for each state (l, v) of $\mathsf{T[P]}$, we have $Prob_{(l,n(v))}^A \{\omega \in Path_{ful}^A(l, n(v)) \mid \omega \models_{\mathsf{M[P]}} \Phi_1 \mathsf{U} \Phi_2\} = Prob_{(l,v)}^{A'} \{\omega \in Path_{ful}^{A'}(l, v) \mid \omega \models_{\mathsf{T[P]}} \Phi_1 \mathsf{U} \Phi_2\}$. By the definition of the semantics of PCTL, given (l, v), we have $(l, v) \models_{\mathsf{T[P]}} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U} \Phi_2)$ if and only if $(l, n(v)) \models_{\mathsf{M[P]}} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U} \Phi_2)$.

The size of M is in $O(|\mathsf{P}| \cdot 2 \cdot |\mathbb{B}|)$ and $|\mathbb{B}|$ is in $O(2 \cdot |prob|)$. Because PCTL model checking is polynomial in the size of the MDP [BdA95], we have obtained a polynomial-time algorithm for PCTL model checking for PTA. $\qquad\square$

4.2. Model checking $\text{PTCTL}^{0/1}[\leq, \geq]$ on 1C-PTA.

In this section, inspired by related work on discrete-time concurrent game structures [LMO06], we first show that model-checking $\text{PTCTL}^{0/1}[\leq, \geq]$ properties of discrete TMDPs can be done efficiently. Then, in Theorem 4.3, using ideas from the TMDP case, we show that model checking $\text{PTCTL}^{0/1}[\leq, \geq]$ on 1C-PTA can also be done in polynomial time.

Proposition 4.2. *Let* $\mathsf{T} = (S, \bar{s}, \rightarrow, lab)$ *be a structurally non-Zeno discrete TMDP and* Φ *be a* $\text{PTCTL}^{0/1}[\leq, \geq]$ *formula. Deciding whether* $\mathsf{T} \models \Phi$ *can be done in time* $O(|\Phi| \cdot |S| \cdot |\rightarrow|)$.

Proof sketch. The model-checking algorithm is based on several procedures to deal with each modality of $\text{PTCTL}^{0/1}[\leq, \geq]$. The boolean operators and the PCTL modalities (without timed subscripts) can be handled in the standard manner, with the PCTL properties verified on the untimed MDP T^u corresponding to T. For formulae $\mathbb{P}_{\bowtie\zeta}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$, we assume that the truth values of subformulae Φ_1 and Φ_2 are known for all states of T. First, given that

the TMDP is structurally non-Zeno, we have the equivalences:

$$\begin{aligned}
\mathbb{P}_{\leq 0}(\Phi_1 \mathsf{U}_{\sim c}\Phi_2) &\equiv \neg\mathsf{E}(\Phi_1 \mathsf{U}_{\sim c}\Phi_2) \\
\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}_{\leq c}\Phi_2) &\equiv \mathsf{A}(\Phi_1 \mathsf{U}_{\leq c}\Phi_2) \\
\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}_{\geq c}\Phi_2) &\equiv \mathsf{A}(\Phi_1 \mathsf{U}_{\geq c}(\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}\Phi_2)))
\end{aligned}$$

where E (respectively, A) stands for the existential (respectively, universal) quantification over paths which exist in the logic TCTL. Thus we can apply the procedure proposed for model checking TCTL formulae – running in time $O(|S| \cdot |\rightarrow|)$ – over weighted graphs [LMS05] (in the case of $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}_{\geq c}\Phi_2)$, by first obtaining the set of states satisfying $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}\Phi_2)$, which can be done on T^u in time $O(|\mathsf{Edges}(\rightarrow)|)$, where $|\mathsf{Edges}(\rightarrow)| = \sum_{(s,d,\nu)\in\rightarrow}|\mathsf{support}(\nu)|)$.

The problem of verifying the remaining temporal properties of PTCTL$^{0/1}[\leq,\geq]$ can be considered in terms of turn-based 2-player games. Such a game is played over the space $S \cup \rightarrow$, and play proceeds as follows: from a state $s \in S$, player P_n (representing nondeterministic choice) chooses a transition $(s,d,\nu) \in \rightarrow$; then, from the transition (s,d,ν), player P_p (representing probabilistic choice) chooses a state $s' \in \mathsf{support}(\nu)$. The duration of the move from s to s' via (s,d,ν) is d. Notions of strategy of each player, and winning with respect to (untimed) path formulae of the form $\Phi_1 \mathsf{U}\Phi_2$, are defined as usual for 2-player games.

For the four remaining formulae, namely $\mathbb{P}_{\bowtie\zeta}(\Phi_1 \mathsf{U}_{\sim c}\Phi_2)$ for $\bowtie\zeta \in \{> 0, < 1\}$, and $\sim\in\{\leq,\geq\}$, we consider the functions $\alpha,\beta,\gamma,\delta : S \rightarrow \mathbb{N}$, for representing minimal and maximal durations of interest. Intuitively, for a state $s \in S$, the value $\alpha(s)$ (respectively, $\gamma(s)$) is the minimal (respectively, maximal) duration that player P_p can ensure, regardless of the counter-strategy of P_n, along a path prefix from s satisfying $\Phi_1 \mathsf{U}\Phi_2$ (respectively, $\Phi_1 \mathsf{U}(\mathbb{P}_{>0}(\Phi_1 \mathsf{U}\Phi_2)))$. Similarly, the value $\beta(s)$ (respectively, $\delta(s)$) is the minimal (respectively, maximal) duration that player P_n can ensure, regardless of the counter-strategy of P_p, along a path prefix from s satisfying $\Phi_1 \mathsf{U}\Phi_2$ (respectively, $\Phi_1 \mathsf{U}(\neg\mathbb{P}_{<1}(\Phi_1 \mathsf{U}\Phi_2)))$.

If there is no strategy for player P_p (respectively, player P_n) to guarantee the satisfaction of $\Phi_1 \mathsf{U}\Phi_2$ along a path prefix from s, then we let $\alpha(s) = \infty$ (respectively, $\beta(s) = \infty$). Similarly, if there is no strategy for player P_p (respectively, player P_n) to guarantee the satisfaction of $\Phi_1 \mathsf{U}(\mathbb{P}_{>0}(\Phi_1 \mathsf{U}\Phi_2))$ (respectively, $\Phi_1 \mathsf{U}(\neg\mathbb{P}_{<1}(\Phi_1 \mathsf{U}\Phi_2)))$ along a path prefix from s, then we let $\gamma(s) = -\infty$ (respectively, $\delta(s) = -\infty$).

Using the fact that the TMDP is structurally non-Zeno, for any state $s \in S$, we can obtain the following equivalences:

- $s \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq c}\Phi_2)$ if and only if $\alpha(s) \leq c$;
- $s \models \mathbb{P}_{<1}(\Phi_1 \mathsf{U}_{\leq c}\Phi_2)$ if and only if $\beta(s) > c$;
- $s \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\geq c}\Phi_2)$ if and only if $\gamma(s) \geq c$;
- $s \models \mathbb{P}_{<1}(\Phi_1 \mathsf{U}_{\geq c}\Phi_2)$ if and only if $\delta(s) < c$.

The functions $\alpha,\beta,\gamma,\delta$ can be computed on the 2-player game by applying the same methods as in [LMO06] for discrete-time concurrent game structures: for each temporal operator $\mathbb{P}_{\bowtie\zeta}(\Phi_1 \mathsf{U}_{\sim c}\Phi_2)$, this computation runs in time $O(|S| \cdot |\rightarrow|)$. We decompose the proof into the following four cases, which depend on the form of the formula to be verified.

$\Phi = \mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq c}\Phi_2)$. To compute the value $\alpha(s)$, we introduce the coefficients $\alpha^i(s)$ defined recursively as follows. Let $\alpha^0(s) = 0$ if $s \models \Phi_2$, let $\alpha^0(s) = \infty$ otherwise, and let:

$$\alpha^{i+1}(s) = \begin{cases} 0 & \text{if } s \models \Phi_2 \\ \infty & \text{if } s \models \neg\Phi_1 \wedge \neg\Phi_2 \\ \max\limits_{(s,d,\nu)\in\rightarrow}\{d + \min\limits_{s'\in\mathsf{support}(\nu)}\{\alpha^i(s')\}\} & \text{if } s \models \Phi_1 \wedge \neg\Phi_2. \end{cases}$$

Fact 1. If $\alpha^i(s) < \infty$, the value $\alpha^i(s)$ is the minimal duration that player P_p can ensure from s with respect to $\Phi_1\mathsf{U}\Phi_2$ in <u>at most $2i$ turns</u>. If $\alpha^i(s) = \infty$, player P_p cannot ensure $\Phi_1\mathsf{U}\Phi_2$ in $2i$ turns.

Proof of Fact 1. The proof proceeds by induction over i. The result is immediate for $i = 0$. Now assume the property holds up to i.

Consider $\alpha^{i+1}(s)$. The cases for $\alpha^{i+1}(s) = 0$, and $\alpha^{i+1}(s) = \infty$ with $s \models \neg\Phi_1 \wedge \neg\Phi_2$, are trivial. Now assume $\alpha^{i+1}(s) = \infty$ and $s \models \Phi_1 \wedge \neg\Phi_2$: by the definition of $\alpha^{i+1}(s)$, there exists a transition $(s, \_, \nu)$ from s such that any possible successor $s' \in \mathsf{support}(\nu)$ verifies $\alpha^i(s') = \infty$. By the induction hypothesis this entails that there is no strategy for P_p to ensure $\Phi_1\mathsf{U}\Phi_2$ in less than $2i$ turns from any $s' \in \mathsf{support}(\nu)$, and then there is no strategy for P_p from s for games with $2(i+1)$ turns.

Assume $\alpha^{i+1}(s) \in \mathbb{N}$. Let θ be the minimal duration that player P_p can ensure with respect to $\Phi_1\mathsf{U}\Phi_2$, for games with at most $2(i+1)$ turns. This duration θ is obtained from a choice of transition (s, d, ν) of P_n and a choice of state $s' \in \mathsf{support}(\nu)$ of P_p, where, by the induction hypothesis, we have $\theta = d + \alpha^i(s')$. We also have that this s' is the best (minimal) choice for P_p among all states in $\mathsf{support}(\nu)$; that is, $\alpha^i(s') = \min_{s''\in\mathsf{support}(\nu)}\{\alpha^i(s'')\}$. Given the definition of $\alpha^{i+1}(s)$, we have that $\alpha^{i+1}(s)$ equals:

$$\max_{(s,d',\nu')\in\rightarrow}\{d' + \min_{s''\in\mathsf{support}(\nu')}\{\alpha^i(s'')\}\} \geq \{d + \min_{s''\in\mathsf{support}(\nu)}\{\alpha^i(s'')\}\} = d + \alpha^i(s') = \theta \ ,$$

However, as θ corresponds to the best (maximal) choice for P_n, we cannot have $\alpha^{i+1}(s) > \theta$, and therefore $\alpha^{i+1}(s) = \theta$. □

We claim that $\alpha^{|S|}(s) = \alpha(s)$. First note that we clearly have $\alpha^{|S|}(s) \geq \alpha(s)$. Now assume $\alpha(s) < \alpha^{|S|}(s)$: this value $\alpha(s)$ is obtained by a strategy (for P_p) that uses more than $2|S|$ turns. Therefore, along some path generated by this strategy there will be at least one occurrence of a state s'. However, as the TMDP is structurally non-Zeno, this loop has a duration strictly greater than 0, and it can be removed by applying earlier in the path the last choice done for state s' along the path[1]. Such a looping strategy is clearly not optimal for P_p and need not be considered when computing $\alpha(s)$. Hence the computation of $\alpha^{|S|}$, and thus α, can be done in time $O(|S| \cdot |\rightarrow|)$.

$\Phi = \mathbb{P}_{>0}(\Phi_1\mathsf{U}_{\geq c}\Phi_2)$. In order to establish the set of states satisfying Φ, we first compute the sets of states satisfying two untimed, auxiliary formulae. The first formula we consider is $\mathbb{P}_{>0}(\Phi_1\mathsf{U}\Phi_2)$: obtaining the set of states satisfying this formula relies on qualitative PCTL analysis of the underlying untimed MDP T^u of T, which can be done in time $O(|\mathsf{Edges}(\rightarrow)|)$. The second formula we consider is $\mathbb{P}_{>0}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2)$, where, for any infinite path $\omega \in Path_{ful}$, we have $\omega \models \Phi_1\mathsf{U}^{\geq 1}\Phi_2$ if and only if there exists $i \geq 1$ such that $\omega(i) \models \Phi_2$, and $\omega(j) \models \Phi_1$ for all $j < i$. The set of states satisfying $\mathbb{P}_{>0}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2)$ can be obtained through a combination of the usual "next" temporal operator of PCTL (see [HJ94, BdA95]) and the formula $\mathbb{P}_{>0}(\Phi_1\mathsf{U}\Phi_2)$, and can be computed in time $O(|\mathsf{Edges}(\rightarrow)|)$.

---

[1]Note that as $\alpha(s) \neq \infty$, the path induced by the strategy of player P_p is finite.

We then proceed to compute, for each state s of T satisfying $\mathbb{P}_{>0}(\Phi_1 \mathsf{U} \Phi_2)$, the maximal duration $\gamma(s)$ that player P_p can ensure with respect to $\Phi_1 \mathsf{U}(\mathbb{P}_{>0}(\Phi_1 \mathsf{U} \Phi_2))$. We compute γ using the following recursive rules:

$$\gamma^0(s) = \begin{cases} -\infty & \text{if } s \models \neg \mathbb{P}_{>0}(\Phi_1 \mathsf{U} \Phi_2) \\ 0 & \text{if } s \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U} \Phi_2) \wedge \neg \mathbb{P}_{>0}(\Phi_1 \mathsf{U}^{\geq 1} \Phi_2) \\ \infty & \text{if } s \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U}^{\geq 1} \Phi_2) \end{cases}$$

$$\gamma^{i+1}(s) = \begin{cases} -\infty & \text{if } s \models \neg \mathbb{P}_{>0}(\Phi_1 \mathsf{U} \Phi_2) \\ 0 & \text{if } s \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U} \Phi_2) \wedge \neg \mathbb{P}_{>0}(\Phi_1 \mathsf{U}^{\geq 1} \Phi_2) \\ \min\limits_{(s,d,\nu) \in \rightarrow} \{d + \max\limits_{s' \in \mathsf{support}(\nu)} \{\gamma^i(s')\}\} & \text{if } s \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U}^{\geq 1} \Phi_2) \end{cases}$$

We have the following fact, the proof of which is similar to that of Fact 1.

Fact 2. If $-\infty < \gamma^i(s) < \infty$, then $\gamma^i(s)$ is the maximal duration that player P_p can ensure from s with respect to $\Phi_1 \mathsf{U}(\mathbb{P}_{>0}(\Phi_1 \mathsf{U} \Phi_2))$ in <u>at most $2i$ turns</u>. If $\gamma^i(s) = \infty$ (respectively, $\gamma^i(s)) = -\infty$), then player P_p can ensure $\mathbb{P}_{>0}(\Phi_1 \mathsf{U}^{\geq 1} \Phi_2)$ continuously during $2i$ turns (respectively, cannot ensure $\Phi_1 \mathsf{U} \Phi_2$).

Proof of Fact 2. Consider $\gamma^{i+1}(s)$. The cases for $\gamma^{i+1}(s) = 0$, and $\gamma^{i+1}(s) = -\infty$ are immediate.

Assume $\gamma^{i+1}(s) = \infty$. Then for any distribution from s, there is a probabilistic choice leading to some s' with $\gamma^i(s') = \infty$. By the induction hypothesis, we deduce that player P_p can ensure $\mathbb{P}_{>0}(\Phi_1 \mathsf{U}^{\geq 1} \Phi_2)$ during $2(i+1)$ turns from s.

Assume $\gamma^{i+1}(s) \in \mathbb{N}$. Let θ be the maximal duration that player P_p can ensure with respect to $\Phi_1 \mathsf{U} \Phi_2$, for games with at most $2(i+1)$ turns. This duration θ is obtained from a choice of (s,d,ν) of P_n and a choice of $s' \in \mathsf{support}(\nu)$ of P_p, where, by the induction hypothesis, we have $\theta = d + \gamma^i(s')$. We also have that this s' is the best (maximal) choice for P_p among all states in $\mathsf{support}(\nu)$; that is, $\gamma^i(s') = \max_{s'' \in \mathsf{support}(\nu)}\{\gamma^i(s'')\}$. We have that $\gamma^{i+1}(s)$ equals:

$$\min_{(s,d',\nu') \in \rightarrow} \{d' + \max_{s'' \in \mathsf{support}(\nu')}\{\gamma^i(s'')\}\} \leq \{d + \max_{s'' \in \mathsf{support}(\nu)}\{\gamma^i(s'')\}\} = d + \gamma^i(s') = \theta .$$

However, as θ corresponds to the best (minimal) choice for P_n, we cannot have $\gamma^{i+1}(s) < \theta$, and therefore $\gamma^{i+1}(s) = \theta$. $\square$

As in the case of the function α, we claim that $\gamma^{|S|}(s) = \gamma(s)$. We clearly have $\gamma^{|S|}(s) \geq \gamma(s)$ (indeed we can prove by induction over i that $\gamma^i(s) \geq \gamma(s)$ for any $i \geq 0$). Assume that $\gamma(s) < \gamma^{|S|}(s)$; then as in the case of α, the value $\gamma(s)$ is obtained by a strategy for P_p which generates a path whose length is greater than $|S|$ along which a state is visited twice. The assumption of structural non-Zenoness means that, if the strategy can choose to repeat s' an arbitrary number of times, the elapsed duration along the path becomes arbitrarily large and $\gamma(s) = \gamma^{|S|}(s) = \infty$. Hence, there is no need to explore further the path. Therefore the computation of $\gamma^{|S|}$, and thus γ, can be done in time $O(|S| \cdot |\rightarrow|)$.

$\Phi = \mathbb{P}_{<1}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$. This case can be treated in a similar manner as the case of $\Phi = \mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$. Here we aim at computing the minimum duration $\beta(s)$ that player P_n can ensure with respect to $\Phi_1 \mathsf{U} \Phi_2$. Then Φ holds for s if and only if $\beta(s) > c$. We compute the

following values $\beta^i(s)$ with $\beta^0(s) = 0$ if $s \models \Phi_2$, $\beta^0(s) = \infty$ otherwise, and:

$$\beta^{i+1}(s) = \begin{cases} 0 & \text{if } s \models \Phi_2 \\ \infty & \text{if } s \models \neg\Phi_1 \wedge \neg\Phi_2 \\ \min_{(s,d,\nu)\in\rightarrow}\{d + \max_{s'\in\mathsf{support}(\nu)}\{\beta^i(s')\}\} & \text{otherwise.} \end{cases}$$

Fact 3. If $\beta^i(s) < \infty$, the value $\beta^i(s)$ is the minimal duration that player P_n can ensure from s with respect to $\Phi_1\mathsf{U}\Phi_2$ in <u>at most $2i$ turns</u>. If $\beta^i(s) = \infty$, player P_n cannot ensure $\Phi_1\mathsf{U}\Phi_2$ in $2i$ turns.

The proof of Fact 3 proceeds in a similar manner to that of Fact 1, but with the roles of players P_n and P_p reversed, and therefore we omit it. Furthermore, we have $\beta^{|S|}(s) = \beta(s)$ for similar reasons that we had $\alpha^{|S|} = \alpha(s)$ (again, with the roles of P_n and P_p reversed), and hence the computation of β can be done in time $O(|S| \cdot | \rightarrow |)$.

$\Phi = \mathbb{P}_{<1}(\Phi_1\mathsf{U}_{\geq c}\Phi_2)$. This property is true when player P_n has no strategy to ensure $\Phi_1\mathsf{U}_{\geq c}\Phi_2$. Similarly to the case of $\mathbb{P}_{>0}(\Phi_1\mathsf{U}_{\geq c}\Phi_2)$, we first compute the sets of states satisfying two untimed formulae, namely $\mathbb{P}_{<1}(\Phi_1\mathsf{U}\Phi_2)$ and $\mathbb{P}_{<1}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2)$, the complexity of which is in $O(|\mathsf{Edges}(\rightarrow)|\sqrt{|\mathsf{Edges}(\rightarrow)|})$ [CJH03]. We then compute, for each state s of T satisfying $\neg\mathbb{P}_{<1}(\Phi_1\mathsf{U}\Phi_2)$, the maximal duration $\delta(s)$ that player P_n can ensure with respect to $\Phi_1\mathsf{U}(\mathbb{P}_{<1}(\neg\Phi_1\mathsf{U}\Phi_2))$. Then $s \models \Phi$ if and only if $\delta(s) < c$. We compute δ using the following recursive rules:

$$\delta^0(s) = \begin{cases} \infty & \text{if } s \models \neg\mathbb{P}_{<1}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2) \\ 0 & \text{if } s \models \neg\mathbb{P}_{<1}(\Phi_1\mathsf{U}\Phi_2) \wedge \mathbb{P}_{<1}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2) \\ -\infty & \text{if } s \models \mathbb{P}_{<1}(\Phi_1\mathsf{U}\Phi_2) \end{cases}$$

$$\delta^{i+1}(s) = \begin{cases} -\infty & \text{if } s \models \mathbb{P}_{<1}(\Phi_1\mathsf{U}\Phi_2) \\ 0 & \text{if } s \models \neg\mathbb{P}_{<1}(\Phi_1\mathsf{U}\Phi_2) \wedge \mathbb{P}_{<1}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2) \\ \max_{(s,d,\nu)\in\rightarrow}\{d + \min_{s'\in\mathsf{support}(\nu)}\{\delta^i(s')\}\} & \text{if } s \models \neg\mathbb{P}_{<1}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2) \end{cases}$$

Fact 4. If $-\infty < \delta^i(s) < \infty$, then $\delta^i(s)$ is the maximal duration that player P_n can ensure from s with respect to $\Phi_1\mathsf{U}(\mathbb{P}_{>0}(\Phi_1\mathsf{U}\Phi_2))$ in <u>at most $2i$ turns</u>. If $\delta^i(s) = \infty$ (respectively, $\delta^i(s) = -\infty$), then player P_n can ensure $\neg\mathbb{P}_{<1}(\Phi_1\mathsf{U}^{\geq 1}\Phi_2)$ during $2i$ turns (respectively, cannot ensure $\Phi_1\mathsf{U}(\neg\mathbb{P}_{<1}(\Phi_1\mathsf{U}\Phi_2)))$ from s.

We can adapt the reasoning used in Fact 2 to prove this fact (as in the case of Fact 3). Finally, with similar reasoning to that used in the case of $\mathbb{P}_{>0}(\Phi_1\mathsf{U}_{\geq c}\Phi_2)$, we can show that $\delta^{|S|}(s) = \delta(s)$, and therefore δ can be computed in time $O(|S| \cdot | \rightarrow |)$.

Finally we obtain an algorithm running in time $O(|\Phi| \cdot |S| \cdot | \rightarrow |)$. $\square$

We use Proposition 4.2 to obtain an efficient model-checking algorithm for 1C-PTA.

Theorem 4.3. *Let* $\mathsf{P} = (L, \bar{l}, \mathcal{X}, inv, prob, \mathcal{L})$ *be a 1C-PTA and* Φ *be a* PTCTL$^{0/1}[\leq, \geq]$ *formula. Deciding whether* $\mathsf{P} \models \Phi$ *can be done in polynomial time.*

Proof sketch. Our aim is to label every state (l, v) of $\mathsf{T}[\mathsf{P}]$ with the set of subformulae of Φ which it satisfies (as $|\mathcal{X}| = 1$, recall that v is a single real value). For each location $l \in L$ and subformula Ψ of Φ, we construct a set $\mathsf{Sat}[l, \Psi] \subseteq \mathbb{R}_{\geq 0}$ of intervals such that $v \in \mathsf{Sat}[l, \Psi]$ if and only if $(l, v) \models \Psi$. We write $\mathsf{Sat}[l, \Psi] = \bigcup_{j=1,\ldots,k}\langle c_j; c_j'\rangle$ with $\langle \in \{[, (\} \text{ and } \rangle \in \{],)\}$. We consider intervals which conform to the following rules: for $1 \leq j \leq k$, we have $c_j < c_j'$ and

$c_j, c'_j \in \mathbb{N} \cup \{\infty\}$, and for $1 \leq j < k$, we have $c'_j < c_{j+1}$. We will see that $|\mathsf{Sat}[l, \Psi]|$ – i.e., the number of intervals corresponding to a particular location – is bounded by $|\Psi| \cdot 2 \cdot |prob|$.

The cases of obtaining the sets $\mathsf{Sat}[l, \Psi]$ for boolean operators and atomic propositions are straightforward, and therefore we concentrate on the verification of subformulae Ψ of the form $\mathbb{P}_{\bowtie \zeta}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$. Assume that we have already computed the sets $\mathsf{Sat}[\_, \_]$ for Φ_1 and Φ_2. Our aim is to compute $\mathsf{Sat}[l, \Psi]$ for each location $l \in L$.

There are several cases depending on the constraint "$\bowtie \zeta$". The equivalence $\mathbb{P}_{\leq 0}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2) \equiv \neg(\mathsf{E}\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$, which holds from the structural non-Zenoness property, can be used to reduce the "≤ 0" case to the appropriate polynomial-time labeling procedure for $\neg(\mathsf{E}\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$ on one-clock timed automata [LMS04], where the 1C-TA is obtained by converting the probabilistic choice of *prob* to nondeterministic choice. In the "≥ 1" case, the equivalence $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2) \equiv \mathsf{A}(\Phi_1 \mathsf{U}_{\sim c}(\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U} \Phi_2)))$ relies on first computing the state set satisfying $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U} \Phi_2)$, which can be handled using a qualitative PCTL model-checking algorithm, applied to a discrete TMDP built from P, $\mathsf{Sat}[l, \Phi_1]$ and $\mathsf{Sat}[l, \Phi_2]$, in time $O(|\mathsf{P}| \cdot |prob| \cdot (|\Phi_1| + |\Phi_2|))$, and second verifying the formula $\mathsf{A}(\Phi_1 \mathsf{U}_{\sim c}(\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U} \Phi_2)))$ using the aforementioned method for one-clock timed automata.

For the remaining cases, our aim is to construct a (finite) discrete TMDP $\mathsf{T}^r = (S^r, \_, \rightarrow^r, lab^r)$, which represents partially the semantic TMDP $\mathsf{T}[\mathsf{P}]$, for which the values of the functions α, β, γ and δ of the proof of Proposition 4.2 can be computed, and then use these functions to obtain the required sets $\mathsf{Sat}[\_, \Psi]$ (the initial state of T^r is irrelevant for the model-checking procedure, and is therefore omitted). The TMDP T^r will take a similar form to the region graph MDP of PTA [KNSS02], but, as in the case of the MDP $\mathsf{M}[\mathsf{P}]$ constructed in the proof of Proposition 4.1, will be of reduced size. More precisely, the size of T^r will be independent of the magnitude of the constants used in invariants and guards, and will ensure a procedure running in time polynomial in $|\mathsf{P}|$.

We now describe the construction of T^r. In the following we assume that the sets $\mathsf{Sat}[l, \Phi_i]$ contain only closed intervals (and possibly intervals of the form $[b; \infty)$) and that the guards and invariant of the PTA contain non-strict comparisons: the general case is explained in Appendix A.

Formally we let $\mathbb{C} = \{0\} \cup \mathsf{Cst}(\mathsf{P}) \cup \bigcup_{i \in \{1,2\}} \bigcup_{l \in L} \mathsf{Cst}(\mathsf{Sat}[l, \Phi_i])$, where, as in the proof of Proposition 4.1, $\mathsf{Cst}(\mathsf{P})$ is the set of constants occurring in the clock constraints of P, and where $\mathsf{Cst}(\mathsf{Sat}[l, \Phi_i])$ is the set of constants occurring as endpoints of the intervals in $\mathsf{Sat}[l, \Phi_i]$. Moreover for any right-open interval $[b; \infty)$ occurring in some $\mathsf{Sat}[l, \_]$ we add the constant $b + c + 1$ to $\mathbb{C}$. We enumerate $\mathbb{C}$ as $b_0, b_1, ..., b_M$ with $b_0 = 0$ and $b_i < b_{i+1}$ for $i < |\mathbb{C}|$. Note that $|\mathbb{C}|$ is bounded by $4 \cdot |\Psi| \cdot |prob|$.

State space of T^r: We consider first the definition of S^r, the state space of T^r. Considering the discrete TMDP corresponding to $\mathsf{T}[\mathsf{P}]$ restricted to states (l, b_i), with $b_i \in \mathbb{C}$, is sufficient to compute the values of functions α, β, γ and δ in any state (l, b_i). However, this does not allow us to deduce the value for any intermediate states in $(b_i; b_{i+1})$: indeed some probabilistic edges enabled from b_i may be disabled throughout the interval $(b_i; b_{i+1})$. Therefore, in T^r, we have to consider also (l, b_i^+) and (l, b_{i+1}^-) corresponding respectively to the leftmost and rightmost points in $(b_i; b_{i+1})$ (when $i < M$). Then S^r is defined as the set including the pairs (l, b_i) with $b_i \in \mathbb{C}$ and $b_i \models inv(l)$, and (l, b_i^+) and (l, b_{i+1}^-) with $b_i \in \mathbb{C}$, $i < M$ and $(b_i; b_{i+1}) \subseteq [\![inv(l)]\!]$. Note that the truth value of any invariant is constant over such intervals $(b_i; b_{i+1})$. Moreover note that all $\mathsf{T}[\mathsf{P}]$ states of the form (l, v) with $v \in (b_i; b_{i+1})$ satisfy the same boolean combinations of Φ_1 and Φ_2, and *enable the*

same probabilistic edges. For any $(l, g, p) \in prob$, we write $b_i^+ \models g$ (and $b_{i+1}^- \models g$) when $(b_i; b_{i+1}) \subseteq [\![g]\!]$. Similarly, we write $b_i^+ \models inv(l)$ (and $b_{i+1}^- \models inv(l)$) when $(b_i; b_{i+1}) \subseteq [\![inv(l)]\!]$. For an interval $I \subseteq \mathbb{R}_{\geq 0}$, we write $b_i^+ \in I$ and $b_{i+1}^- \in I$ when $(b_i; b_{i+1}) \subseteq I$. We also consider the ordering $b_0 < b_0^+ < b_1^- < b_1 < b_1^+ < \cdots < b_M^- < b_M < b_M^+$.

Transitions of T^r: We now define the set $\rightarrow^r$ of transitions of T^r as the smallest set such that $((l, \lambda), d, \nu) \in \rightarrow^r$, where $\lambda \in \{b_i^-, b_i, b_i^+\}$ for some $b_i \in \mathbb{C}$, if there exists $\lambda' \geq \lambda$, where $\lambda' \in \{b_j^-, b_j, b_j^+\}$ for some $b_j \in \mathbb{C}$, and $(l, g, p) \in prob$ such that:

- $d = b_j - b_i$, $\lambda' \models g$, and both $\lambda'' \models inv(l)$ and $\lambda'' \subseteq \mathsf{Sat}[l, \Phi_1] \setminus \mathsf{Sat}[l, \Phi_2]$ for any $\lambda \leq \lambda'' \leq \lambda'$;
- for each $(X, l') \in \mathsf{support}(p)$, we have $0 \models inv(l')$ if $X = \{x\}$, and $\lambda' \models inv(l')$ if $X = \emptyset$;
- for each $(l', \lambda'') \in S^r$, we have $\nu(l', \lambda'') = \nu_0(l', \lambda'') + \nu_\lambda(l', \lambda'')$, where $\nu_0(l', \lambda'') = p(l', \{x\})$ if $\lambda'' = [0, 0]$ and $\nu_0(l', \lambda'') = 0$ otherwise, and $\nu_\lambda(l', \lambda'') = p(l', \emptyset)$ if $\lambda'' = \lambda'$ and $\nu_\lambda(l', \lambda'') = 0$ otherwise.

Labelling function of T^r: To define lab^r, for a state (l, b_i), we let $a_{\Phi_j} \in lab^r(l, b_i)$ if and only if $b_i \in \mathsf{Sat}[l, \Phi_j]$, for $j \in \{1, 2\}$. The states (l, b_i^+) and (l, b_{i+1}^-) are labeled depending on the truth value of the Φ_j's in the interval $(b_i; b_{i+1})$: if $(b_i; b_{i+1}) \subseteq \mathsf{Sat}[l, \Phi_j]$, then $a_{\Phi_j} \in lab^r(l, b_i^+)$ and $a_{\Phi_j} \in lab^r(l, b_{i+1}^-)$. Note that, given the "closed intervals" assumption made on $\mathsf{Sat}[l, \Phi_j]$, we have $lab^r(l, b_i^+) \subseteq lab^r(l, b_i)$ and $lab^r(l, b_{i+1}^-) \subseteq lab^r(l, b_i)$.

Note that the fact that P is structurally non-Zeno means that T^r is structurally non-Zeno. The size of T^r is in $O(|\mathsf{P}|^2 \cdot |\Psi|)$.

Now we can apply the algorithms defined in the proof of Proposition 4.2 and obtain the value of the coefficients α, β, γ or δ for the states of T^r. Our next task is to define functions $\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\delta} : S \rightarrow \mathbb{R}_{\geq 0}$, where S is the set of states of $\mathsf{T}[\mathsf{P}]$, which are analogues of α, β, γ or δ defined on $\mathsf{T}[\mathsf{P}]$. Our intuition is that we are now considering an infinite-state 2-player game with players P_n and P_p, as in the proof of Proposition 4.2, over the state space of $\mathsf{T}[\mathsf{P}]$. Consider location $l \in L$. For $b \in \mathbb{C}$, we have $\overline{\alpha}(l, b) = \alpha(l, b)$, $\overline{\beta}(l, b) = \beta(l, b)$, $\overline{\gamma}(l, b) = \gamma(l, b)$ and $\overline{\delta}(l, b) = \delta(l, b)$. For intervals of the form $(b_i; b_{i+1})$, the functions $\overline{\alpha}$ and $\overline{\delta}$ decrease (with slope -1) throughout the interval, because, for all states of the interval, the optimal choice of player P_n is to delay as much as possible inside any interval. Hence, the value $\overline{\alpha}(l, v)$ for $v \in (b_i; b_{i+1})$ is defined entirely by $\alpha(l, b_{i+1}^-)$ as $\overline{\alpha}(l, v) = \alpha(l, b_{i+1}^-) + b_{i+1} - v$. Similarly, $\overline{\delta}(l, v) = \delta(l, b_{i+1}^-) + b_{i+1} - v$.

Next we consider the values of $\overline{\beta}$ and $\overline{\gamma}$ over intervals $(b_i; b_{i+1})$. In this case, the functions will be constant over a portion of the interval (possibly an empty portion, or possibly the entire interval), then decreasing with slope -1. The constant part corresponds to those states in which the optimal choice of player P_n is to take a probabilistic edge, whereas the decreasing part corresponds to those states in which it is optimal for player P_n to delay until the end of the interval. The value $\overline{\beta}(l, v)$ for $v \in (b_i; b_{i+1})$ is defined both by $\beta(l, b_i^+)$ and $\beta(l, b_{i+1}^-)$ as $\overline{\beta}(l, v) = \beta(l, b_i^+)$ if $b_i < v \leq b_{i+1} - (\beta(l, b_i^+) - \beta(l, b_{i+1}^-))$, and as $\overline{\beta}(l, v) = \beta(l, b_{i+1}^-) - (v - \beta(l, b_i^+))$ otherwise. An analogous definition holds also for $\overline{\gamma}$.

From the functions $\overline{\alpha}$, $\overline{\beta}$, $\overline{\gamma}$ and $\overline{\delta}$ defined above, it becomes possible to define $\mathsf{Sat}[l, \Psi]$ by keeping in this set of intervals only the parts satisfying the thresholds $\leq c$, $> c$, $\geq c$ and $< c$, respectively, as in the proof of Proposition 4.2. We can show that the number of intervals in $\mathsf{Sat}[l, \Psi]$ is bounded by $2 \cdot |\Psi| \cdot |prob|$. For the case in which a function $\overline{\alpha}$, $\overline{\beta}$, $\overline{\gamma}$ or $\overline{\delta}$ is decreasing throughout an interval, then an interval in $\mathsf{Sat}[l, \Phi_1]$ which corresponds to several consecutive intervals in T^r can provide at most one (sub)interval in $\mathsf{Sat}[l, \Psi]$,

because the threshold can cross at most once the function in at most one interval. For the case in which a function $\overline{\beta}$ or $\overline{\gamma}$ combines a constant part and a part with slope -1 within an interval, the threshold can cross the function in several intervals $(b_i; b_{i+1})$ contained in a common interval of $\mathsf{Sat}[l, \Phi_1]$. However, such a cut is due to a guard $x \geq k$ of a given transition, and thus the number of cuts in bounded by $|prob|$. Moreover a guard $x \leq k$ may also add an interval. Thus the number of new intervals in $\mathsf{Sat}[q, \Psi]$ is bounded by $2 \cdot |prob|$.

In addition to these cuts, any interval in $\mathsf{Sat}[l, \Phi_2]$ may provide an interval in $\mathsf{Sat}[l, \Psi]$. This gives the $2 \cdot |\Psi| \cdot |prob|$ bound for the size of $\mathsf{Sat}[l, \Psi]$. $\square$

Corollary 4.4. *The* PTCTL$^{0/1}[\leq, \geq]$ *model-checking problem for 1C-PTA is P-complete.*

4.3. **Model checking** PTCTL$^{0/1}$ **on 1C-PTA.** We now consider the problem of model-checking PTCTL$^{0/1}$ properties on 1C-PTA. An EXPTIME algorithm for this problem exists by the definition of an MDP analogous to the region graph used in non-probabilistic timed automata verification [KNSS02]. We now show that the problem is also EXPTIME-hard by the following three steps. First we introduce *countdown games*, which are a simple class of turn-based 2-player games with discrete timing, and show that the problem of deciding the winner in a countdown game is EXPTIME-complete. Secondly, we reduce the countdown game problem to the PTCTL$^{0/1}$ model-checking problem on TMDPs. Finally, we adapt the reduction to TMDPs to reduce also the countdown game problem to the PTCTL$^{0/1}$ model-checking problem on 1C-PTA.

A *countdown game* $\mathcal{C}$ consists of a weighted graph (S, T), where S is the set of *states* and $\mathsf{T} \subseteq \mathsf{S} \times \mathbb{N} \setminus \{0\} \times \mathsf{S}$ is the *transition relation*. If $\mathsf{t} = (\mathsf{s}, d, \mathsf{s}') \in \mathsf{T}$ then we say that the *duration* of the transition t is d. A configuration of a countdown game is a pair (s, c), where $\mathsf{s} \in \mathsf{S}$ is a state and $c \in \mathbb{N}$. A *move* of a countdown game from a configuration (s, c) is performed in the following way: first player 1 chooses a number d, such that $0 < d \leq c$ and $(\mathsf{s}, d, \mathsf{s}') \in \mathsf{T}$, for some state $\mathsf{s}' \in \mathsf{S}$; then player 2 chooses a transition $(\mathsf{s}, d, \mathsf{s}') \in \mathsf{T}$ of duration d. The resulting new configuration is $(\mathsf{s}', c - d)$. There are two types of *terminal* configurations, i.e., configurations (s, c) in which no moves are available. If $c = 0$ then the configuration (s, c) is terminal and is a *winning configuration for player 1*. If for all transitions $(\mathsf{s}, d, \mathsf{s}') \in \mathsf{T}$ from the state s, we have that $d > c$, then the configuration (s, c) is terminal and it is a *winning configuration for player 2*. The algorithmic problem of *deciding the winner* in countdown games is, given a weighted graph (S, T) and a configuration (s, c), where all the durations of transitions in (S, T) and the number c are given in binary, to determine whether player 1 has a strategy to reach a winning configuration, regardless of the strategy of player 2, from the configuration (s, c). If the state from which the game is started is clear from the context then we sometimes specify the initial configuration by giving the number c alone.

Theorem 4.5. *Deciding the winner in countdown games is* EXPTIME-*complete.*

Proof sketch. Observe that every configuration of a countdown game played from a given initial configuration can be written down in polynomial space and every move can be computed in polynomial time; hence the winner in the game can be determined by a straightforward alternating PSPACE algorithm. Therefore the problem is in EXPTIME because APSPACE = EXPTIME.

We now prove EXPTIME-hardness by a reduction from the problem of the acceptance of a word by a linearly-bounded alternating Turing machine [CKS81]. Let $M =$

$(\Sigma, Q, q_0, q_{acc}, Q_\exists, Q_\forall, \Delta)$ be an alternating Turing machine, where Σ is a finite alphabet, $Q = Q_\exists \cup Q_\forall$ is a finite set of states partitioned into existential states $Q_\exists$ and universal states $Q_\forall$, $q_0 \in Q$ is an initial state, $q_{acc} \in Q$ is an accepting state, and $\Delta \subseteq Q \times \Sigma \times Q \times \Sigma \times \{L, R\}$ is a transition relation. Let us explain the interpretation of elements of the transition relation. Let $t = (q, \sigma, q', \sigma', D) \in \Delta$ be a transition. If machine M is in state $q \in Q$ and its head reads letter $\sigma \in \Sigma$, then it rewrites the contents of the current cell with the letter σ', it moves the head in direction D (either left if $D = L$, or right if $D = R$), and it changes its state to q'.

Let $G > 2 \cdot |Q \times \Sigma|$ be an integer constant and let $w \in \Sigma^n$ be an input word. Without loss of generality, we can assume that the alternating Turing machine M uses exactly n tape cells when started on the word w, and hence a configuration of machine M is a word $\mathbf{b}_0 \mathbf{b}_1 \cdots \mathbf{b}_{n-1} \in (\Sigma \cup Q \times \Sigma)^n$. Let $\langle \cdot \rangle : (\Sigma \cup Q \times \Sigma) \to \{0, 1, \ldots, G-1\}$ be an injection. For every $\mathbf{a} \in \Sigma \cup Q \times \Sigma$, it is convenient to think of $\langle \mathbf{a} \rangle$ as a G-ary digit, and we can encode a configuration $u = \mathbf{b}_0 \mathbf{b}_1 \cdots \mathbf{b}_{n-1} \in (\Sigma \cup Q \times \Sigma)^n$ of machine M as the number $N(u) = \sum_{i=0}^{n-1} \langle \mathbf{b}_i \rangle \cdot G^i$.

We first define countdown games which have the role of checking the contents of the tape; these countdown games will be used as gadgets later in the overall reduction. Let $i \in \mathbb{N}$, $0 \le i < n$, be a tape cell position, and let $\mathbf{a} \in \Sigma \cup Q \times \Sigma$. We define a countdown game $\text{Check}^{i,\mathbf{a}}$, such that for every configuration $u = \mathbf{b}_0 \cdots \mathbf{b}_{n-1}$ of machine M, player 1 has a winning strategy from the configuration $(\mathbf{s}_0^{i,\mathbf{a}}, N(u))$ of the countdown game $\text{Check}^{i,\mathbf{a}}$ if and only if $\mathbf{b}_i = \mathbf{a}$. The game $\text{Check}^{i,\mathbf{a}}$ has states $\{\mathbf{s}_0^{i,\mathbf{a}}, \ldots, \mathbf{s}_n^{i,\mathbf{a}}\}$, and for every k, $0 \le k < n$, we have a transition $(\mathbf{s}_k^{i,\mathbf{a}}, d, \mathbf{s}_{k+1}^{i,\mathbf{a}}) \in \mathtt{T}$, if:

$$d = \begin{cases} \langle \mathbf{a} \rangle \cdot G^k & \text{if } k = i, \\ \langle \mathbf{b} \rangle \cdot G^k & \text{if } k \ne i \text{ and } \mathbf{b} \in \Sigma \cup S \times \Sigma. \end{cases}$$

There are no transitions from the state $\mathbf{s}_n^{i,\mathbf{a}}$. Observe that if $\mathbf{b}_i = \mathbf{a}$ then the winning strategy for player 1 in game $\text{Check}^{i,\mathbf{a}}$ from $N(u)$ is to choose the transitions $(\mathbf{s}_k^{i,\mathbf{a}}, \mathbf{b}_k \cdot G^k, \mathbf{s}_{k+1}^{i,\mathbf{a}})$, for all k, $0 \le k < n$. If, however, $\mathbf{b}_i \ne \mathbf{a}$ then there is no way for player 1 to count down from $N(u)$ to 0 in the game $\text{Check}^{i,\mathbf{a}}$.

Now we define a countdown game $\mathcal{C}_M$, such that machine M accepts a word $w = \sigma_0 \sigma_1 \ldots \sigma_{n-1}$ if and only if player 1 has a winning strategy in $\mathcal{C}_M$ from configuration $(q_0, N(u))$, where $u = (q_0, \sigma_0) \sigma_1 \ldots \sigma_{n-1}$ is the initial configuration of tape contents of machine M with input w. The main part of the countdown game $\mathcal{C}_M$ is a gadget that allows the countdown game to simulate one step of the Turing machine M. Note that one step of a Turing machine makes only local changes to the configuration of the machine: if the configuration is of the form $u = \mathbf{a}_0 \ldots \mathbf{a}_{n-1} = \sigma_0 \ldots \sigma_{i-1}(q, \sigma_i)\sigma_{i+1} \ldots \sigma_{n-1}$, then performing one step of M can only change entries in positions $i-1$, i, or $i+1$ of the tape. For every tape position i, $0 \le i < n$, for every triple $\tau = (\sigma_{i-1}, (q, \sigma_i), \sigma_{i+1}) \in \Sigma \times (Q \times \Sigma) \times \Sigma$, and for every transition $t = (q, \sigma, q', \sigma', D) \in \Delta$ of machine M, we now define the number $d_t^{i,\tau}$, such that if $\sigma_i = \sigma$ and performing transition t at position i of configuration u yields configuration $u' = \mathbf{b}_0 \ldots \mathbf{b}_{n-1}$, then $N(u) - d_t^{i,\tau} = N(u')$. For example, assume that $i > 0$ and that $D = L$; from the above comment about locality of Turing machine transitions we have that $\mathbf{b}_k = \mathbf{a}_k = \sigma_k$, for all $k \notin \{i-1, i, i+1\}$ and $\mathbf{b}_{i+1} = \mathbf{a}_{i+1} = \sigma_{i+1}$. Moreover we

have that $\mathbf{b}_{i-1} = (q', \sigma_{i-1})$, and $\mathbf{b}_i = \sigma'$. We define $d_t^{i,\tau}$ as follows:

$$d_t^{i,\tau} = (\langle \mathbf{b}_{i-1} \rangle - \langle \mathbf{a}_{i-1} \rangle) \cdot G^{i-1} + (\langle \mathbf{b}_i \rangle - \langle \mathbf{a}_i \rangle) \cdot G^i$$
$$= (\langle (q', \sigma_{i-1}) \rangle - \langle \sigma_{i-1} \rangle) \cdot G^{i-1} + (\langle \sigma' \rangle - \langle (q, \sigma_i) \rangle) \cdot G^i.$$

The gadget for simulating one transition of Turing machine M from a state $q \in Q \backslash \{q_{acc}\}$ has three layers. In the first layer, from a state $q \in Q \backslash \{q_{acc}\}$, player 1 chooses a pair (i, τ), where i, $0 \leq i < n$, is the position of the tape head, and $\tau = (\mathbf{a}, \mathbf{b}, \mathbf{c}) \in \Sigma \times (Q \times \Sigma) \times \Sigma$ is his guess for the contents of tape cells $i-1$, i, and $i+1$. In this way the state (q, i, τ) of the gadget is reached, where the duration of this transition is 0. Intuitively, in the first layer player 1 has to declare that he knows the position i of the head in the current configuration as well as the contents $\tau = (\mathbf{a}, \mathbf{b}, \mathbf{c})$ of the three tape cells in positions $i-1$, i, and $i+1$. In the second layer, in a state (q, i, τ) player 2 chooses between four successor states: the state $(q, i, \tau, *)$ and the three subgames $\mathrm{Check}^{i-1,\mathbf{a}}$, $\mathrm{Check}^{i,\mathbf{b}}$, and $\mathrm{Check}^{i+1,\mathbf{c}}$. The four transitions are of duration 0. Intuitively, in the second layer player 2 verifies that player 1 declared correctly the contents of the three tape cells in positions $i-1$, i, and $i+1$. Finally, in the third layer, if $q \in Q_\exists$ (respectively, $q \in Q_\forall$), then from a state $(q, i, \tau, *)$ player 1 (respectively, player 2) chooses a transition $t = (q, \sigma, q', \sigma', D)$ of machine M, such that $\mathbf{b} = (q, \sigma)$, reaching the state $q' \in Q$ of the gadget, with a transition of duration $d_t^{i,\tau}$.

Note that the gadget described above violates some conventions that we have adopted for countdown games. Observe that durations of some transitions in the gadget are 0 and the duration $d_t^{i,\tau}$ may even be negative, while in the definition of countdown games we required that durations of all transitions are positive. In order to correct this we add the number G^n to the durations of all transitions described above. This change requires a minor modification to the subgames $\mathrm{Check}^{i,\mathbf{a}}$: we add an extra transition $(\mathbf{s}_n^{i,\mathbf{a}}, G^n, s_n^{i,\mathbf{a}})$. We need this extra transition because instead of starting from $(q_0, N(u))$ as the initial configuration of the countdown game $\mathcal{C}_M$, where u is the initial configuration of M running on w, we start from the configuration $(q_0, G^{3n} + N(u))$. In this way the countdown game can perform a simulation of at least G^n steps of machine M; note that G^n is an upper bound on the number of all configurations of machine M.

Without loss of generality, we can assume that whenever the alternating Turing machine M accepts an input word w then it finishes its computation with blanks in all tape cells, its head in position 0, and in the unique accepting state q_{acc}; we write u_{acc} for this unique accepting configuration of machine M. Moreover, assume that there are no transitions from the accepting state q_{acc} in machine M. In order to complete the definition of the countdown game G_M, we add a transition of duration $N(u_{acc})$ from the state q_{acc} of game $\mathcal{C}_M$. $\square$

Proposition 4.6. *The* $\mathrm{PTCTL}^{0/1}$ *model-checking problem for structurally non-Zeno discrete TMDPs is* EXPTIME-*complete.*

Proof. An EXPTIME algorithm can be obtained by employing the algorithms of [LS05]. We now prove EXPTIME-hardness of $\mathrm{PTCTL}^{0/1}$ model checking on discrete TMDPs by a reduction from countdown games. Let $\mathcal{C} = (\mathtt{S}, \mathtt{T})$ be a countdown game and $(\overline{\mathbf{s}}, c)$ be its initial configuration. We construct a TMDP $\mathsf{T}_{\mathcal{C},(\overline{\mathbf{s}},c)} = (S, \overline{s}, \rightarrow, lab)$ such that player 1 wins $\mathcal{C}$ from $(\overline{\mathbf{s}}, c)$ if and only if $\mathsf{T}_{\mathcal{C},(\overline{\mathbf{s}},c)} \models \neg \mathbb{P}_{<1}(\mathsf{F}_{=c} \mathtt{true})$. Let $S = \mathtt{S}$ and $\overline{s} = \overline{\mathbf{s}}$. We define $\rightarrow$ to be the smallest set satisfying the following: for each $\mathbf{s} \in \mathtt{S}$ and $d \in \mathbb{N}_{>0}$, if $(\mathbf{s}, d, \mathbf{s}') \in \mathtt{T}$ for some $\mathbf{s}' \in \mathtt{T}$, we have $(\mathbf{s}, d, \nu) \in \rightarrow$, where ν is an arbitrary distribution

over S such that $\mathsf{support}(\nu) = \{ \mathsf{s}' \mid (\mathsf{s}, d, \mathsf{s}') \in \mathsf{T} \}$. The labelling condition *lab* is arbitrary. Then we can show that player 1 wins $\mathcal{C}$ from the configuration $(\overline{\mathsf{s}}, c)$ if and only if there exists an adversary of $\mathsf{T}_{\mathcal{C},(\overline{\mathsf{s}},c)}$ such that a state is reached from $\bar{s} = \overline{\mathsf{s}}$ after exactly c time units with probability 1. The latter is equivalent to $\bar{s} \models \neg \mathbb{P}_{<1}(\mathsf{F}_{=c}\mathtt{true})$. $\square$

We now show that the proof of Proposition 4.6 can be adapted to show the EXPTIME-completeness of the analogous model-checking problem on 1C-PTA.

Theorem 4.7. *The* PTCTL$^{0/1}$ *model-checking problem for 1C-PTA is* EXPTIME-*complete.*

Proof. Recall that there exists an EXPTIME algorithm for model-checking PTCTL$^{0/1}$ properties on structurally non-Zeno PTA [KNSS02]; hence, it suffices to show EXPTIME-hardness for PTCTL$^{0/1}$ and 1C-PTA. Let $\mathcal{C}$ be a countdown game with an initial configuration $(\overline{\mathsf{s}}, c)$. We construct the 1C-PTA $\mathsf{P}^{1C}_{\mathcal{C},(\overline{\mathsf{s}},c)} = (L, \bar{l}, \{x\}, inv, prob, \mathcal{L})$ which simulates the behaviour of the TMDP $\mathsf{T}_{\mathcal{C},(\overline{\mathsf{s}},c)}$ of the proof of Proposition 4.6 in the following way. Each state $\mathsf{s} \in \mathsf{S}$ of $\mathsf{T}_{\mathcal{C},(\overline{\mathsf{s}},c)}$ corresponds to two distinct locations l^1_{s} and l^2_{s} of $\mathsf{P}^{1C}_{\mathcal{C},(\overline{\mathsf{s}},c)}$. Let $L^i = \{l^i_{\mathsf{s}} \mid \mathsf{s} \in \mathsf{S}\}$ for $i \in \{1, 2\}$, let $L = L^1 \cup L^2$, and let $\bar{l} = l^1_{\overline{\mathsf{s}}}$. For every transition $(\mathsf{s}, d, \nu) \in \rightarrow$ of $\mathsf{T}_{\mathcal{C},(\overline{\mathsf{s}},c)}$, we have the probabilistic edges $(l^1_{\mathsf{s}}, x = 0, p^1), (l^2_{\mathsf{s}}, x = d, p^2) \in prob$, where $p^1(\{x\}, l^2_{\mathsf{s}}) = 1$, and $p^2(\{x\}, l^1_{\mathsf{s}'}) = \nu(\mathsf{s}')$ for each location s'. For each state $\mathsf{s} \in \mathsf{S}$, let $inv(l^1_{\mathsf{s}}) = (x \leq 0)$ and $inv(l^2_{\mathsf{s}}) = \mathtt{true}$. Therefore the PTA $\mathsf{P}^{1C}_{\mathcal{C},(\overline{\mathsf{s}},c)}$ moves from the location l^1_{s} to l^2_{s} instantaneously. Locations in L^1 are labelled by the atomic proposition a, whereas locations in L^2 are labelled by $\emptyset$. Then we can observe that $\mathsf{P}^{1C}_{\mathcal{C},(\overline{\mathsf{s}},c)} \models \neg \mathbb{P}_{<1}(\mathsf{F}_{=c}a)$ if and only if $\mathsf{T}_{\mathcal{C},(\overline{\mathsf{s}},c)} \models \neg \mathbb{P}_{<1}(\mathsf{F}_{=c}\mathtt{true})$. As the latter problem has been shown to be EXPTIME-hard in the proof of Proposition 4.6, we conclude that model checking PTCTL$^{0/1}$ on 1C-PTA is also EXPTIME-hard. $\square$

In Figure 2, we illustrate the transformation from countdown games to TMDP, then to 1C-PTA, for a fragment of a countdown game. For simplicity, we omit guards of the form $x = 0$ and invariant conditions of the form $\mathtt{true}$.

5. MODEL CHECKING TWO-CLOCKS PROBABILISTIC TIMED AUTOMATA

We now show EXPTIME-completeness of the simplest problems that we consider on 2C-PTA.

Theorem 5.1. *Qualitative probabilistic reachability problems for 2C-PTA are* EXPTIME-*complete.*

Proof. EXPTIME algorithms exist for probabilistic reachability problems on structurally non-Zeno PTA [KNSS02], and therefore it suffices to show EXPTIME-hardness. We proceed by reduction from deciding the winner in countdown games. Let $\mathcal{C}$ be a countdown game with initial configuration $(\overline{\mathsf{s}}, c)$, and let $\mathsf{P}^{1C}_{\mathcal{C},(\overline{\mathsf{s}},c)} = (L, \bar{l}, \{x\}, inv, prob, \mathcal{L})$ be the 1C-PTA constructed in the proof of Theorem 4.7. We define the 2C-PTA $\mathsf{P}^{2C}_{\mathcal{C},(\overline{\mathsf{s}},c)} = (L \cup \{l^{\star}\}, \bar{l}, \{x, y\}, inv', prob', \mathcal{L}')$ from $\mathsf{P}^{1C}_{\mathcal{C},(\overline{\mathsf{s}},c)}$ in the following way. The set of probabilistic edges $prob'$ is obtained by adding to $prob$ the following: for each location $l \in L^1$, we extend the set of outgoing probabilistic edges of l with $(l, y = c, p^{l^{\star}})$, where $p^{l^{\star}}(\emptyset, l^{\star}) = 1$; we also add $(l^{\star}, \mathtt{true}, p^{l^{\star}})$ to $prob'$. For each $l \in L$, let $inv'(l) = inv(l)$, and let $inv'(l^{\star}) = \mathtt{true}$. Finally, we let $\mathcal{L}'(l^{\star}) = a$, and $\mathcal{L}(l) = \emptyset$ for all $l \in L$. Then $\mathsf{P}^{2C}_{\mathcal{C},(\overline{\mathsf{s}},c)} \models \neg \mathbb{P}_{<1}(\mathsf{F}a)$ if and only
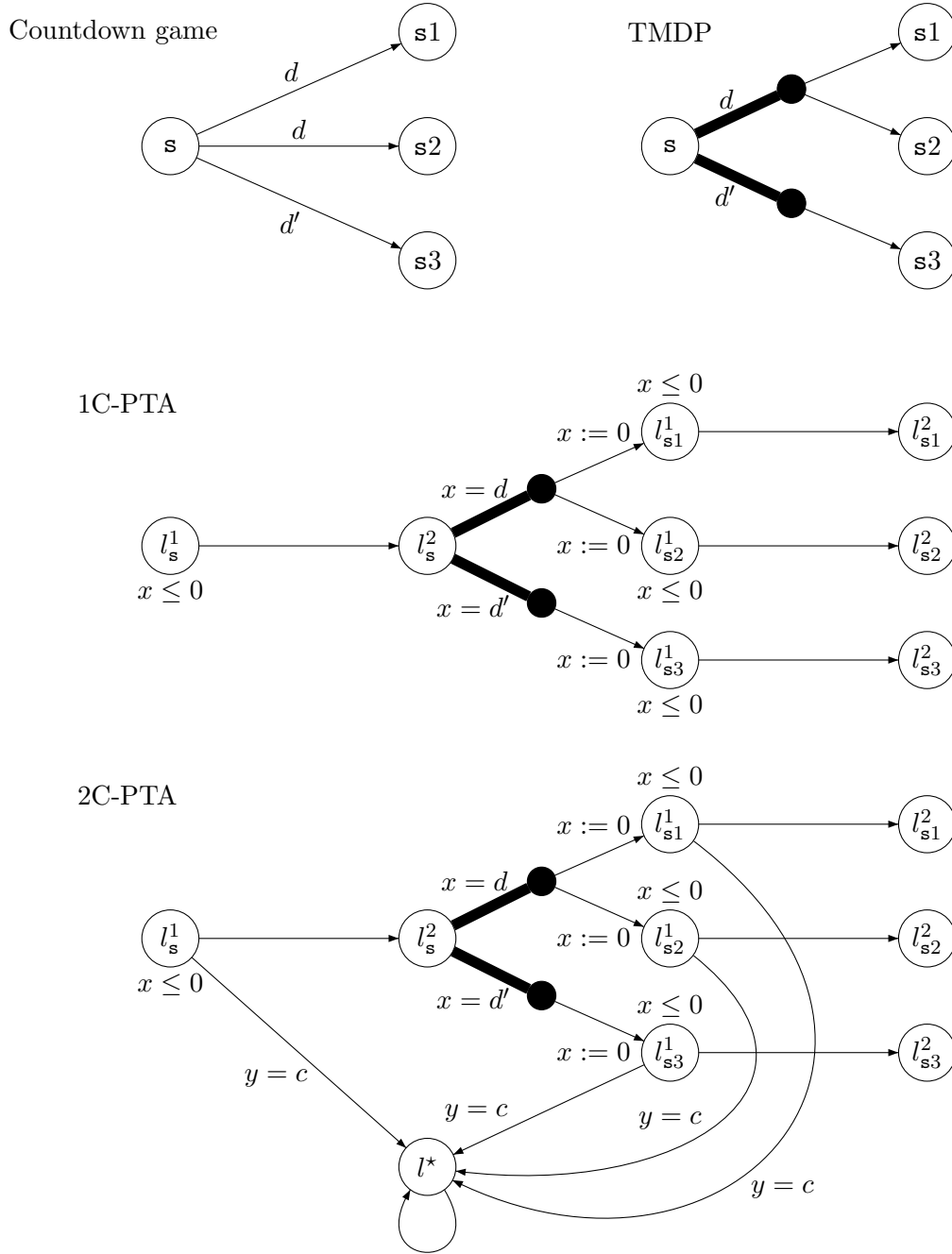
Figure 2: Reduction from countdown games

if $\mathsf{P}^{1C}_{\mathcal{C},(\overline{\mathbf{s}},c)} \models \neg\mathbb{P}_{<1}(\mathsf{F}_{=c}a)$. The EXPTIME-hardness of the latter problem has been shown in the proof of Theorem 4.7, and hence checking qualitative probabilistic reachability properties such as $\neg\mathbb{P}_{<1}(\mathsf{F}a)$ on 2C-PTA is EXPTIME-hard. $\qquad\square$

In Figure 2 we illustrate the reduction from countdown games to 2C-PTA (via the reduction to TMDPs and 1C-PTA).

Corollary 5.2. *The* PCTL*,* PTCTL$^{0/1}[\leq, \geq]$*,* PTCTL$^{0/1}$*,* PTCTL$[\leq, \geq]$ *and* PTCTL *model-checking problems for 2C-PTA are* EXPTIME*-complete.*

6. FORWARD REACHABILITY FOR ONE-CLOCK PROBABILISTIC TIMED AUTOMATA

Model-checking tools for non-probabilistic timed automata such as UPPAAL [BDL$^+$06] are generally based on algorithms for *forward reachability* through the state space: such algorithms start from the initial state and explore the state space by executing transitions either in a depth-first or breadth-first manner, and representing sets of clock valuations symbolically using *zones*. Forward reachability algorithms can be used for verifying reachability properties, such as "the location *error* is reachable from the initial state".

We recall that the zone-based forward reachability approach has been adapted for PTA by Kwiatkowska et al. [KNSS02], and can be used to reason about the maximal probability of reaching a certain set of locations. More precisely, an (untimed) MDP is constructed by exploring the state space of the PTA from its initial state. Then the maximal probability of reaching a set of locations is computed on the MDP. The appeal of this approach is its practical applicability [DKN04]. A disadvantage of the approach is that, in general, it can be used only to obtain an *upper* bound on the maximal probability of reaching a set of locations of a PTA, rather than the actual maximal probability of reaching the locations. In particular, Kwiatkowska et al. [KNSS02] present an example of a 2C-PTA in which the forward reachability approach does not compute the actual maximal probability of reaching a set of locations.

In this section, we consider the application of the forward reachability approach of Kwiatkowska et al. [KNSS02] to 1C-PTA, and show that the maximal and minimal probabilities computed on the untimed MDP corresponds to the actual maximal and minimal probabilities of reaching a set of locations of the 1C-PTA.[2]

First we introduce some notation. Consider the 1C-PTA $\mathsf{P} = (L, \bar{l}, \{x\}, inv, prob, \mathcal{L})$, which we assume to be fixed throughout this section. As in the proof of Proposition 4.1, we use $\mathbb{B} = \mathsf{Cst}(\mathsf{P}) \cup \{0\}$ to refer to the set of constants used in the guards and invariants of P (and 0). Let $\mathcal{I}_{\mathsf{FR}}$ be the set of intervals of the form $\langle b; b' \rangle$, where $b \in \mathbb{B}$, $b' \in \mathbb{B} \cup \{\infty\}$, $\langle \in \{(, [\}$ and $\rangle \in \{),]\}$. The aim of forward exploration is to compute state sets represented by pairs of the form (l, I), where $l \in L$ is a location and $I \in \mathcal{I}_{\mathsf{FR}}$ is an interval of the above form. The pair (l, I) represents all states (l, v) of $\mathsf{T}[\mathsf{P}]$ such that $v \in I$.

We define the operator post, which maps a location-interval pair, a probabilistic edge, a reset set and a location, to a location-interval pair. Intuitively, post returns the set of states obtained after executing a probabilistic edge (including making the probabilistic choice concerning the target location and clock reset) and then letting time pass. First consider a clock constraint $\psi \in CC(\{x\})$, and recall that $[\![\psi]\!] = \{v \in \mathbb{R}_{\geq 0} \mid v \models \psi\}$. By definition $[\![\psi]\!] \in \mathcal{I}_{\mathsf{FR}}$. For all $I, I' \in \mathcal{I}_{\mathsf{FR}}$, note that $I \cap I' \in \mathcal{I}_{\mathsf{FR}}$. Furthermore, let $I_l^{\uparrow} = \langle b; \infty) \cap [\![inv(l)]\!]$, and recall that $I[\{x\} := 0] = [0; 0]$ and $I[\emptyset := 0] = I$. Let $(l, I) \in L \times \mathcal{I}_{\mathsf{FR}}$, let $(l, g, p) \in prob$, and let $(X, l') \in \mathsf{support}(p)$. Then $\mathsf{post}((l, I), (l, g, p), X, l') = (l', (([\![g]\!] \cap I)[X := 0])_{l'}^{\uparrow})$.

---

[2]Readers familiar with Kwiatkowska et al. [KNSS02] will note that the presentation below is simplified with regard to that for PTA with an arbitrary number of clocks. In particular, to ease notation, we consider that forward reachability can consider states reached *after* reaching the target set of locations.

We now proceed to define formally an untimed MDP, the states of which are intervals of the form $(l, I) \in L \times \mathcal{I}_{\mathsf{FR}}$ and which are obtained by forward exploration from the initial state of P. The probabilistic transition relation of the untimed MDP is derived from the probabilistic edge relation of P.

Definition 6.1. The *forward reachability MDP* of the PTA P is the untimed MDP $\mathsf{FR}[\mathsf{P}] = (S_{\mathsf{FR}}, \bar{s}_{\mathsf{FR}}, \rightarrow_{\mathsf{FR}}, lab_{\mathsf{FR}})$ where:

- $S_{\mathsf{FR}} \subseteq L \times \mathcal{I}_{\mathsf{FR}}$ is the least set of location-interval pairs such that:
$$\{(\bar{l}, [0;0]_{\bar{l}}^{\uparrow})\} \cup \bigcup_{(l,I)\in S_{\mathsf{FR}}} \bigcup_{(l,g,p)\in prob} \bigcup_{(X,l')\in \mathsf{support}(p)} \mathsf{post}((l,I),(l,g,p),X,l') \subseteq S_{\mathsf{FR}} .$$

- $\bar{s}_{\mathsf{FR}} = (\bar{l}, [0,0]_{\bar{l}}^{\uparrow})$ is the initial state.

- $\rightarrow_{\mathsf{FR}}$ is the least set such that $((l,I),\rho) \in \rightarrow_{\mathsf{FR}}$ if there exists a probabilistic edge $(l,g,p) \in prob$ such that:
  (1) $I \cap [\![g]\!] \neq \emptyset$;
  (2) for any $(X,l') \in \{\{x\},\emptyset\} \times L$, we have that $p(X,l') > 0$ implies $(I \cap [\![g]\!])[X := 0] \cap [\![inv(l')]\!] \neq \emptyset$;
  (3) for any $(l',I') \in S_{\mathsf{FR}}$, we have that $\rho(l',I') = \rho_0(l',I') + \rho_I(l',I')$, where $\rho_0(l',I') = p(\{x\},l')$ if $(l',I') = \mathsf{post}((l,I),(l,g,p),\{x\},l')$ and $\rho_0(l',I') = 0$ otherwise, and where $\rho_I(l',I') = p(\emptyset,l')$ if $(l',I') = \mathsf{post}((l,I),(l,g,p),\emptyset,l')$ and $\rho_I(l',I') = 0$ otherwise.

- lab_{FR} is such that $lab_{\mathsf{FR}}(l,I) = \mathcal{L}(l)$ for each state $(l,I) \in S_{\mathsf{FR}}$.

We now show that reachability properties can be verified on $\mathsf{FR}[\mathsf{P}]$. The overall proof of this results proceeds by relating $\mathsf{FR}[\mathsf{P}]$ to the untimed MDP $\mathsf{M}[\mathsf{P}]$ of Proposition 4.1, which we have established can be used to verify reachability properties (because the set of reachability properties is a subset of PCTL). Recall the definition of the set of intervals $\mathcal{I}_{\mathbb{B}}$ and the untimed MDP $\mathsf{M}[\mathsf{P}] = (S_{\mathsf{M}}, \bar{s}_{\mathsf{M}}, \rightarrow_{\mathsf{M}}, lab_{\mathsf{M}})$ of Proposition 4.1. We define the function $\mathsf{1stInt} : \mathcal{I}_{\mathsf{FR}} \to \mathcal{I}_{\mathbb{B}}$ in the following way: given $I \in \mathcal{I}_{\mathsf{FR}}$, let $\mathsf{1stInt}(I) = \min\{B \in \mathcal{I}_{\mathbb{B}} \mid B \subseteq I\}$. We define a restricted version of $\mathsf{M}[\mathsf{P}]$, namely $\mathsf{1st}[\mathsf{P}] = (S_{\mathsf{1st}}, \bar{s}_{\mathsf{M}}, \rightarrow_{\mathsf{1st}}, lab_{\mathsf{M}})$, where $S_{\mathsf{1st}} = \{(l, \mathsf{1stInt}(I)) \mid (l,I) \in S_{\mathsf{FR}}\}$, and where $\rightarrow_{\mathsf{1st}} \subseteq \rightarrow_{\mathsf{M}}$ is defined as the least set such that $((l,B),\nu) \in \rightarrow_{\mathsf{1st}}$ if conditions (1), (2) and (3) of the definition of $\rightarrow_{\mathsf{M}}$ are satisfied, and additionally (4) $B = \mathsf{1stInt}(I)$ for some $I \in \mathcal{I}_{\mathsf{FR}}$ such that $(l,I) \in S_{\mathsf{FR}}$. The untimed MDP $\mathsf{1st}[\mathsf{P}]$ will be used as an intermediate model to relate $\mathsf{FR}[\mathsf{P}]$ to $\mathsf{M}[\mathsf{P}]$. First we consider the relationship between $\mathsf{FR}[\mathsf{P}]$ and $\mathsf{1st}[\mathsf{P}]$.

Lemma 6.2. (1) *For each $((l,I),\rho) \in \rightarrow_{\mathsf{FR}}$, there exists $((l,\mathsf{1stInt}(I)),\nu) \in \rightarrow_{\mathsf{1st}}$ such that, for all $(l',I') \in S_{\mathsf{FR}}$, we have $\rho(l',I') = \nu(l',\mathsf{1stInt}(I'))$.*
(2) *For each $(l,I) \in S_{\mathsf{FR}}$, and for each $((l,\mathsf{1stInt}(I)),\nu) \in \rightarrow_{\mathsf{1st}}$, there exists $((l,I),\rho) \in \rightarrow_{\mathsf{FR}}$ such that, for all $(l',I') \in S_{\mathsf{FR}}$, we have $\nu(l',\mathsf{1stInt}(I')) = \rho(l',I')$.*

Proof. We prove part (1), noting that part (2) can be shown in a similar manner. Let $((l,I),\rho) \in \rightarrow_{\mathsf{FR}}$. Then there exists a probabilistic edge $(l,g,p) \in prob$ satisfying the conditions of Definition 6.1. We identify the transition $((l,\mathsf{1stInt}(I)),\nu) \in \rightarrow_{\mathsf{1st}}$ in the following way. Noting that $I \cap [\![g]\!] \neq \emptyset$ (condition (1) of Definition 6.1), we let $B = \mathsf{1stInt}(I \cap [\![g]\!])$. Therefore $B \geq \mathsf{1stInt}(I)$. Furthermore, we have that $B' \subseteq [\![inv(l)]\!]$ for all $\mathsf{1stInt}(I) \leq B' \leq B$, satisfying condition (1) of the definition of $\mathsf{M}[\mathsf{P}]$ (see Proposition 4.1). Furthermore, condition (2) for $\rightarrow_{\mathsf{FR}}$ of Definition 6.1 implies condition (2) of the definition of $\mathsf{M}[\mathsf{P}]$.

It remains to show that, for all $(l', I') \in S_{\mathsf{FR}}$, we have $\rho(l', I') = \nu(l', \mathsf{1stInt}(I'))$. By definition, it suffices to show that for all $(l', I') \in S_{\mathsf{FR}}$, we have $\rho_0(l', I') = \nu_0(l', \mathsf{1stInt}(I'))$ and $\rho_I(l', I') = \nu_{\mathsf{1stInt}(I)}(l', \mathsf{1stInt}(I'))$.

If $(l', I') = \mathsf{post}((l, I), (l, g, p), \{x\}, l')$, then $\mathsf{1stInt}(I') = [0; 0]$, and by definition we have $\rho_0(l', I') = p(\{x\}, l') = \nu_0(l', \mathsf{1stInt}(I'))$. If $(l', I') \neq \mathsf{post}((l, I), (l, g, p), \{x\}, l')$, then $\mathsf{1stInt}(I') \neq [0; 0]$, and $\rho_0(l', I') = 0 = \nu_0(l', \mathsf{1stInt}(I'))$.

If $(l', I') = \mathsf{post}((l, I), (l, g, p), \emptyset, l')$, then, by definition of post, we have $I' = ((\llbracket g \rrbracket \cap I)[\emptyset := 0])_{l'}^{\uparrow} = (\llbracket g \rrbracket \cap I)_{l'}^{\uparrow}$. We then conclude that $\mathsf{1stInt}(I') = \mathsf{1stInt}(I \cap \llbracket g \rrbracket)$. Hence, by definition of $\mathsf{M}[\mathsf{P}]$, we have that $\nu_{\mathsf{1stInt}(I)}(l', \mathsf{1stInt}(I')) = p(\emptyset, l')$. By Definition 6.1, we have $\rho_I(l', I') = p(\emptyset, l')$, and therefore $\rho_I(l', I') = \nu_{\mathsf{1stInt}(I)}(l', \mathsf{1stInt}(I'))$. If $(l', I') \neq \mathsf{post}((l, I), (l, g, p), \emptyset, l')$, then we obtain $\rho_I(l', I') = 0 = \nu_{\mathsf{1stInt}(I)}(l', \mathsf{1stInt}(I'))$.

We conclude that $\rho(l', I') = \nu(l', \mathsf{1stInt}(I'))$ for all $(l', I') \in S_{\mathsf{FR}}$. $\qquad\square$

We say that two untimed MDPs $\mathsf{M}_1^u = (S_1, \bar{s}_1, \rightarrow_1, lab_1)$ and $\mathsf{M}_2^u = (S_2, \bar{s}_2, \rightarrow_2, lab_2)$ are *isomorphic* if there exists a bijection $f : S_1 \rightarrow S_2$ such that:

(1) for each state $s \in S_1$, we have $lab_1(s) = lab_2(f(s))$;
(2) $f(\bar{s}_1) = \bar{s}_2$;
(3) $(s, \nu) \in \rightarrow_1$ if and only if $(f(s), f(\nu)) \in \rightarrow_2$, where $f(\nu) \in \mathsf{Dist}(S_2)$ is the distribution defined by $f(\nu)(s') = \nu(f^{-1}(s'))$ for each $s' \in S_2$.

Lemma 6.3. *The untimed MDPs* $\mathsf{FR}[\mathsf{P}]$ *and* $\mathsf{1st}[\mathsf{P}]$ *are isomorphic.*

Proof. We consider the bijection $f : S_{\mathsf{FR}} \rightarrow S_{\mathsf{1st}}$ such that $f(l, I) = (l, \mathsf{1stInt}(I))$ for each $(l, I) \in S_{\mathsf{FR}}$. First we have that $lab_{\mathsf{FR}}(l, I) = \mathcal{L}(l) = lab_{\mathsf{M}}(l, \mathsf{1stInt}(I))$. Second we have that $f(\bar{s}_{\mathsf{FR}}) = f((\bar{l}, [0; 0]_{\bar{l}}^{\uparrow})) = (\bar{l}, \mathsf{1stInt}([0; 0]_{\bar{l}}^{\uparrow})) = (\bar{l}, [0; 0]) = \bar{s}_{\mathsf{M}}$. Third, Lemma 6.2 establishes that $((l, I), \rho) \in \rightarrow_{\mathsf{FR}}$ if and only if $((\bar{l}, \mathsf{1stInt}(I), f(\rho)) \in \rightarrow_{\mathsf{1st}}$. $\qquad\square$

Given that isomorphism is as least as strict as probabilistic bisimilarity [SL95], and that, for any adversary A of an MDP, we can define a corresponding adversary A' of a probabilistically bisimilar MDP such that A and A' have the same reachability probabilities, we obtain the following corollary.

Corollary 6.4. *Let* $a \in AP$. *For any adversary* $A \in Adv_{\mathsf{FR}[\mathsf{P}]}$, *there exists an adversary* $A' \in Adv_{\mathsf{1st}[\mathsf{P}]}$ *such that:*

$$Prob_{\bar{s}_{\mathsf{FR}}}^A \{\omega \in Path_{ful}^A(\bar{s}_{\mathsf{FR}}) \mid \omega \models_{\mathsf{FR}[\mathsf{P}]} \mathsf{F}a\} = Prob_{\bar{s}_{\mathsf{1st}}}^{A'} \{\omega \in Path_{ful}^{A'}(\bar{s}_{\mathsf{1st}}) \mid \omega \models_{\mathsf{1st}[\mathsf{P}]} \mathsf{F}a\}. \quad (6.1)$$

Conversely, for any adversary $A' \in Adv_{\mathsf{1st}[\mathsf{P}]}$, *there exists an adversary* $A \in Adv_{\mathsf{FR}[\mathsf{P}]}$ *such that Equation 6.1 holds.*

It remains to relate $\mathsf{1st}[\mathsf{P}]$ to $\mathsf{M}[\mathsf{P}]$. The intuition underlying the following results is the following: while $\mathsf{1st}[\mathsf{P}]$ is a restriction of $\mathsf{M}[\mathsf{P}]$, the additional transitions of $\mathsf{M}[\mathsf{P}]$ only result in states from which the ability to enable probabilistic edges is weakened. For any two states $(l, B), (l, B')$ of $\mathsf{M}[\mathsf{P}]$, we write $(l, B) \preceq (l', B')$ if $l = l'$ and $B \leq B'$. Furthermore, for the distribution $\nu \in \mathsf{Dist}(S_{\mathsf{1st}})$ and $\nu' \in \mathsf{Dist}(S_{\mathsf{M}})$, we write $\nu \preceq \nu'$ if there exists a bijection $f : \mathsf{support}(\nu) \rightarrow \mathsf{support}(\nu')$ such that $f(\nu) = \nu'$, and, for each $(l, B) \in \mathsf{support}(\nu)$, we have $(l, B) \preceq f(l, B)$. The following lemma can be derived directly from the definitions of $\mathsf{1st}[\mathsf{P}]$ and $\mathsf{M}[\mathsf{P}]$.

Lemma 6.5. *Let* $(l, B) \in S_{\mathsf{1st}}$ *and* $(l, B') \in S_{\mathsf{M}}$ *be such that* $(l, B) \preceq (l', B')$. *Then, for each* $((l, B'), \nu') \in \rightarrow_{\mathsf{M}}$, *there exists* $((l, B), \nu) \in \rightarrow_{\mathsf{1st}}$ *such that* $\nu \preceq \nu'$.

Lemma 6.5 then allows us to construct, for any adversary A of $\mathsf{M[P]}$, an adversary A' of $\mathsf{1st[P]}$ such that the probability of reaching a given set of locations from the initial state is the same for A and A' (this fact also follows by noting that $(\preceq)^{-1}$ is a probabilistic simulation [SL95]). The converse result, which states that, for any adversary A of $\mathsf{1st[P]}$, an adversary A' of $\mathsf{M[P]}$ such that the probability of reaching a given set of locations from the initial state is the same for A and A', follows from the fact that $\mathsf{1st[P]}$ is a restriction of $\mathsf{M[P]}$. We then obtain the following corollary.

Corollary 6.6. *Let $a \in AP$. For any adversary $A \in Adv_{\mathsf{1st[P]}}$, there exists an adversary $A' \in Adv_{\mathsf{M[P]}}$ such that:*

$$Prob_{\bar{s}_{\mathsf{1st}}}^{A}\{\omega \in Path_{ful}^{A}(\bar{s}_{\mathsf{1st}}) \mid \omega \models_{\mathsf{1st[P]}} \mathsf{F}a\} = Prob_{\bar{s}_{\mathsf{M}}}^{A'}\{\omega \in Path_{ful}^{A'}(\bar{s}_{\mathsf{M}}) \mid \omega \models_{\mathsf{M[P]}} \mathsf{F}a\} . \ (6.2)$$

Conversely, for any adversary $A' \in Adv_{\mathsf{M[P]}}$, there exists an adversary $A \in Adv_{\mathsf{1st[P]}}$ such that Equation 6.2 holds.

Combining Corollary 6.4 and Corollary 6.6, and using the proof of Proposition 4.1, which states that the results of model checking a PCTL formula (including reachability properties of the form $\mathbb{P}_{\sim\lambda}(\mathsf{F}a)$) on $\mathsf{M[P]}$ correspond to the satisfaction of the formula on $\mathsf{T[P]}$, we conclude with the following corollary.

Corollary 6.7. *Let $a \in AP$, $\sim \in \{<, \leq, \geq, >\}$ and $\lambda \in [0,1]$. We have $\mathsf{FR[P]} \models \mathbb{P}_{\sim\lambda}(\mathsf{F}a)$ if and only if $\mathsf{T[P]} \models \mathbb{P}_{\sim\lambda}(\mathsf{F}a)$.*

7. Conclusion

We have shown that probabilistic model-checking problems for 1C-PTA can be performed efficiently if qualitative properties with non-punctual timing bounds are considered. If the temporal logic features punctual timing bounds, the problem becomes EXPTIME-complete. We have also shown that the forward reachability algorithm of Kwiatkowska et al. [KNSS02] can be used to compute the exact probability of reaching a state set for 1C-PTA. For future work, we intend to consider the complexity of model checking 1C-PTA against *quantitative* properties without punctual timing bounds (that is, properties of PTCTL[$\leq, \geq$]). On the other hand, we have shown that model-checking problems for 2C-PTA are EXPTIME-complete, regardless of the probability threshold and timing bounds used.

References

[AB06]   R. Alur and M. Bernadsky. Bounded model checking for GSMP models of stochastic real-time systems. In *Proc. of the 9th International Workshop on Hybrid Systems Computation and Control (HSCC'06)*, volume 3927 of *LNCS*, pages 19–33. Springer, 2006.

[ACD91]   R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking for probabilistic real-time systems. In *Proc. of the 18th International Conference on Automata, Languages and Programming (ICALP'91)*, volume 510 of *LNCS*, pages 115–136. Springer, 1991.

[ACD93]   R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.

[AD94]   R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[ADOW05] P. A. Abdulla, J. Deneux, J. Ouaknine, and J. Worrell. Decidability and complexity results for timed automata via channel machines. In *Proc. of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *LNCS*, pages 1089–1101. Springer, 2005.

[ATP04]   R. Alur, S. La Torre, and G. Pappas. Optimal paths in weighted timed automata. *Theoretical Computer Science*, 318(3):297–322, 2004.

[BCH$^+$07] C. Baier, L. Cloth, B. Haverkort, M. Kuntz, and M. Siegle. Model checking action- and state-labelled Markov chains. *IEEE Transactions on Software Engineering*, 33(4):209–224, 2007.

[BdA95]   A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. of the 15th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'95)*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[BDL$^+$06] G. Behrmann, A. David, K. G. Larsen, J. Håkansson, P. Pettersson, W. Yi, and M. Hendriks. UPPAAL 4.0. In *Proc. of the 3rd International Conference on Quantitative Evaluation of Systems (QEST'06)*, pages 125–126. IEEE Computer Society Press, 2006.

[BHHK03] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.

[BK98]   C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.

[CGP99]   E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT Press, 1999.

[CJH03]   K. Chatterjee, M. Jurdziński, and T. Henzinger. Simple stochastic parity games. In *Proc. of the 12th International Conference on Computer Science Logic (CSL'03)*, volume 2803 of *LNCS*, pages 100–113. Springer, 2003.

[CKS81]   A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.

[CY95]   C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[dA97a]   L. de Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.

[dA97b]   L. de Alfaro. Temporal logics for the specification of performance and reliability. In *Proc. of the 14th Annual Symposium on Theoretical Aspects of Computer Science (STACS'97)*, volume 1200 of *LNCS*, pages 165–176. Springer, 1997.

[DHS07]   S. Donatelli, S. Haddad, and J. Sproston. CSLTA: an expressive logic for continuous-time Markov chains. In *Proceedings of the 4th International Conference on Quantitative Evaluation of Systems (QEST'07)*, pages 31–40. IEEE Computer Society Press, 2007.

[DKN04]   C. Daws, M. Kwiatkowska, and G. Norman. Automatic verification of the IEEE 1394 root contention protocol with KRONOS and PRISM. *Software Tools for Technology Transfer*, 5(2–3):221–236, 2004.

[HJ94]   H. A. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[Jen96]   H. E. Jensen. Model checking probabilistic real time systems. In *Proc. of the 7th Nordic Workshop on Programming Theory*, pages 247–261. Chalmers Institute of Technology, 1996.

[KNPS06]   M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design*, 29:33–78, 2006.

[KNSS02]   M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 286:101–150, 2002.

[LMO06]   F. Laroussinie, N. Markey, and G. Oreiby. Model checking timed ATL for durational concurrent game structures. In *Proc. of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *LNCS*, pages 245–259. Springer, 2006.

[LMS04]   F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking timed automata with one or two clocks. In *Proc. of the 15th International Conference on Concurrency Theory (CONCUR'04)*, volume 3170 of *LNCS*, pages 387–401. Springer, 2004.

[LMS05]   F. Laroussinie, N. Markey, and Ph. Schnoebelen. Efficient timed model checking for discrete-time systems. *Theoretical Computer Science*, 353(1–3):249–271, 2005.

[LS05]   F. Laroussinie and J. Sproston. Model checking durational probabilistic systems. In *Proc. of the 8th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'05)*, volume 3441 of *LNCS*, pages 140–154. Springer, 2005.

[LS07]   F. Laroussinie and J. Sproston. State explosion in almost-sure probabilistic reachability. *Information Processing Letters*, 102(6):236–241, June 2007.

[LW05]   S. Lasota and I. Walukiewicz. Alternating timed automata. In *Proc. of the 8th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'05)*, volume 3441 of *LNCS*, pages 299–314. Springer, 2005.

[PT87]   C. Papadimitriou and J. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[SL95]   R. Segala and N. A. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.

[Spr01]   J. Sproston. *Model checking for probabilistic timed and hybrid systems*. PhD thesis, University of Birmingham, School of Computer Science, 2001.

[Sto02]   M. Stoelinga. *Alea jacta est: verification of probabilistic, real-time and parametric systems*. PhD thesis, University of Nijmegen, the Netherlands, April 2002.

[TYB05]   S. Tripakis, S. Yovine, and A. Bouajjani. Checking timed Büchi automata emptiness efficiently. *Formal Methods in System Design*, 26(3):267–292, 2005.

[Var85]   M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of the 16th Annual Symposium on Foundations of Computer Science (FOCS'85)*, pages 327–338. IEEE Computer Society Press, 1985.

Appendix A. Model checking $\text{PTCTL}^{0/1}[\leq, \geq]$ over PTAs with strict constraints

Here we describe briefly the general case for the model-checking algorithm of Theorem 4.3, that is when the guards and invariants of P may be strict and when the intervals in $\mathsf{Sat}[l, \Phi_i]$ may be open (or half-open). This makes the algorithm more difficult to describe even if the complexity remains polynomial. Here we will only give the main idea about how to deal with these kind of constraints.

First note that an optimal strategy of either of the players P_n or P_p cannot always be restricted to perform transitions at integer points: if a transition has to be performed as soon as possible and if it has a guard $x > d$, then it is not possible to perform it from the position d, and in some cases it is not optimal to wait until $d + 1$. In fact, sometimes there is even no optimal strategy corresponding to the optimal values (for α, β, γ and δ). The same remark holds for the notion of optimal (timed) path in timed automata [ATP04]. We have to define the optimal value as a constant k such that there exist strategies with a cost arbitrarily close (above or below) to k. Thus the optimal value will be denoted as "$\epsilon\ k$" with $\epsilon \in \{<, =, >\}$. For example, "< 2" will mean that the optimal value is less than 2 but arbitrarily close to 2.

The method proposed for the simple case has to be modified in order to handle the (non)strict value. For each $\text{PTCTL}^{0/1}[\leq, \geq]$ modality, we can use a variant of the finite discrete TMDP T^r defined in the proof of Theorem 4.3: again we consider the singular states (l, b_i) and the "symbolic states" $(l, (b_i; b_{i+1}))$ with $b_i \in \mathbb{B}$, with the two special positions b_i^+ and b_{i+1}^-.

Consider the case of subformulae of the form $\mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$. Then we want to compute the function α for any configuration (l, v) of $\mathsf{T}[\mathsf{P}]$. Figure 3 shows two simple examples where the value for α is indicated for every integer point and for the left and right side of the intervals. Note that in these examples, we just assume that *prob* contains the two probabilistic edges $(l, x > 1, p)$ (respectively, $(l, x = 2, p)$) where $p(\{x\}, l')$, and $(l', x = 1, p')$

where $p'(\{x\}, l'')$. Moreover the only state satisfying Φ_2 is $(l'', 0)$, and all states satisfy Φ_1. The value α corresponds to the duration between the current state and $(l'', 0)$. This example is sufficient to illustrate the problem of strict and non-strict values.

Let us consider the structure of the function α. For the singular points (l, b_i) the value can be of the form "$< k$", "$= k$", "$> k$", or ∞ when there exists a strategy for P_n to avoid Φ_2 forever. Note that the case "$> k$" can occur for a state (l, b_j) when the property Φ_2 holds for an interval $(l, (b_i; b_{i+1}))$: reaching this interval from (l, b_j) can be done by a duration strictly greater than $b_i - b_j$. The other cases are illustrated on Figure 3.
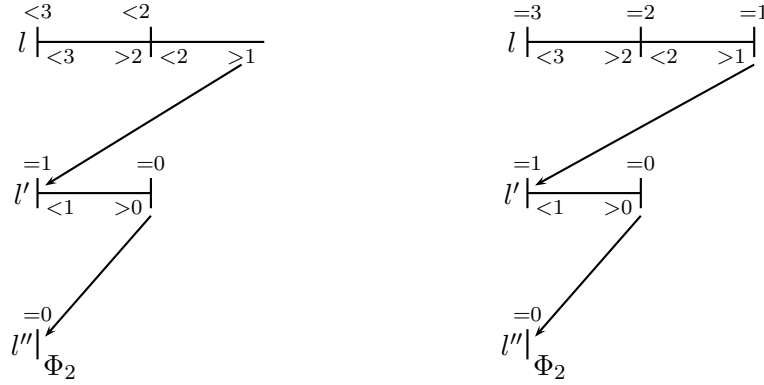

Figure 3: Example of optimal value for α

Now consider the case of symbolic states $(l, (b_i; b_{i+1}))$. The structure of α over such an interval is always decreasing: indeed either the best strategy for P_n consists in performing a distribution from the current interval, in which case it is always better to delay until the last point (b_{i+1}^-) of the interval, or the best strategy consists in delaying until a future state or interval. We can see that the value of the rightmost position inside the interval will be always of the form "$> k$": indeed it depends either on the value in b_{i+1} (if the strategy goes through this point) or on the value in some (l', b_0) if there is transition with a reset of clock x. Assume that this value is "$\epsilon\ k$" and consider a point (l, v) with $v \in (b_i; b_{i+1})$. Then any duration in $(0; b_{i+1} - v)$ is sufficient to reach Φ_2 in more than k time units in case of an optimal strategy: note that this fact does not depend on ϵ. Given a value "$> k$" for the rightmost position of $(b_i; b_{i+1})$, we can deduce the function α for any position v in the interval: it is $b_{i+1} - v + k$.

Therefore (1) the optimal strategies use only the singular points and the rightmost positions b_{i+1}^- in the intervals, and (2) the function α over an interval can be derived from the value in the rightmost position. Thus we will restrict the computation of coefficients α to these points.

Thus the algorithm consists in computing the function α by using values of the form "$< k$", "$= k$" or "$> k$". This is slightly more technical than the basic case.

Finally similar techniques can be used also for the other functions (β, γ and δ).