VNU Journal of Science: Comp. Science & Com. Eng., Vol. 32, No. 1 (2016) 58–73

Towards Model-checking Probabilistic Timed Automata against Probabilistic Duration Properties☆

Van Hung Dang[1,*], Miaomiao Zhang[2], Dinh Chinh Pham[1]

[1] *VNU University of Engineering and Technology, Hanoi, Vietnam*
[2] *School of Software Engineering, Tongji University, Shanghai, China*

Abstract

In this paper, we consider a subclass of Probabilistic Duration Calculus formula called Simple Probabilistic Duration Calculus (SPDC) as a language for specifying dependability requirements for real-time systems, and address the two problems: to decide if a probabilistic timed automaton satisfies a SPDC formula, and to decide if there exists a strategy of a probabilistic timed automaton satisfies a SPDC formula. We prove that the both problems are decidable for a class of SPDC called probabilistic linear duration invariants, and provide model checking algorithms for solving these problems.

Received 25 November 2015, revised 20 December 2015, accepted 31 December 2015

Keywords: Probabilistic Duration Calculus, Probabilistic Timed Automata, Model-checking, Markov Decision Process.

1. Introduction

In 1992, Chaochen Zhou, Hoare C.A.R and Anders Ravn introduced Duration Calculus [1] as a logic for reasoning about real-time systems. The calculus has attracted a great deal of attention, and was then developed further in many other works because of its rich meanings. Many of those works have been summarized in the monograph [2]. For specifying the dependability of real-time systems, a kind of probabilistic extension of Duration Calculus has been introduced in [3, 4]. No rigorous syntax has been introduced in these papers, and the authors just focused on the development of techniques for reasoning instead of the ones for checking. A version with a proof system of Probabilistic Duration Calculus with infinite interval was then developed by Dimitar Guelev [5], and in [6] we have shown that the calculus is useful for reasoning about QoS contracts in component-based real-time systems.

For Duration Calculus, some techniques for checking if a timed automaton satisfies a duration calculus formula written in the form of linear duration invariants have been developed [7, 8, 9, 10, 11, 8]. However, to our knowledge, not many works have been done for checking if a probabilistic real-time system satisfies a PDC formula. This is, perhaps, because in the model of probabilistic systems, there is too much randomization and nondeterminism, and this makes model checking too complicated.

Kwiatkowska et al in [12, 13] proposed a variant of probabilistic timed automata that allows probabilistic choice only at discrete transitions. To resolve the nondeterminism between the passage of time and discrete transitions they used the concept of strategy which is essentially a deterministic schedule

☆ This research was funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.03-2014.23.

* Corresponding author. Email: dvh@vnu.edu.vn

policy. Then, the set of executions of a probabilistic timed automaton according to a strategy forms a Markov chain, and hence the satisfaction of a probabilistic timed CTL formula by this set can be defined, and then based on the region graph of the timed automaton the satisfaction of a probabilistic timed CTL formula by the timed automaton can be also verified. The idea of fixing a strategy when studying the probabilistic behavior of a probabilistic timed automaton restricts the scope of the verification problem significantly, making the checking problem more tractable. Then, verifying the set of all strategies against a given probabilistic property can be done by searching for the "worst case" strategy according to the probabilistic property and then apply the verification technique to it. This idea is a motivation for us to reconsider the problem of checking a probabilistic timed automaton for a PDC formula that we gave up before.

In this paper, we introduced a simple probabilistic extension of DC called Probabilistic Duration Calculus for specifying dependability requirements of real-time systems. The extension is conservative in the sense that a formula of DC is also a formula of PDC with semantics adapted to probabilistic domain. PDC also consists of formulas representing the constraints for the probability of the satisfaction of a DC formula by a strategy for an interval. We use the behavioral model proposed by Kwiatkowska et al to define the semantics of our logic. Since probabilistic timed CTL and PDC are not comparable, and since for many probabilistic properties PDC is more convenient to specify, a model checking technique for checking probabilistic timed automata against PDC properties is useful. To solve this problem, we first develop a technique to decide if a strategy in a probabilistic timed automaton satisfies a PDC formula of a certain form. This technique is essentially an extension of our technique developed earlier in [10, 9] to check if a timed automaton satisfies a DC formula in the form of linear duration invariants or discretisable DC formulas based on searching in the integral reachability graph of the timed automaton. Then, we generalize this technique to achieve our goal with a model-checking algorithm.

The first version of this paper was published in [14]. In this extended version, in addition to the problem of verification, we formulate also the problem of strategy synthesis, i.e. to decide if there is a strategy for a probabilistic timed automaton that satisfies a probabilistic linear duration invariant and show that this problem is also solvable. We provide all proof details and algorithms for doing model-check.

Our paper is organized as follows. In the next section we present the Probabilistic Timed Automata model. Section 3 presents syntax and semantics of our PDC. Our main results is presented in Section 4 where we formulate our model checking problem and give our solution to it. The last section is the conclusion of the paper.

2. Probabilistic Timed Automata

In this section, we recall the concepts of probabilistic timed automata model and probabilistic timed structure as its semantics from [15, 12]. We use a simple model of gas burners to illustrate the concepts as its requirement specification is a typical example for time duration properties.

Probability distributions and Markov decision processes. A discrete probability distribution over a set S is a mapping $p : S \to [0, 1]$ such that the set $\{s \mid s \in S \text{ and } p(s) > 0\}$ is finite, and $\sum_{s \in S} p(s) = 1$. The set of all discrete probability distributions over S is denoted by $\mu(S)$.

A Markov decision process is a tuple $(Q, Steps)$, where Q is a set of states, and $Steps : Q \to 2^{\mu(Q)}$ is a function assigning a set of probability distributions to each state. The intuition is that the Markov decision process traverses the state space by making transitions determined by $Steps$: in a state s, the process selects nondeterministically a probability distribution p in $Steps(s)$, and then makes a probabilistic choice according to p as to which state to move to. As in [12] we label the action selecting a probability distribution with a letter

from Σ, and assume that $Steps : Q \to 2^{\Sigma \times \mu(Q)}$ and Σ is a set of actions. The intuition now becomes that the Markov decision process traverses the state space by making transitions determined by *Steps*: in a state s, the process performs an action $a \in \Sigma$ selecting nondeterministically a probability distribution p in $Steps(s)$, and then makes a probabilistic choice according to p as to which state to move to. So, a transition is of the form $s \xrightarrow{a,p} s'$, where $(a, p) \in \Sigma \times \mu(Q)$ is the label of the transition. We also assume a labeling function $L : Q \to 2^{AP}$, where AP is a set of atomic propositions, that associates a state s with the set of atomic propositions that hold at state s. Then, a labeled Markov decision process is a tuple $(Q, Steps, L)$.

Labeled paths (or execution sequences) are nonempty finite or infinite sequence of consecutive transitions of the form

$$\omega = s_0 \xrightarrow{l_0} s_1 \xrightarrow{l_1} s_2 \xrightarrow{l_2} \dots,$$

where s_i are states and l_i are labels for transitions. For a path ω, let $first(\omega)$ denote the first state of ω, and if ω is finite then let $last(\omega)$ denote the last state of ω. $|\omega|$ is the length of ω and is defined as the number of transition occurrences in ω which is ∞ if ω is infinite. For $k \leq |\omega|$, let $\omega(k)$ denote the kth state of ω, and $step(\omega, k)$ denote the label of the kth transition in ω. For two paths $\omega = s_0 \xrightarrow{l_0} s_1 \xrightarrow{l_1} s_2 \xrightarrow{l_2} \dots s_n$ and $\omega' = s'_0 \xrightarrow{l'_0} s'_1 \xrightarrow{l'_1} s'_2 \xrightarrow{l'_2} \dots$ such that $s_n = s'_0$, the concatenation of ω and ω' is defined as $\omega\omega' = s_0 \xrightarrow{l_0} s_1 \xrightarrow{l_1} s_2 \xrightarrow{l_2} \dots s_n \xrightarrow{l'_0} s'_1 \xrightarrow{l'_1} s'_2 \xrightarrow{l'_2} \dots$.

Clocks, clock valuations, clock constraints. Let $\mathbb{R}^{\geq 0}$ denote the set of non negative real numbers. A clock is a real-valued variable which increases at the same rate as real time. Let $C = \{x_1 \dots, x_n\}$ be a set of clocks. A clock valuation is a function $\nu : C \to \mathbb{R}^{\geq 0}$ that assigns a real value to each clock. Let $(\mathbb{R}^{\geq 0})^C$ denote the set of all clock valuations, and $\mathbf{0}$ denote the clock valuation that assigns 0 to each clock in C. For a set of clocks $X \subseteq C$ we denote by $\nu[X := 0]$ the clock valuation that assigns 0 to all clocks in X and agrees with ν on all other clocks. For $t \in \mathbb{R}^{\geq 0}$, we write $\nu + t$ for the clock valuation that assigns $\nu(x) + t$ to each clock $x \in C$. A constraint over C is an expression of the form $x_i \sim c$ or $x_i - x_j \sim c$, where $i \neq j$, $i, j \leq n$ and $\sim \in \{<, \leq, >, \geq\}$ and $c \in \mathbb{N}$. A clock valuation ν satisfies a clock constraint $x_i \sim c$ ($x_i - x_j \sim c$) iff $\nu(x_i) \sim c$ ($\nu(x_i) - \nu(x_j) \sim c$). A zone of C is a convex subset of the valuation space $(\mathbb{R}^{\geq 0})^C$ described by a conjunction of constraints. For a zone ζ and a set of clocks $X \subseteq C$ the set $\{\nu[X := 0] \mid \nu \in \zeta\}$ is also a zone, and is denoted by $\zeta[X := 0]$. Let $\mathbf{Z}_C$ denote the set of all zones of C.

Probabilistic timed automata and probabilistic timed structures. Timed automata were introduced in [16] as a model of real-time systems. They are extended with discrete probability distribution to model probabilistic real-time systems. In the sequel, let AP be a given set of atomic propositions.

Definition 1. *A probabilistic timed automaton (PTA) is a tuple* $G = (\mathcal{S}, \mathcal{L}, \bar{s}, C, inv, prob, \langle \tau_s \rangle_{s \in \mathcal{S}})$ *consisting of*

- *a finite set $\mathcal{S}$ of nodes, a start node $\bar{s} \in \mathcal{S}$, a finite set C of clocks,*
- *a function $\mathcal{L} : \mathcal{S} \to 2^{AP}$ assigning to each node of the automaton a set of atomic propositions that are supposed to be those that are true in that node, a function $inv : \mathcal{S} \to \mathbf{Z}_C$ assigning to each node an invariant condition,*
- *a function $prob : \mathcal{S} \to 2^{\mu(\mathcal{S} \times 2^C)}$ assigning to each node a set of discrete probability distributions on $\mathcal{S} \times 2^C$,*
- *a family of functions $\langle \tau_s \rangle_{s \in \mathcal{S}}$ where, for any $s \in \mathcal{S}$, $\tau_s : prob(s) \to \mathbf{Z}_C$ assigns to each $p \in prob(s)$ an enabling condition.*

The last item in the definition says that all the probabilistic choices according to a probabilistic distribution (selected at a node) have the same enabling condition. The probabilistic timed automaton behaves nearly in the same way as a

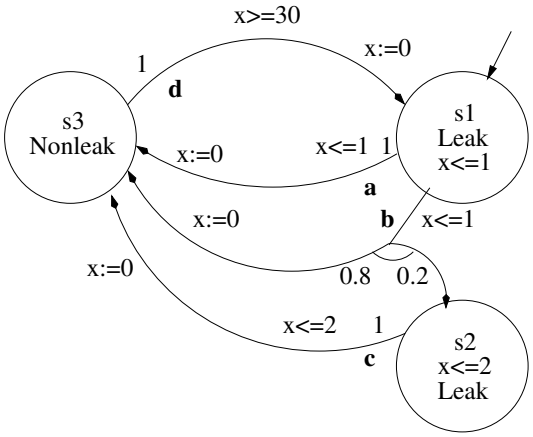

Fig. 1: A probabilistic timed automaton for a simple gas burner.

timed automaton does, except that it has to select a probability distribution at each discrete step.

We denote by $\mathbf{Z}_C(G)$ the set of all clock zones occurring in G,

$$\begin{aligned}\mathbf{Z}_C(G) = \quad & \{inv(s) \in \mathbf{Z}_C \mid s \in \mathcal{S}\} \cup \\ & \{\tau_s(p) \in \mathbf{Z}_C \mid s \in \mathcal{S} \text{ and} \\ & \quad p \in prob(s)\}.\end{aligned}$$

Example 1. Fig. 1 shows a probabilistic timed automaton for a simple gas burner.

The system starts at the node $s1$, with the gas valve is opened without flame being on, hence gas is leaking. At this state, there are two nondeterministic choices. The first choice denoted by transition **a** is that with the probability 1, the flame is turned on within one second ($x \leq 1$) and the system moves to node $s3$ for which gas is not leaking. The second choice denoted by transition **b** is as follows: with the probability 0.8, the flame is turned on within one second and the system moves to node $s3$ for which gas is not leaking, and with probability 0.2 the flame fails to be on within one second, and the system moves to node $s2$ for which gas is still leaking. In state $s2$, with probability 1, the gas valve is closed successfully within 2 seconds since the time the system entered $s1$ last time, and the system moves to node $s3$. At state $s3$, the gas burner will move to the state s! only after it has stayed there at least 30 seconds. Formally, in this example, the function $prob$ is given as: $prob(s1) = \{p0, p1\}$, $prob(s2) = \{p2\}$, $prob(s3) = \{p3\}$, where $p0(s3, \{x\}) = 1$, $p1(s3, \{x\}) = 0.8$, $p1(s2, \emptyset) = 0.2$. $p2(s3, \{x\}) = 1$, $p3(s1, \{x\}) = 1$, and $\tau_{s1}(p0) = \tau_{s1}(p1) = \{x \leq 1\}$, $\tau_{s2}(p2) = \{x \leq 2\}$ and $\tau_{s3}(p3) = \{x \geq 30\}$. The function inv is defined as $inv(s1) = \{x \leq 1\}$, $inv(s2) = \{x \leq 2\}$ and $inv(s3) = true$. The labels of states are given by function $\mathcal{L}$ defined as $\mathcal{L}(s1) = \mathcal{L}(s2) = leak$, and $\mathcal{L}(s3) = nonleak$.

As in [12] we use probabilistic timed structures as underlying semantics model for PTA.

Definition 2. *A probabilistic timed structure $\mathcal{M}$ is a labeled Markov decision process $(Q, Steps, L)$ where Q is a set of states, $Steps : Q \to 2^{\mathbb{R}^{\geq 0} \times \mu(Q)}$ is a function which assigns to each state $q \in Q$ a set $Steps(q)$ of pairs of the form (t, p), where $t \in \mathbb{R}^{\geq 0}$ and $p \in \mu(Q)$, and $L : Q \to 2^{AP}$ is a state labeling function.*

Function *Steps* specifies the set of transitions that $\mathcal{M}$ can choose nondeterministically at each state. Therefore, if at state $q \in Q$, $\mathcal{M}$ chooses $(t, p) \in Steps(q)$, then after t time units have elapsed, a probabilistic transition is made to state q' with probability $p(q')$. A path of $\mathcal{M}$ is a nonempty finite or infinite sequence:

$$\omega \quad = \quad q_0 \xrightarrow{t_0, p_0} q_1 \xrightarrow{t_1, p_1} q_2 \xrightarrow{t_2, p_2} \dots$$

where $q_i \in Q$, $(t_i, p_i) \in Steps(s_i)$, and $p_i(q_{i+1}) > 0$ for all $0 \leq i \leq |\omega|$. For a given probabilistic timed structures $\mathcal{M}$ we denote by $Path_{fin}$ ($Path_{inf}$) the set of finite (infinite) paths, and by $Path_{fin}(q)$ ($Path_{inf}(q)$) the set of paths in $Path_{fin}$ ($Path_{inf}$) that start from state q. Let ω be infinite. A position of ω is a pair (i, t), where $i \in \mathbb{N}$ and $t \in \mathbb{R}^{\geq 0}$ such that $0 \leq t \leq t_i$. The state at position (i, t) is denoted by $state_\omega(i, t)$. Given two positions (i, t) and (j, t') of ω, we say (j, t') precedes (i, t) (in ω, written by $(j, t') \prec (i, t)$) if $j < i$ or $j = i$ and $t' < t$.

Definition 3. *For any path ω of a probabilistic timed structure $\mathcal{M}$ and $0 \leq i \leq |\omega|$ we define $\mathcal{D}_\omega(i)$, the elapsed time until the ith transition, as follows: $\mathcal{D}_\omega(0) = 0$ and for any $1 \leq i \leq |\omega|$:*

$$\mathcal{D}_\omega(i) \quad = \quad \textstyle\sum_{j=0}^{i-1} t_j.$$

An infinite path ω is said to be divergent iff for any $t \in \mathbb{R}^{\geq 0}$, there exists $j \in \mathbb{N}$ such that $\mathcal{D}_\omega(j) > t$. Let ω be infinite. For each state $q \in Q$, we define a $\{0,1\}$-valued function $q_\omega : \mathbb{R}^{\geq 0} \to \{0,1\}$ as

$$q_\omega(t) = \begin{cases} 1 & \text{iff there exists a position } (i,t') \text{ s.t.} \\ & t' > 0,\ state_\omega(i,t') = q \text{ and} \\ & t = \mathcal{D}_\omega(i) + t', \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, $q_\omega(t) = 1$ means that in the path ω, state q is present in an interval $(t-\delta, t]$ for some $\delta > 0$, and otherwise $q_\omega(t) = 0$.

The concept of strategy was introduced in the literature (see, e.g. Kwiakowska [12]) as a schedule for resolving all the nondeterministic choices of the model. Note that we have restricted ourselves to discrete probability distributions only.

Definition 4. *A strategy (or scheduler) of a probabilistic timed structure $\mathcal{M} = (Q, Steps, L)$ is a function A mapping every nonempty finite path ω of $\mathcal{M}$ to a pair (t,p) such that $A(\omega) \in Steps(last(\omega))$, and the empty path ϵ to a state in Q. Let $\mathcal{A}$ be the set of all strategies of $\mathcal{M}$.*

Let us denote a prefix of length i of ω by $\omega^{(i)}$, and define for a given strategy A

$$Path^A_{fin} = \left\{ \omega \in Path_{fin} \;\middle|\; \begin{array}{l} A(\epsilon) = \omega^{(0)}, \text{ and} \\ step(\omega, i) = A(\omega^{(i)}) \\ \text{for } 0 \leq i < |\omega| \end{array} \right\}$$

$$Path^A_{inf} = \left\{ \omega \in Path_{inf} \;\middle|\; \begin{array}{l} A(\epsilon) = \omega^{(0)}, \text{ and} \\ step(\omega, i) = A(\omega^{(i)}) \\ \text{for } 0 \leq i \end{array} \right\}$$

Recall that $step(\omega, i)$ returns the label of the ith transition in ω. From Definition 4, all ω in $Path^A_{fin}$ and $Path^A_{inf}$ start from the same state defined by $A(\epsilon)$, and intuitively they represent the behaviors of $\mathcal{M}$ according to the scheduler A,

A sequential Markov chain $MC^A = (Path^A_{fin}, \mathbf{P}^A)$ is associated with a strategy A in a natural way to express the executions of $\mathcal{M}$ according to A, where $\mathbf{P}^A$ is defined as

$$\mathbf{P}^A(\omega, \omega') = \begin{cases} p(q) & \text{if } A(\omega) = (t,p) \text{ and} \\ & \omega' = \omega \xrightarrow{t,p} q, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{F}^A_{Path}$ be the smallest σ-algebra on $Path^A_{inf}$ which for all $\omega' \in Path^A_{fin}$ contains the sets $\{\omega \mid \omega \in Path^A_{inf}$ and ω' is a prefix of $\omega\}$. Let $Prob^A_{fin} : Path^A_{fin} \to [0,1]$ be the mapping defined inductively on the length of paths in $Path^A_{fin}$ as follows. If $|\omega| = 0$ then $Prob^A_{fin}(\omega) = 1$. Let $\omega' \in Path^A_{fin}$ be a finite path of A. If $\omega' = \omega \xrightarrow{t,p} q$ for some $\omega \in Path^A_{fin}$, then we let $Prob^A_{fin}(\omega') = Prob^A_{fin}(\omega)\mathbf{P}^A(\omega, \omega')$. The measure $Prob^A$ on $\mathcal{F}^A_{Path}$ is the unique measure such that $Prob^A(\{\omega \mid \omega \in Path^A_{inf}$ and ω' is a prefix of $\omega\}) = Prob^A_{fin}(\omega')$. In this paper, we assume that the strategies under consideration are divergent in the probabilistic sense, i.e. we assume that for any strategy A, $Prob^A(\{\omega \mid \omega \in Path^A_{inf}$ and ω is divergent$\}) = 1$.

We now define the behavior of probabilistic timed automata by associating every probabilistic timed automaton with a probabilistic timed structure. A state of the structure consists of a state of the automaton, and a valuation for the clock variables.

Definition 5. *For any probabilistic timed automaton G as in Definition 1, define the probabilistic timed structure $\mathcal{M}_G = (Q_G, Steps_G, L_G)$ as follows.*

- $Q_G = \{\langle s, \nu \rangle \mid s \in \mathcal{S}, \nu \in (\mathbb{R}^{\geq 0})^C\}$
- *The function $Steps_G : Q_G \to 2^{\mathbb{R}^{\geq 0} \times \mu(Q_G)}$ assigns to each state in Q_G a set of transitions, each of which takes the form $(t, \bar{p})$ where $t \in \mathbb{R}^{\geq 0}$ and $\bar{p}$ is a discrete probabilistic distribution on Q, and is defined as:*
  - *$(t, \bar{p}) \in Steps_G(\langle s, \nu \rangle)$ if there exists $p \in prob(s)$ such that (a) the valuation $\nu + t$ satisfies $\tau_s(p)$ and $\nu + t'$ satisfies $inv(s)$ for all $0 \leq t' \leq t$, and (b) for any $\langle s', \nu' \rangle \in Q_G$: $\bar{p}(\langle s', \nu' \rangle) = \sum_{X \subseteq C \wedge (\nu+t)[X:=0]=\nu'} p(s', X)$. For convenience, we refer to $\bar{p}$ as having type p, denoted by $\mathtt{type}(\bar{p}) = p$.*
  - *Let $(t, \bar{p}) \in Steps_G(\langle s, \nu \rangle)$ if (a) the valuation $\nu + t'$ satisfies $inv(s)$ for all $0 \leq t' \leq t$, and (b) for any $\langle s', \nu' \rangle \in$*

Q_G: $\bar{p}(\langle s', \nu' \rangle) = 1$ *if* $\langle s', \nu' \rangle = \langle s, \nu + t \rangle$, *and* $\bar{p}(\langle s', \nu' \rangle = 0$ *otherwise. We refer to* $\bar{p}$ *as having type* $\top$, *i.e.* $type(\bar{p}) = \top$.

- *The labeling function* $L : Q \to 2^{AP}$ *is defined as:* $L_G(\langle s, \nu \rangle) = \mathcal{L}(s)$ *for all* $\langle s, \nu \rangle \in Q_G$.

The second item of the definition of the function *Steps* allows the automaton to stay in a state forever from a time if the invariant for the state is never violated from that time, and the corresponding path is infinite.

Any strategy for the timed structure $\mathcal{M}_G$ is also called strategy for probabilistic timed automaton G.

Example 2. The following path is a path of (a strategy of) the timed structure of the timed automaton in Fig. 1:

$$\begin{aligned}\omega \;=\; & \langle s1, 0\rangle \xrightarrow{.9,.8} \langle s3, 0\rangle \xrightarrow{31,1} \langle s1, 0\rangle \xrightarrow{.7,.2} \\ & \langle s2, .7\rangle \xrightarrow{1.2,1} \langle s_3, 0\rangle \xrightarrow{30,1} \langle s3, 30\rangle \xrightarrow{100,1} \\ & \langle s3, 130\rangle \xrightarrow{100,1} \ldots .\end{aligned}$$

For a given infinite divergent path ω of $\mathcal{M}_G$, for an atomic proposition $P \in AP$, let us define a $\{0, 1\}$-valued function $P_\omega : \mathbb{R}^{\geq 0} \to \{0, 1\}$ by $P_\omega(t) = \max\{q_\omega(t) \mid q = \langle s, \nu \rangle \in Q_G$ and $P \in \mathcal{L}(s)\}$ (note that there can be several regions $\langle s, \nu \rangle$ in the path ω for which $P \in \mathcal{L}(s)$). So, $P_\omega(t) = 1$ means that there is an semi-interval $(t - \delta, t]$ in which P holds. Otherwise, $P_\omega(t) = 0$. Since we have assumed that ω is divergent, P_ω has the finite variability, i.e. it has only finite number of discontinuity points within any finite interval.

3. Probabilistic Duration Calculus

In this section we introduce a simple form of Probabilistic Duration Calculus. A complete probabilistic interval logic (which DC is based on) with a proof system has been introduced in [5]. However the definition of the semantics in that paper for the calculus is rather complicated and less intuitive. The calculus introduced in this paper has an intuitive semantics based on probabilistic timed automata, and has a simple grammar that allows to write formulas to reason about the probability of the satisfaction of a duration formula by a probabilistic timed automaton as well as to specify real-time properties of the system itself.

Definition 6. *Let* R *stand for relations (e.g.* $\leq, =$*), and* F *stand for functions (e.g.* $+, -$*). The syntax of Probabilistic Duration Calculus is defined as follows.*

$$\begin{aligned}\Phi \;&::=\; \Psi \mid [\Psi]_{\sqsupseteq\lambda} \mid \neg\Phi \mid \Phi \wedge \Phi, \\ \Psi \;&::=\; R(\eta, \ldots, \eta) \mid \neg\Psi \mid \Psi \wedge \Psi \mid \Psi; \Psi, \\ \eta \;&::=\; \textstyle\int S \mid F(\eta, \ldots, \eta), \\ S \;&::=\; \mathbf{1} \mid P \mid \neg S \mid S \wedge S,\end{aligned}$$

where Φ *stands for Probabilistic Duration Calculus formulas,* Ψ *stands for Duration Calculus formulas,* η *stands for duration terms,* S *stands for state expressions, and* P *is a symbol in the set of atomic proposition* AP.

We will use a probabilistic timed automaton G as underlying model to define the semantics for Probabilistic Duration Calculus formulas as well as for Duration Calculus formulas. Let *Intv* denote the set of all intervals on $\mathbb{R}^{\geq 0}$.

Given a path ω of $\mathcal{M}_G$ according to a strategy A. The interpretation of state expression S is a $\{0, 1\}$-valued function $I^\omega_S : \mathbb{R}^{\geq 0} \to \{0, 1\}$ defined inductively as: $I^\omega_{\mathbf{1}}(t) = 1$ for all $t \in \mathbb{R}^{\geq 0}$, $I^\omega_P = P_\omega$ where P_ω is defined as in Section 2, $I^\omega_{\neg S} = 1 - I^\omega_S$, and $I^\omega_{S1 \wedge S2} = \min\{I^\omega_{S1}, I^\omega_{S2}\}$. (Note that the operations on functions is defined point-wise.) The interpretation of a term η is a function $I^\omega_\eta : Intv \to \mathbb{R}^{\geq 0}$ defined as $I^\omega_{\int S}([a, b]) = \int_a^b I^\omega_S(t)dt$, and $I^\omega_{f(\eta 1, \ldots, \eta k)}([a, b]) = f(I^\omega_{\eta 1}([a, b]), \ldots, I^\omega_{\eta k}([a, b]))$ for any interval $[a, b] \in Intv$.

A model for DC formulas is a pair $(\omega, [a, b])$ of a divergent path ω and an interval $[a, b]$. The semantics of Duration Calculus formulas is essentially the satisfaction relation $\models$ between a model $(\omega, [a, b])$ and a DC formula Ψ which is defined as follows.

- $(\omega, [a, b]) \models R(\eta 1, \ldots, \eta k)$ iff $R(I^\omega_{\eta 1}([a, b]), \ldots, I^\omega_{\eta k}([a, b]))$,

- $(\omega, [a, b]) \models \neg\Psi$ iff $(\omega, [a, b]) \not\models \Psi$,
- $(\omega, [a, b]) \models \Psi 1 \wedge \Psi 2$ iff $(\omega, [a, b]) \models \Psi 1$ and $(\omega, [a, b]) \models \Psi 2$,
- $(\omega, [a, b]) \models \Psi 1; \Psi 2$ iff $(\omega, [a, m]) \models \Psi 1$ and $(\omega, [m, b]) \models \Psi 2$ for some $m \in [a, b]$.

The probability measure $Prob^A$ will come to play role in the definition of semantics of PDC formulas. A model for a PDC formula consists of a strategy A of $\mathcal{M}_G$ and a time point t (recall that A defines an "initial" state, not necessary to be $\langle \bar{s}, \mathbf{0} \rangle$; to be meaningful, we may need the restriction that the "initial" state of A is $\langle \bar{s}, \mathbf{0} \rangle$, we will assume this whenever necessary). The satisfaction relation $\models_{PDC}$ between PDC models (A, t) and PDC formulas Φ is defined as:

- For a DC formula Ψ, $(A, t) \models_{PDC} \Psi$ iff $Prob^A(\{\omega \mid \omega \in Path^A_{inf}$ and ω is divergent and $(\omega, [0, t]) \models \Psi\}) = 1$,
- For a DC formula Ψ, $(A, t) \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ iff $Prob^A(\{\omega \mid \omega \in Path^A_{inf}$ and ω is divergent and $(\omega, [0, t]) \models \Psi\}) \geq \lambda$,
- $(A, t) \models_{PDC} \neg\Phi$ iff $(A, t) \not\models_{PDC} \Phi$
- $(A, t) \models_{PDC} \Phi 1 \wedge \Phi 2$ iff $(A, t) \models_{PDC} \Phi 1$ and $(A, t) \models_{PDC} \Phi 2$.

The reason for a using a strategy to define a model of PDC formulas is clear since the probability is defined just for subsets of paths induced by A, not for a single path. But the reason for selecting an interval of the form $[0, t]$ instead of $[a, b]$ is just for convenience. The computation of $Prob^A(B)$ for a set B of paths satisfying a DC formula Ψ in an interval $[a, b]$ needs the prefixes in the whole interval $[0, b]$ of paths in B. Intuitively, a strategy A of probabilistic timed automaton G satisfies a DC formula Ψ in the probabilistic setting at a time t iff the set of infinite divergent paths ω produced by A that satisfy Ψ in the interval $[0, t]$ has the probability 1.

A DC formula Φ is said to be valid iff $(\omega, [a, b]) \models \Phi$ holds for any probabilistic timed automaton G, any path ω of G, and any time interval $[a, b]$. A PDC formula Φ is said to be valid iff $(A, t) \models_{PDC} \Phi$ holds for any probabilistic timed automaton G, strategy A of G, and $t \in \mathbb{R}^{\geq 0}$. In [17, 2] a proof system for DC for deriving valid formulas has been presented. It follows directly from the definition of semantics that PDC is a conservative extension of DC. Besides, some obvious properties of the probabilities can be translated into valid formulas in PDC easily.

These observations are formulated in the following theorem.

Theorem 1. *For any DC formulas* Φ, $\Phi 1$ *and* $\Phi 2$

- $[\Phi]_{\sqsupseteq 1} \Leftrightarrow \Phi$ *is a valid PDC formula,*
- *If* Φ *is a valid DC formula, then it is a valid PDC formula,*
- $((\Phi 1 \Rightarrow \Phi 2) \wedge [\Phi 1]_{\sqsupseteq\lambda}) \Rightarrow [\Phi 2]_{\sqsupseteq\lambda}$ *is a valid PDC formula*
- $\neg(\Phi 1 \wedge \Phi 2) \wedge [\Phi 1]_{\sqsupseteq\lambda 1} \wedge [\Phi 2]_{\sqsupseteq\lambda 2} \Rightarrow [\Phi 1 \vee \Phi 2]_{\sqsupseteq\lambda 1+\lambda 2}$ *is a valid PDC formula.*

Proof. Straightforward from the definition of semantics of DC and PDC. □

As usual in DC, we use the following abbreviations: $\ell \widehat{=} \int \mathbf{1}$, $True \widehat{=} \ell \geq 0$, $\Diamond\Psi \widehat{=} True; \Psi; True$ (there exists a subinterval for which Ψ is satisfied), $\Box\Psi \widehat{=} \neg\Diamond\neg\Psi$ (for all subintervals Ψ is satisfied), $\lceil S \rceil \widehat{=} \int S = \ell \wedge \ell > 0$.

Note that PDC can express the safety and bounded liveness properties, but not unbounded liveness properties. For example, PDC formula $\Box(\lceil P \rceil; \ell > b \Rightarrow \ell \leq b; \lceil Q \rceil)$ says that it is almost certain that whenever P becomes true for non-zero time period, Q must become true for non-zero time period within b time units.

Example 3. Let us consider the simple gas burner in Example 1 (see Fig. 1). Let one of the requirements for the gas burner is that for any observation interval the length of which is not shorter than 60 seconds, the accumulated leakage time is not longer than 4% of the length of the observation interval. This requirement is formalized as a DC formula $R \widehat{=} \Box(\ell \geq 60 \Rightarrow \int leak \leq 4\% * \ell)$. ($\widehat{=}$ stands for "being by

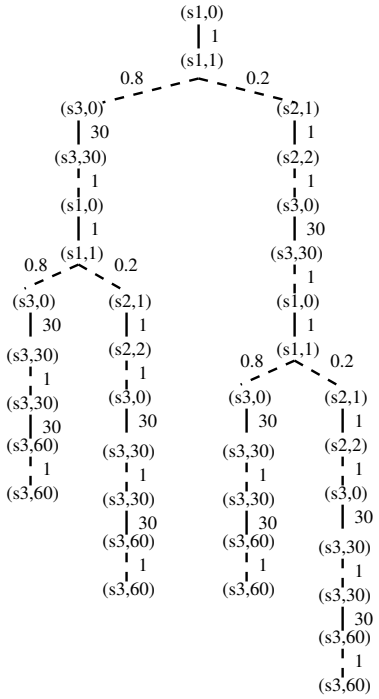

Fig. 2: A part of a strategy A for the simple gas burner.

definition"). Let ω be given as in Example 2. Then, $(\omega, [0, 60]) \not\models (\ell \geq 60 \Rightarrow \int leak \leq 4\% * \ell)$. This is so because the accumulated time for the leakage in the interval $[0, 60]$ is $.9 + .7 + 1.2 = 2.8$ which is longer than $4\% * 60$ $(= 2.4)$.

Let strategy A that schedules the system producing the paths be as shown by the tree in Fig. 2 in which the dashed edges represent discrete transitions labeled with probability, and the non-dashed edges represent time advance transitions labeled with their corresponding amount of time units. Only those paths that have a prefix represented by the leftmost branch of the tree, satisfy the requirement R in the interval $[0, 60]$. The set of these paths has the probability $0.8 * 0.8 = 0.64$. Hence, $(A, 60) \models [R]_{\sqsupseteq .6}$ (note that this example is for the sake of illustrating the concepts only).

4. Model checking probabilistic timed automata against PDC properties

Duration Calculus formulas are highly undecidable, only a very small class of chop free formulas is decidable (see [18]). In this section, we develop a technique to verify if a set of all PDC models generated by a probabilistic timed automaton G satisfies a PDC formula in discrete time. Namely, we consider the problem to decide $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ for all $A \in \mathcal{A}$ and all $t \in \mathbb{R}^{\geq 0}$, where $\mathcal{A}$ is the set af all integral strategies of a timed automaton G. In the sequel, for simplicity by saying "strategy" we actually mean "integral strategy" unless differently stated.

Depending on different forms of model sets we can have different model checking problems as:

1. Single strategy single time: given a strategy A, given a time t, to decide $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$. This problem is decidable. It is so because the fact that a path ω satisfies Ψ in $[0, t]$ or not depends only on the smallest prefix $\omega^{(i)}$ such that $\mathcal{D}_\omega(i) \geq t$. The set $\{\omega' \mid \omega' \in Path^A_{fin}$ and $\mathcal{D}_{\omega'}(|\omega'|) \geq t$ and $\mathcal{D}_{\omega'}(|\omega'| - 1) < t\}$ is finite, and computable if A is computable. From the assumption, the set $\{\omega' \mid \omega' \in Path^A_{fin}$ and $\mathcal{D}_{\omega'}(|\omega'|) \geq t$ and $\mathcal{D}_{\omega'}(|\omega'|-1) < t$ and $(\omega', [0, t]) \models \Psi\}$ is computable, and finite. Hence, $Prob^A(\{\omega \in Path^A_{fin} \mid (\omega, [0, t]) \models \Psi\})$ is computable, and therefore, $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ is decidable.
2. Multiple strategy single time: Given a set of strategies $\mathcal{A}$ which have a finite representation, given a time t, decide $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ for all $A \in \mathcal{A}$. If $\mathcal{A}$ is finite, the problem is decidable. Hence the decidability of the problem depends on the form of the computable set $\mathcal{A}$ of strategies.
3. Single strategy with arbitrary time: Given a strategy A which has a finite representation, decide if $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ for all $t \in \mathbb{R}^{\geq 0}$. This problem in general is undecidable even for $\lambda = 1$ because DC is undecidable in general.
4. Multiple strategy with arbitrary time: Given a set of strategies $\mathcal{A}$ which have a finite representation, decide $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ for all $A \in \mathcal{A}$ and all $t \in \mathbb{R}^{\geq 0}$. This problem is most general, and undecidable because DC is undecidable in general.
5. Strategy synthesis: To find a strategy A such that $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ for a given t or for all t.

In this section, we will restrict ourselves to some instances of the problems mentioned in the items 3 and 4.

We are interested specially in the PDC formulas of the form $[\Psi]_{\sqsupseteq\lambda}$, where Ψ has the form $\Box(a \leq \ell \leq b \Rightarrow \sum_{i=1}^{k} c_i \int P_i \leq M)$ called linear duration invariants (LDI) [7], where M, a and b are integers, b could be ∞. A dependability requirement for the simple gas burner could be expressed as $[\Box(\ell \geq 60 \Rightarrow \int leak \leq 4\% * \ell)]_{\sqsupseteq.99}$ which says that with the probability .99, the accumulated time for gas leaking is not more than 4% of the observation time whenever the observation time is longer than 60 seconds. So, the $(A, [0, 10^5]) \models [\Box(\ell \geq 60 \Rightarrow \int leak \leq 4\% * \ell)]_{\sqsupseteq.99}$ for any strategy A says about the reliability of the gas burner: its requirement is satisfied with the probability .99 whenever it is operated for less than 10^5 seconds.

For simplicity and as motivated by the discretisability of LDI [9] (i.e. an LDI is satisfied by all models if and only if it is satisfied by all integral models), we restrict ourselves to those strategies in which each transition is of the form (t, p) where $t \in \mathbb{N}$ only.

Now, we recall a very important technique from timed automata with some adaptations to probabilistic timed automata. Let, in the sequel, G be a PTA.

Integral Region Graph. The key idea for reducing the state space of timed automata to a finite space is the clock equivalence relation introduced in [16]. In this subsection we recall this standard notions restricted to the set $\mathbb{N}^C$ of integral clock valuations. Let c be the max of integers occurring in clock constraints in G.

Definition 7. *The valuations $\nu, \nu' \in \mathbb{N}^C$ are clock equivalent, denoted by $\nu \cong \nu'$ iff*

1. *$\forall x \in C$, either $\nu(x) = \nu'(x)$, or both $\nu(x) > c$ and $\nu'(x) > c$,*
2. *$\forall x, x' \in C$, either $\nu(x) - \nu(x') = \nu'(x) - \nu'(x')$, or both $\nu(x) - \nu(x') > c$ and $\nu'(x) - \nu'(x') > c$*

One important property of the clock equivalence relation $\cong$ is that it has finite index and the valuations from the same equivalence class satisfy the same set of clock constraints as formulated as the following lemma (taken from [16, 9]):

Lemma 1. *Let $\nu, \nu' \in \mathbb{N}^C$, $X \in 2^C$, and $\nu \cong \nu'$. Then*

1. *$\nu[X := 0] \cong \nu'[X := 0]$*
2. *for any zone $\zeta \in \mathbf{Z}_C(G)$ appearing in the description of G, ν satisfies ζ if and only if ν' satisfies ζ.*

Let $\mathcal{G}$ be the set of all equivalence classes of $\cong$. An equivalence class $\alpha \in \mathcal{G}$ satisfies a clock constraint $\zeta \in \mathbf{Z}_C(G)$ iff ν satisfies ζ for some $\nu \in \alpha$. From the item 2 of Lemma 1, it follows that α satisfies a clock constraint ζ if and only if ν satisfies ζ for any $\nu \in \alpha$. An equivalence class β is said to be the successor of an equivalence class α, denoted by $succ(\alpha)$ iff for each $\nu \in \alpha$, there exists $t \in \mathbb{N}$ such that $\nu + t \in \beta$ and $\nu + t' \in \alpha \cup \beta$ for all $t' \leq t$ and $t' \in \mathbb{N}$. Let $d_\alpha = \sup\{t \in \mathbb{N} \mid \nu \in \alpha \text{ and } \nu + t \in succ(\alpha) \text{ and } \nu + t' \in \alpha \cup \beta \text{ for all } t' \leq t \text{ and } t' \in \mathbb{N}\}$. It follows from the definition of $succ(\alpha)$ that either $d_\alpha = 1$ or $d_\alpha = \infty$. The latter happens only when $succ(\alpha)$ satisfies $x > c$ for all $x \in C$. The nondeterministic discrete time behaviors of PTA G can now be described by the region graph $R(G)$ defined as follows.

Definition 8. *The region graph $R(G)$ is the Markov decision process $(V^*, Steps^*, L^*)$, where*

- *the vertex set $V^* \widehat{=} \{\langle s, \alpha\rangle \mid s \in \mathcal{S}$ and $\alpha \in \mathcal{G}$ and α satisfies $inv(s)\}$, and*
- *the transition function $Steps^* : V^* \to 2^{\mathbb{N} \times \mu(V^*)}$ is defined as follows. For each vertex $\langle s, \alpha\rangle \in V^*$:*

  1. *If the invariant condition $inv(s)$ is satisfied by $succ(\alpha)$ then for any $\langle s', \beta\rangle \in V^*$, let $p^{s,\alpha}_{succ}(\langle s', \beta\rangle) = \begin{cases} 1 & \text{if } \langle s', \beta\rangle = \langle s, succ(\alpha)\rangle, \\ 0 & \text{otherwise.} \end{cases}$*

  Then $(t, p^{s,\alpha}_{succ}) \in Steps^*(s, \alpha)$ for any $t \in \mathbb{N}$, $0 < t \leq d_\alpha$. In this case, we say $type(p^{s,\alpha}_{succ}) = \top$.

2. *If there exists* $p' \in prob(s)$ *such that* α *satisfies the enabling condition* $\tau_s(p')$*, then for any* $\langle s', \beta\rangle \in V^*$ *let* $p^{s,\alpha}_{p'}(\langle s', \beta\rangle) = \sum_{X \subseteq C, \alpha[X:=0]=\beta} p'(s', X)$*. Then,* $(0, p^{s,\alpha}_{p'}) \in Steps^*(\langle s, \alpha\rangle)$*. In this case, we say* $\mathtt{type}(p^{s,\alpha}_{p'}) = p'$*.*

In the definition of $Steps^*$ the item (1) represents the time transitions, and the item (2) represents the discrete transitions.

Definition 9. *A strategy* A^* *on the region graph is a function mapping every nonempty finite path* ω^* *of* $R(G)$ *to a pair of integral time* t *and distribution* p *such that* $(t, p) \in Steps^*(last(\omega^*))$*, and mapping* ϵ *to* $\langle \bar{s}, \mathbf{0}\rangle$*.*

By the definition of transition function $Steps^*$, the number of the (time) transitions of $R(G)$ between a node (s, α) and $(s, succ(\alpha))$ is infinite when $d_\alpha = \infty$. In the graph, those transitions are combined into one transition which is labeled by $(*, 1)$, where 1 is the probability distribution assigning probability 1 to the transition from (s, α) to $(s, succ(\alpha))$. This transition expresses that we can choose nondeterministically an arbitrary integer for time step, and then with the probability 1, move to the region $(s, succ(\alpha))$. Therefore, a strategy A of $R(G)$ will replace $*$ by an integer each time it travels through this transition. From the definition of the region graph $R(G)$ and the timed structure $\mathcal{M}_G$, the paths in $R(G)$ and the paths in $\mathcal{M}_G$ are closely related. Namely, if in $\mathcal{M}_G$ there is a transition $\langle s, \nu\rangle \xrightarrow{t,\bar{p}} \langle s', \nu'\rangle$, where $\mathtt{type}(\bar{p}) = p'$ and $t \in \mathbb{N}$ then in $R(G)$ there is a path $\langle s, \alpha_0\rangle \xrightarrow{t_1,p_1} \ldots \xrightarrow{t_k,p_k} \langle s, \alpha_k\rangle \xrightarrow{0,p^{s,\alpha_k}_{p'}} \langle s', \beta\rangle$ such that $\mathtt{type}(p_i) = \top$, $\alpha_i = succ(\alpha_{i-1})$ for $1 \le i \le k$, $\mathtt{type}(p^{s,\alpha_k}_{p'}) = p'$, $\nu \in \alpha_0$, $\nu' \in \beta$, $inv(s)$ is satisfied by all α_i, $t = t_1 + \ldots + t_k$, and α_k satisfies $\tau_s(p')$. Furthermore, if in $\mathcal{M}_G$ there is a transition $\langle s, \nu\rangle \xrightarrow{t,\bar{p}} \langle s, \nu'\rangle$ where $\mathtt{type}(\bar{p}) = \top$ and $t \in \mathbb{N}$ then in $R(G)$ there is a path $\langle s, \alpha_0\rangle \xrightarrow{t_1,p_1} \ldots \xrightarrow{t_k,p_k} \langle s, \alpha_k\rangle$ such that $\mathtt{type}(p_i) = \top$, for $1 \le i \le k$, $\alpha_i = succ(\alpha_{i-1})$ and satisfies $inv(s)$, $\nu \in \alpha_0$, $\nu' \in \alpha_k$, $t = t_1 + \ldots + t_k$.

Conversely, for each transition in $R(G)$ of the form $\langle s, \alpha\rangle \xrightarrow{t,p^{s,\alpha}} \langle s', \beta\rangle$, for any $\nu \in \alpha$ there is a transition $\langle s, \nu\rangle \xrightarrow{t,\bar{p}} \langle s', \nu'\rangle$ in $\mathcal{M}_G$ with $\mathtt{type}(\bar{p}) = \mathtt{type}(p^{s,\alpha})$ and $\nu' \in \beta$.

From this observation each strategy A^* of $R(G)$ corresponds one-to-one with an integral strategy A of $\mathcal{M}_G$ in a sense that will be made precise soon.

With each strategy A^* of $R(G)$ we can associate a Markov chain $MC^{A^*} = (Path^{A^*}_{fin}, \mathbf{P}^{A^*})$ where for $\omega^*, \omega'^* \in Path^{A^*}_{fin}$ and $\langle s, \alpha\rangle, \langle s', \alpha'\rangle$ such that $last(\omega^*) = \langle s, \alpha\rangle$,

$$\mathbf{P}^{A^*}(\omega^*, \omega'^*) = \begin{cases} p^{s,\alpha} & \text{if } A^*(\omega^*) = (t, p^{s,\alpha}) \text{ and} \\ & \omega'^* = \omega^* \xrightarrow{(t,p^{s,\alpha})} \langle s', \alpha'\rangle, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the probabilistic measure $Prob^{A^*}$ on the smallest σ-algebra $\mathcal{F}^{A^*}_{Path}$ on $Path^{A^*}_{inf}$ containing the sets of the forms $\{\omega^* \mid \omega^* \in Path^{A^*}_{inf}$ and ω'^* is a prefix of $\omega^*\}$ for any $\omega'^* \in Path^{A^*}_{fin}$ is defined as before for a probabilistic timed structure. Recall that from probabilistic timed automaton G, we have defined a probabilistic timed structure $\mathcal{M}_G$ which generates the probabilistic measure $Prob^A$ on the smallest σ-algebra $\mathcal{F}^A_{Path}$ on $Path^A_{inf}$. From the relationship between strategies A^* of $R(G)$ and strategies A of $\mathcal{M}_G$ observed earlier we can derive a relationship for $Prob^{A^*}$ and $Prob^A$ which plays key role in model checking PDC formulas. The relation between $R(G)$ and $\mathcal{M}_G$ is expressed formally as:

Lemma 2. *Let* A *be an integral strategy of probabilistic timed automaton* G *(i.e. an integral strategy of* $\mathcal{M}_G$*). Then, there exists an strategy* A^* *of the integral region graph* $R(G)$ *and an one-to-one mappings* $\gamma : Path^A_{inf} \to Path^{A^*}_{inf}$ *such that:*

1. $Prob^A(\Omega) = Prob^{A^*}(\gamma(\Omega))$ *for all* $\Omega \in \mathcal{F}^A_{Path}$*,*
2. $P_\omega(t) = P_{\gamma(\omega)}(t)$ *almost everywhere in* $\mathbb{R}^{\ge 0}$ *for all* $\omega \in Path^A_{inf}$

Proof. Let γ be the homomorphism defined from the relation between transitions in $\mathcal{M}_G$ and $R(G)$ observed as above. Given strategy A, strategy A^* is defined based on mapping γ which simulates

A by splitting one step (t, p) into several time steps $(1, 1), \ldots, (1, 1), (0, p)$ as given by mapping γ. Item 2 follows directly from the construction of A^*, and Item 1 follows from the fact that for all $\omega \in Path^A_{fin}$, $Prob^A_{fin}(\omega) = Prob^{A^*}_{fin}(\gamma(\omega))$. The detailed proof is omitted here. □

Item 2 of Lemma 2 implies that $(\omega, [a, b]) \models \Psi$ if and only if $(\gamma(\omega), [a, b]) \models \Psi$ for any DC formula Ψ, for any $\omega \in Path^A_{inf}$ and interval $[a, b]$. Combined with Item 1, this implies that $A, t \models_{PDC} \Phi$ if and only if $A^*, t \models_{PDC} \Phi$ for any PDC formula Φ and $t \in \mathbb{R}^{\geq 0}$.

Depending on how integral strategy A of G is given, the corresponding strategy A^* of $R(G)$ can be found easily based on A. For simplicity, firstly we consider the problem to decide if $A, t \models_{PDC} \Phi$ for $t \in \mathbb{R}^{\geq 0}$. Now consider the following case for PDC formula Φ:

$$\Phi = [\Psi]_{\sqsupseteq\lambda}, \quad \Psi = \Box\Psi 1 \tag{1}$$

where $\Psi 1$ is a DC formula (to be more general Ψ is not necessary to be LDI). We have that

$$\left\{\omega \;\middle|\; \begin{array}{l} \omega \in Path^{A^*}_{inf} \text{ and } \omega \text{ is divergent and} \\ (\omega, [0, n]) \models \Psi \text{ for all } n \in \mathbb{N} \end{array}\right\}$$

$$= \bigcap_{n\geq 0} \left\{\omega \;\middle|\; \begin{array}{l} \omega \in Path^{A^*}_{inf} \text{ and } \omega \text{ is} \\ \text{divergent and } (\omega, [0, n]) \models \Psi \end{array}\right\}.$$

Because the set sequence $\{\omega \mid \omega \in Path^{A^*}_{inf}$ and ω is divergent and $(\omega, [0, n]) \models \Psi\}$ is decreasingly monotonic (according to the set inclusion relation) when n increases, we have that $Prob^{A^*}(\{\omega \mid \omega \in Path^{A^*}_{inf}$ and ω is divergent and $(\omega, [0, n]) \models \Psi$ for all $n \in \mathbb{N}\}) = \inf_{n\in\mathbb{N}}\{Prob^{A^*}(\{\omega \mid \omega \in Path^{A^*}_{inf}$ and ω is divergent and $(\omega, [0, n]) \models \Psi\})$.

Hence, if we can compute $Prob^{A^*}(\{\omega \mid \omega \in Path^{A^*}_{inf}$ and ω is divergent and $(\omega, [0, n]) \models \Psi$ for all $n \in \mathbb{N}\})$, we can solve the problem to decide if $A^*, t \models \Phi$ for all $t \geq 0$.

Let $\mathcal{P}$ be a path in the region graph $R(G)$ that generates a DC model not satisfying $\Psi 1$. Assume that a path in $Path^{A^*}_{inf}$ that does not satisfy DC formula Ψ in an interval if and only if it has a prefix that includes $\mathcal{P}$. Then all the paths in $Path^{A^*}_{inf}$ that satisfy Ψ for any interval are those that do not include $\mathcal{P}$. From integral graph $R(G)$, we can find all such paths $\mathcal{P}$ that can generate a DC model not satisfying $\Psi 1$, and can construct a graph that generate all the paths in $Path^{A^*}_{inf}$ that do not include any such path $\mathcal{P}$ (i.e. those paths that satisfy Ψ for any interval). We assume that any two paths in $\mathcal{P}$ are not nested (if for two paths in $\mathcal{P}$, one is nested in the other, we can remove the later without changing the meaning of $\mathcal{P}$). From the labels of the constructed graph, the probability of the set of paths can be calculated. To apply this procedure we need: (a) a technique to construct the finite set of paths $\mathcal{P}$ in $R(G)$ that correspond to all DC models that do not satisfy $\Psi 1$, (b) the set of paths in $Path^{A^*}_{inf}$ that do not include any such path $\mathcal{P}$ are finitely representable by a graph, and (c) a technique to compute the probability of the set of infinite paths resulting from item (b).

Regarding Item (a), the following lemma is from [9, 10], which says that given a linear duration invariant Ψ, the set of paths that do not satisfy Ψ is computable by searching in $R(G)$.

Lemma 3.

1. *Given a path $\omega \in Path^{A^*}_{inf}$. A linear duration invariant Ψ is satisfied by model $(\omega, [a, b])$ for any interval $[a, b]$ if and only if it is satisfied by model $(\omega, [m, n])$ for any integral interval $[m, n]$.*
2. *The set of paths of integral region graph $R(G)$ that correspond to a DC integral model that does not satisfy Ψ is constructable.*

Regarding Item (b), we have to restrict ourselves to the class of so-called finitely representable strategies A^* of the region graph $R(G)$. A strategy A^* of $R(G)$ is finitely representable iff for any path ω^* of $R(G)$ the value of $A^*(\omega^*)$ depends only on the suffix of the length k of ω^* for a fixed k. An finitely representable strategy A^* of $R(G)$ for the case $k = 1$ is called simple strategy. Such a finitely representable strategy will be represented by a graph with no nondeterminism, complete probabilistic choices, and fully embedded in $R(G)$.

Definition 10. *Given a finitely representable strategy A^*. A graph representation of A^* is a deterministic Markov decision process $G(A^*)$ =*

$(V_{A^*}, Steps_{A^*}, L_{A^*})$ *which is embedded in the region graph* $R(G) = (V^*, Steps^*, L^*)$ *by a mapping* ρ*, where* $\rho : V_{A^*} \to V^*$*, and the following conditions are satisfied:*

- *There is an initial node called* v_0*, and* $\rho(v_0) = \langle \bar{s}, \boldsymbol{0} \rangle$,.
- $G(A^*)$ *is deterministic, i.e.* $Steps_{A^*}(v)$ *has only one element, denoted by* $Steps_{A^*}(v)$ *itself,*
- $L_{A^*}(v) = L^*(\rho(v))$ *for all* $v \in V_{A^*}$
- *Let* $Steps_{A^*}(v) = (t, p)$*, where* p *is a distribution in* $\mu(V_{A^*})$*. The restriction of* ρ *on* $\{v' \in V_{A^*} \mid p(v') > 0\}$ *is an one-to-one mapping, and the distribution* ρ_p *defined by* $\forall s \in V^* \bullet \rho_p(s) = \max\{p(v') \mid \rho(v') = s\}$ *(by our convention,* $\max \emptyset = 0$*) is a distribution in* $\mu(V^*)$*, and* $(t, \rho_p) \in Steps^*(\rho(v))$.

Figure 3 shows the integral region graph of Simple Gas Burner in Fig 1 and graph representations for finitely representable strategies A_1^* and A_2^*. The embedding mapping ρ maps a node in A_1^* and A_2^* to the node with the same label in the integral region graph.

Regarding Item (c) of the condition for applying the checking procedure, we have

Lemma 4. *Given a graph representation of a finitely representable strategy* A^**,* $G(A^*) = (V_{A^*}, Steps_{A^*}, L_{A^*})$*. Given a finite set* $\mathcal{P}$ *of finite paths of* $G(A^*)$*. Let* Ω *be the set of all infinite paths of* $G(A^*)$ *starting from* v_0 *which do not include any path in* $\mathcal{P}$*. The probability* $Prob^{A^*}(\Omega)$ *is computable.*

Proof. Let $\Delta(v)$ be the set of all infinite paths of $G(A^*)$ starting from v which do not include any path in $\mathcal{P}$, A_v^* be the strategy represented by $G(A^*)$ with v as initial node, and $P(v) = Prob^{A_v^*}(\Delta(v))$. Let for each v, $\mathcal{P}(v) = \{\omega'' | \omega'' \in \mathcal{P}$ and ω'' starts from $v\}$. Let v^+ be the set of one-step paths formed by outgoing edges of v. Then, $\Delta(v)$ satisfies: $\Delta(v) =$

$(\cup_{e \in v^+}(e\Delta(last(e)))) \setminus (\cup_{e\omega \in \mathcal{P}(v)} e\omega\Delta(last(\omega)))$.

Although all paths in $\mathcal{P}$ are not nested in one another, but some of them may overlap some suffixes of ω for a given finite path ω. Let $\mathcal{P}_\omega$ be the set of those such paths of $\mathcal{P}$, $\mathcal{P}_\omega = \{\omega' \in \mathcal{P} | \omega' = xz$ and $\omega = yx$ for some paths $x \neq \epsilon, y, z\}$. Then

$\omega\Delta(last(\omega)) \setminus \Delta(last(e)) =$

$\cup_{\omega' \in \mathcal{P}_\omega}(\omega \ominus \omega')\omega'\Delta(last(\omega'))$,

where for $\omega = yx$ $(x \neq \epsilon)$ and $\omega' = xz \in \mathcal{P}_\omega$ we define $\omega \ominus \omega' = y$. From the definition of the functions $Prob^{A_v^*}$, $v \in V_{A^*}$ it follows

$Prob^{A^*_{last(e)}}(\Delta(last(e)) \setminus \omega\Delta(last(\omega))) =$

$Prob^{A^*_{last(e)}}(\Delta(last(e)))-$

$Prob^{A^*_{last(e)}}(\omega\Delta(last(\omega)))+$

$Prob^{A^*_{last(e)}}(\cup_{\omega' \in \mathcal{P}_\omega}(\omega \ominus \omega')\omega'\Delta(last(\omega')))$

Because all paths in $\mathcal{P}$ are not nested in one another, for $e\omega, e\omega'' \in \mathcal{P}(v)$ with $\omega \neq \omega''$, we have $\omega\Delta(last(\omega)) \cap \omega''\Delta(last(\omega'')) = \emptyset$. For simplicity, we assume that for $\omega_1', \omega_2' \in \mathcal{P}_\omega$ with $\omega_1' \neq \omega_2'$, $(e(\omega \ominus \omega_1')\omega_1'\Delta(last(\omega_1'))) \cap (e(\omega \ominus \omega_2')\omega_2'\Delta(last(\omega_2'))) = \emptyset$. (without this assumption, we have to modify the technique a little). Therefore, the definition of $Prob^{A_n^*}$, $n \in V_{A^*}$ implies

$Prob^{A_v^*}(\Delta(v)) =$

$\sum_{e \in v^+} Prob^{A^*}_{fin}(e) Prob^{A^*_{last(e)}}(\Delta(last(e)))-$

$\sum_{e\omega \in \mathcal{P}(v)} Prob^{A^*}_{fin}(e\omega) Prob^{A^*_{last(\omega)}}(\Delta(last(\omega)))) +$

$\sum_{e\omega \in \mathcal{P}(v)} \sum_{\omega' \in \mathcal{P}_\omega} (Prob^{A^*}_{fin}(e(\omega \ominus \omega')\omega') \times$

$Prob^{A^*_{last(\omega')}}(\Delta(last(\omega'))))$ Let us denote $Prob^{A_v^*}(\Delta(v))$ by $P(v)$. This means that $P(v)$, $v \in V_{A^*}$ satisfy:

$P(v) =$

$\sum_{e \in v^+} Prob^{A^*}_{fin}(e) * P(last(e))-$

$\sum_{\omega \in \mathcal{P}(v)} Prob^{A^*}_{fin}(\omega) * P(last(\omega))+$

$\sum_{e\omega \in \mathcal{P}(v)} \sum_{\omega' \in \mathcal{P}_\omega} Prob^{A^*}_{fin}(e(\omega \ominus \omega')\omega')P(last(\omega'))$ and $P(v) = 1$ if no path in $\mathcal{P}$ is reachable from v. These conditions form a linear equation system for $P(v)$, $v \in V_{A^*}$. Solving it, we can find the value of $P(v_0)$ which is the value of $Prob^{A^*}(\Omega)$. □

The following theorem follows immediately from these lemmas.

Theorem 2. *For a PDC formula* Φ *of the form (1) where* Ψ *is a linear duration invariant, it is decidable whether a finitely representable*

integral strategy A of probabilistic timed automaton G satisfies Φ at any time point t.

Decision Procedure 1. Given a PTA G, given a finitely representable strategy A of $\mathcal{M}_G$, our procedure to decide if $A, t \models_{PDC} \Phi$ for all $t \in \mathbb{R}^{\geq 0}$, where $\Phi = [\Psi]_{\sqsupseteq\lambda}$, $\Psi = \Box\Psi 1$ and $\Psi 1$ is an LDI, consists of the following steps:

1. Construct the integral region graph $R(G)$ for G.
2. Construct the finitely representable strategy A^* of $R(G)$ corresponding to A according to Lemma 2.
3. Construct the set $\mathcal{P}$ of all paths $R(G)$ that corresponds to a a DC model that does not satisfy $\Psi 1$ (using the technique mentioned in Lemma 3.
4. Find a graph representation of A^* as mentioned in Definition 10.
5. Let Ω be the set of all infinite paths of $G(A^*)$ starting from v_0 which do not include any path in $\mathcal{P}$. Compute the probability $Prob^{A^*}(\Omega)$ using the technique in Lemma 4.If this probability is greater than λ, then the answer is positive. Otherwise, give the negative answer.

Note that using the same techniques, the model checking problem mentioned in Item 3 at the beginning of this section is solvable for a PDC formula Φ of the form (1) where Ψ is a formula expressing the bounded liveness $\Box(\lceil P\rceil; \ell > b \Rightarrow \ell \leq b; \lceil Q\rceil)$. In general, the problem is solvable for the case that the set of paths of integral region graph $R(G)$ that correspond to a DC integral model that does not satisfy Ψ is constructable. In [10] we proposed some form for such formulas.

Example 4. Fig. 3 shows the integral region graph $R(G)$ of the simple gas burner in Fig. 1, and Fig. 4 shows two strategies A_1^* and A_2^* of the region graph. We will decide which one among A_1^* and A_2^* satisfies the requirement R in Example 2 with a probability not lower than 0.6 using the technique mentioned above.

Any infinite path ω of strategy A_1^* that goes through the path
$\mathcal{P}_1 = (s1, 0)(s1, 1)(s2, 1)(s2, 2)$ contains a model

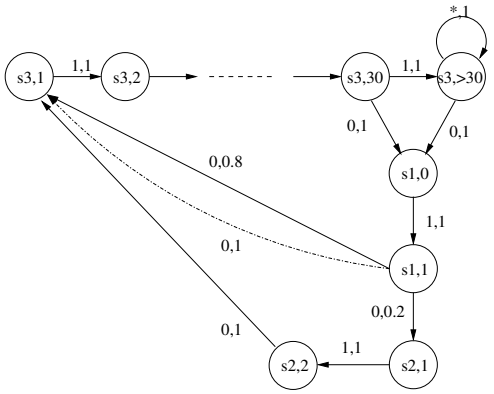

Fig. 3: Integral Region Graph for Gas Burner.

that does not satisfy R. Indeed, ω containing $\mathcal{P}_1$ should contain an interval with length 60 for which the accumulated leakage time is at least 3 ($3 > 2.4 = 4\% * 60$). Any infinite path ω of strategy A_1^* that does not contain $\mathcal{P}_1$ as a sub path satisfies R in any interval. Using the technique in the proof of Lemma 4, we have the following system of linear equations $P(\langle s1, 0\rangle) = P(\langle s1, 1\rangle - 1 * 0.2 * 1 * P(\langle s2, 2\rangle$
$P(\langle s1, 1\rangle) = 0.8P(\langle s3, 1\rangle) + 0.2P(\langle s2, 1\rangle)$
$P(\langle s2, 1\rangle) = P(\langle s2, 2\rangle) = P(\langle s3, 1\rangle) = \ldots$
$= P(\langle s1, 0\rangle)$ Solving this system, we get $P(\langle s1, 0\rangle) = 0$. Hence, we can conclude that A_1^* does not satisfies requirement $[R]_{\sqsupseteq 0.6}$.

Now consider strategy A_2^*. The linear equation system for this case is: $P(\langle s1, 0\rangle) = P(\langle s1, 1\rangle - 1 * 0.2 * 1 * P(\langle s2, 2\rangle$
$P(\langle s1, 1\rangle) = 0.8P(\langle s3, 1\rangle) + 0.2P(\langle s2, 1\rangle)$
$P(\langle s2, 1\rangle) = P(\langle s2, 2\rangle) = P(\langle s3, 1\rangle) = \ldots$
$= P(\langle (s1, 0)^{(1)}\rangle) = 1$

Solving this equation system, we have $P(\langle s1, 0\rangle) = 0.8$. Hence, $(A_2^*, t) \models_{PDC} [R]_{\sqsupseteq 0.8}$ for all $t \in \mathbb{R}^{\geq 0}$.

Now we return to our general problem mentioned at the beginning of this section. We will solve this problem by analyzing the graph $R(G)$. Let $\mathcal{A}$ be the set of all strategies of $R(G)$. For $A \in \mathcal{A}$ let Δ_A be the set of all infinite paths of A starting from the initial vertex of $R(G)$ that do not include any path in $\mathcal{P}$. Recall that in general a strategy A^* is represented as a tree, and is embedded in the graph $R(G)$ in the same way as

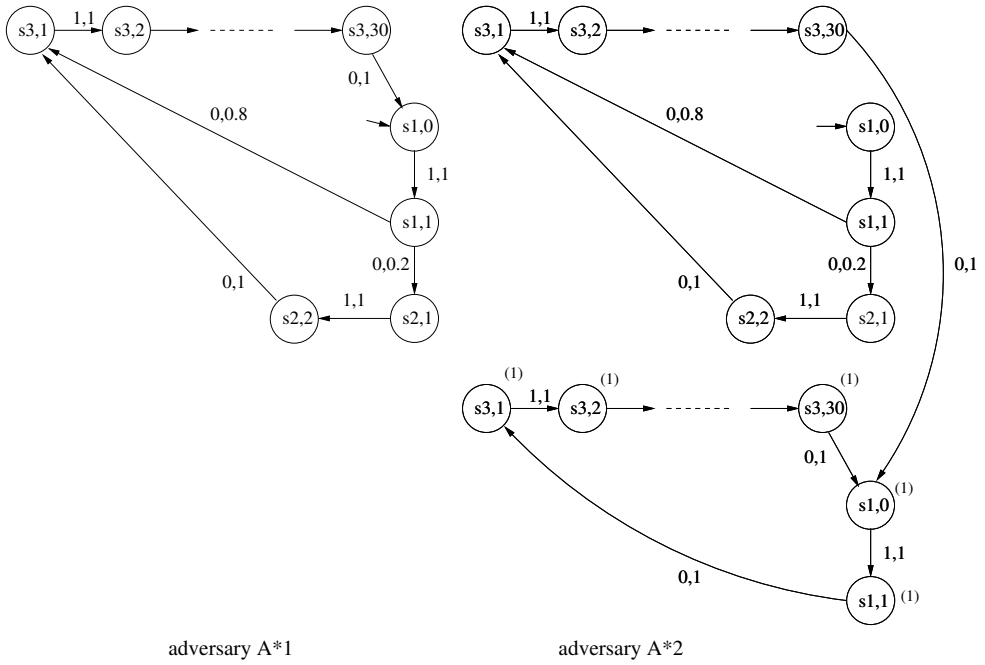

Fig. 4: Strategies A_1^* and A_2^*.

in Definition 10. Hence, we can identify a node and a path in A^* with a node and a path in $R(G)$ respectively.

For any strategy A^* a node v of A^* is said to be k-similar to a node v' of A^* iff any outgoing path with the length k of v is the same (when embedded to $R(G)$) as an outgoing path with the length k of v' and vice-versa. Since $R(G)$ is a finite graph, the number of subtrees representing probabilistic choices with the height k is finite. Hence the k-similarity relation between nodes of A^* has finite index.

Let $P_{A^*}(v)$ be the probability of the set of all infinite paths of A^* starting from the node v of the tree representation of A^* which do not include any path in $\mathcal{P}$ (with condition that the current node is v). Let for each node v in A^*, $\mathcal{P}(v)$ and $\mathcal{P}_\omega$ be defined as in the proof of Lemma 4. Let $v^+_{A^*}$ be the set of one-step paths of A^* formed by outgoing edges of v in the graph $R(G)$. Similar to the proof of Lemma 4, $P_{A^*}(v)$ satisfies:

$$P_{A^*}(v) = \sum_{e \in v^+_{A^*}} Prob^{A^*}_{fin}(e) * P_{A^*}(last(e)) - \sum_{\omega \in \mathcal{P}(v)} Prob^{A^*}_{fin}(\omega) * P_{A^*}(last(\omega)) + Prob^{A^*_v}(\cup_{e\omega \in \mathcal{P}(v)} \cup_{\omega' \in \mathcal{P}_\omega} (e(\omega \ominus \omega')\omega')\Delta(last(\omega')))$$

Let $k = 1 + \max\{1, 2|\omega| \mid \omega \in \mathcal{P}\}$. From these conditions, we have that if nodes v and v' are k-similar then $P_{A^*}(v) = P_{A^*}(v')$. Hence, we can replace v by its equivalence class of the k-similarity relation, and get a finite equation system which is the same as the one for some k-finitely representable strategy B^*. Therefore, $P_{A^*}(v_0) = P_{B^*}(v'_0)$ where v_0 and v'_0 are the root of A^* and B^* respectively. Consequently, for any strategy A^*, there is a k-finitely representable B^* such that $P_{A^*}(v_0) = P_{B^*}(v'_0)$. This ensures that $\inf\{Prob^A(\Delta_A) \mid A \in \mathcal{A}\} = \min\{Prob^A(\Delta_A) \mid A \in \mathcal{A}_k\}$ where $\mathcal{A}_k$ denotes the set of all k-finitely representable strategies in $\mathcal{A}$.

Because $\mathcal{A}_k$ is a finite set, we can use the technique in Lemma 4 to find $Prob^A(\Delta_A)$ for all $A \in \mathcal{A}_k$, and then compute $\min\{Prob^A(\Delta_A) \mid A \in \mathcal{A}_k\}$. We formulate this result as the following theorem.

Theorem 3. *For a PDC formula Φ of the form (1) where Ψ is a linear duration invariant, it is decidable whether Φ is satisfied by all integral strategies of a probabilistic timed automaton G at any time point.*

The decision procedure of this theorem is formulated as follows.

Decision Procedure 2. Given a PTA G, our procedure to decide if $A, t \models_{PDC} \Phi$ for all finitely representable strategies A of $\mathcal{M}_G$, for all $t \in \mathbb{R}^{\geq 0}$, where $\Phi = [\Psi]_{\sqsupseteq\lambda}$, $\Psi = \Box\Psi 1$ and $\Psi 1$ is an LDI, consists of the following steps:

1. Construct the integral region graph $R(G)$ for G.
2. Construct the set $\mathcal{P}$ of all paths $R(G)$ that corresponds to a a DC model that does not satisfy $\Psi 1$ (using the technique mentioned in Lemma 3. Let $k = 1 + \max\{1, 2|\omega| \mid \omega \in \mathcal{P}\}$.
3. Construct the finite set $\mathcal{A}_k$ of all k-finitely representable strategies in $\mathcal{A}$.
4. For each $A \in \mathcal{A}_k$, find $Prob^A(\Delta_A)$ using Lemma 4, where Δ_A be the set of all infinite paths of A starting from the initial vertex of $R(G)$ that do not include any path in $\mathcal{P}$.
5. Compute $\min\{Prob^A(\Delta_A) \mid A \in \mathcal{A}_k\}$. If this probability is greater than λ, then the answer is positive. Otherwise, give the negative answer.

This procedure also helps to solve the strategy synthesis problem. Namely, if we can find a strategy $A \in \mathcal{A}_k$ such that $Prob^A(\Delta_A)$ is greater than λ, then such a strategy is a solution for the strategy synthesis problem. Therefore, we have:

Theorem 4. *Given a PTA G and a PDC formula $\Phi = [\Psi]_{\sqsupseteq\lambda}$, where Ψ is an LDI, we can decide if there exists a finitely representable strategy A such that $A, t \models_{PDC} [\Psi]_{\sqsupseteq\lambda}$ for all t, and in the case such a strategy exists, we can find it.*

5. Conclusion

We have presented the problem of checking probabilistic timed automata against probabilistic duration calculus formulas. The problem is decidable for a class of PDC formulas of the form $[\Psi]_{\sqsupseteq\lambda}$ where Ψ is a linear duration invariant, or a DC formula for bounded liveness. The technique for model checking is an extension of our techniques for checking if a timed automaton satisfies a linear duration invariant using a searching method in the integral region graph of the timed automaton. The complexity of the decision procedure is high in general. Since the problem possesses a potential high complexity, we have not implemented the technique yet. Hope that with the increasing computing power in the future, we can develop an effective tool for model-checking based on the technique. At the mean time, we are looking for some special cases of the problem which are simpler and still useful for which our technique can work well, and then implement it as a tool to assist checking the dependability for embedded systems.

References

[1] Z. Chaochen, C. Hoare, A. P. Ravn, A calculus of durations, Information Processing Letters 40 (5) (1992) 269–276.

[2] C. Zhou, M. R. Hansen, Duration Calculus: A Formal Approach to Real-Time Systems, Springer-Verlag, 2004.

[3] L. Zhiming, A. Ravn, E. Sorensen, Z. Chaochen, Towards a Calculus of Systems Dependability, Journal of High Integrity Systems 1 (1) (1994) 49–65.

[4] D. V. Hung, Z. Chaochen, Probabilistic duration calculus for continuous time, Formal Asp. Comput. 11 (1) (1999) 21–44.

[5] D. P. Guelev, Probabilistic interval temporal logic and duration calculus with infinite intervals: Complete proof systems, Logical Methods in Computer Science 3 (3).

[6] D. P. Guelev, D. V. Hung, Reasoning about qos contracts in the probabilistic duration calculus, Electr. Notes Theor. Comput. Sci. 238 (6) (2010) 41–62.

[7] C. Zhou, Linear duration invariants, in: Formal Techniques in Real-Time and Fault-Tolerant Systems, Third International Symposium Organized Jointly with the Working Group Provably Correct Systems - ProCoS, Lübeck, Germany, September 19-23, Proceedings, 1994, pp. 86–109.

[8] M. Zhang, D. V. Hung, Z. Liu, Verification of linear duration invariants by model checking CTL properties, in: J. S. Fitzgerald, A. E. Haxthausen, H. Yenigün (Eds.), Theoretical Aspects of Computing - ICTAC 2008, 5th International Colloquium, Istanbul, Turkey, September 1-3, 2008. Proceedings, Vol. 5160 of Lecture Notes in Computer Science, Springer, 2008, pp. 395–409.

[9] P. H. Thai, D. V. Hung, Verifying linear duration constraints of timed automata, in: Z. Liu, K. Araki (Eds.), Theoretical Aspects of Computing - ICTAC 2004, First International Colloquium, Guiyang, China, September 20-24, 2004, Revised Selected Papers, Vol.

3407 of Lecture Notes in Computer Science, Springer, 2004, pp. 295–309.

[10] J. Zhao, D. V. Hung, Checking timed automata for linear duration properties, J. Comput. Sci. Technol. 15 (5) (2000) 423–429.

[11] C. Changil, D. V. Hung, On verification of linear occurrence properties of real-time systems, Electr. Notes Theor. Comput. Sci. 207 (2008) 107–120.

[12] M. Kwiatkowska, G. Norman, R. Segala, J. Sproston, Automatic verification of real-time systems with discrete probability distributions, Theoretical Computer Science 282 (1) (2002) 101–150.

[13] M. Kwiatkowska, D. Parker, Automated verification and strategy synthesis for probabilistic systems, in: D. V. Hung, M. Ogawa (Eds.), Automated Technology for Verification and Analysis, 11th International Symposium, ATVA 2013, Vol. 8172 of LNCS, Springer, 2013, pp. 5–22.

[14] D. V. Hung, M. Zhang, On verification of probabilistic timed automata against probabilistic duration properties, in: 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007), 21-24 August 2007, Daegu, Korea, 2007, pp. 165–172.

[15] C. Baier, M. Kwiatkowska, Model Checking for a Probabilistic Branching Time Logic with Fairness, Distributed Computing 11 (3) (1998) 125–155.

[16] R. Alur, D. Dill, A Theory of Timed Automata, Theoretical Computer Science (1994) 183–235.

[17] M. R. Hansen, C. Zhou, Duration calculus: Logical foundations, Formal Aspects of Computing 9 (1997) 283–330.

[18] Z. Chaochen, H. M. R., S. P, Decidability and Undecidability Results in Duration Calculus, in: Proc. of the 10th Annual Symposium on *Theoretical Aspects of Computer Science (STACS 93)*, no. 665 in LNCS, Springer Verlag, 1993.