On Verification of Probabilistic Timed Automata against Probabilistic Duration Properties

Dang Van Hung
International Institute for Software Technology
The United Nations University, P.O.Box 3058, Macau
email: dvh@iist.unu.edu

Miaomiao Zhang
School of Software Engineering
Tongji University
email: miaomiao@mail.tongji.edu.cn

Abstract

In this paper, we introduce an extension of Duration Calculus called Simple Probabilistic Duration Calculus (SPDC) to express dependability requirements for real-time systems, and address the problem to decide if a probabilistic timed automaton satisfies a SPDC formula. We prove that the problem is decidable for a class of SPDC called probabilistic linear duration invariants, and provide a model checking algorithm for solving this problem.

1. Introduction

Duration Calculus was introduced in [3] as a logic for specification of real-time systems. It is then developed further in many other works that have been summarized in the monograph published recently [15]. Some techniques for checking if a timed automaton satisfies a duration calculus formula written in the form of linear duration invariants have been developed [4, 12, 6, 13]. To specify the dependability of real-time systems, a kind of probabilistic extension of Duration Calculus has been introduced in [14, 8]. No rigorous syntax has been introduced in these papers, and the authors just focused on the development of techniques for reasoning instead of checking. This is because in their model, there is too much randomization and nondeterminism to perform model checking.

In this paper, we introduced a simple probabilitic 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 an adversary 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 an adversary 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, 13] 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 generalise this technique to achieve our goal with a model-checking algorithm.

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 [2, 11]. 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 $p : S \rightarrow [0,1]$ 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 $(\mathcal{Q}, Steps)$, where $\mathcal{Q}$ is a set of states, and $Steps : \mathcal{Q} \rightarrow 2^{\mu(\mathcal{Q})}$ 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 [11] we label the action selecting a probability distribution with a letter from Σ, and assume that $Steps : \mathcal{Q} \to 2^{\Sigma \times \mu(\mathcal{Q})}$ and Σ is a set of actions. The intuition is 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 $q \xrightarrow{a,p} q'$, where $a, p \in \Sigma \times \mu(Q)$ is the label of the transition. We also assume a labeling function $L : \mathcal{Q} \to 2^{AP}$, where AP is a set of atomic propositions, that associates a state q with the set of atomic propositions that hold at state q. Then, a labeled Markov decision process is a tuple $(\mathcal{Q}, Steps, L)$.

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

$$\omega \quad = \quad q_0 \xrightarrow{l_0} q_1 \xrightarrow{l_1} q_2 \xrightarrow{l_2} \ldots,$$

where q_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 $\omega = q_0 \xrightarrow{l_0} q_1 \xrightarrow{l_1} q_2 \xrightarrow{l_2} \ldots q_n$ and $\omega' = q_0' \xrightarrow{l_0'} q_1' \xrightarrow{l_1'} q_2' \xrightarrow{l_2'} \ldots$ such that $q_n = q_0'$, the concatenation of ω and ω' is defined as $\omega\omega' = q_0 \xrightarrow{l_0} q_1 \xrightarrow{l_1} q_2 \xrightarrow{l_2} \ldots q_n \xrightarrow{l_0'} q_1' \xrightarrow{l_1'} q_2' \xrightarrow{l_2'} \ldots$.

Clocks, clock valuations, clock constraints: Let $\mathbb{R}$ 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 $\mathcal{C} = \{x_1 \ldots, x_n\}$ be a set of clocks. A clock valuation is a function $\nu : \mathcal{C} \to \mathbb{R}$ that assigns a real value to each clock. Let $\mathbb{R}^{\mathcal{C}}$ denote the set of all clock valuations, and **0** denote the clock valuation that assigns 0 to each clock in $\mathcal{C}$. For a set of clocks $X \subseteq \mathcal{C}$ 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 $t \in \mathbb{R}$, we write $\nu + t$ for the clock valuation that assigns $\nu(x) + t$ to each clock $x \in \mathcal{C}$. A constraint over $\mathcal{C}$ 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 $\mathcal{C}$ is a convex subset of the valuation space $\mathbb{R}^{\mathcal{C}}$ described by a conjunction of constraints. For a zone ζ and a set of clocks $X \subseteq \mathcal{C}$ the set $\{\nu[X := 0] \mid \nu \in \zeta\}$ is also a zone, and is denoted by $\zeta[X := 0]$. Let $\mathbf{Z}_{\mathcal{C}}$ denote the set of all zones of $\mathcal{C}$.

Probabilistic timed automata and probabilistic timed structures Timed automata were introduced in [1] as a model of real-time systems. They are extended with discrete probability distribution to model probabilistic real-time systems.

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

- *a finite set $\mathcal{S}$ of nodes, a start node $\bar{s} \in \mathcal{S}$, a finite set $\mathcal{C}$ of clocks,*
- *a function $\mathcal{L} : \mathcal{S} \to 2^{AP}$ assigning to each node of the automaton the set of atomic propositions that are true in that node, a function $inv : \mathcal{S} \to \mathbf{Z}_{\mathcal{C}}$ assigning to each node an invariant condition,*
- *a function $prob : \mathcal{S} \to 2^{\mu(\mathcal{S}\times 2^{\mathcal{C}})}$ assigning to each node a set of discrete probability distributions on $\mathcal{S} \times 2^{\mathcal{C}}$,*
- *a family of functions $\langle\tau_s\rangle_{s\in\mathcal{S}}$ where, for any $s \in \mathcal{S}$, $\tau_s : prob(s) \to \mathbf{Z}_{\mathcal{C}}$ 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 timed automaton does, except that it has to select a probability distribution at each discrete step.

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

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

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

The system starts in node $s1$, with the gas valve is opened without flame being on, hence gas is leaking. At this state, there are two choices. The first choice (denoted by the dotted arrow) is that with probability 1, the flame is turned on within one second and the system moves to node $s3$ for which gas is not leaking. The second choice is as follows. With 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

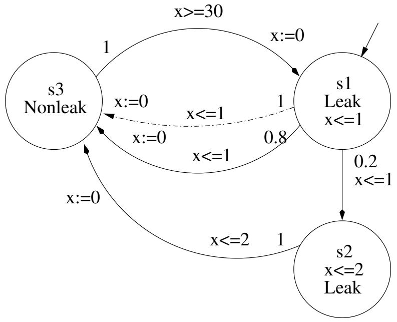

Figure 1. A probabilistic timed automaton for simple gas burner

moves to node $s3$. 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 [11] we use probabilistic timed structures as underlying model for PTA.

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

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

where $q_i \in \mathcal{Q}$, $(t_i, p_i) \in Steps(q_i)$, 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 $t \in \mathbb{R}$ 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 divergent if for any $t \in \mathbb{R}$, there exists $j \in \mathbb{N}$ such that $\mathcal{D}_\omega(j) > t$. Let ω be infinite. For each state $q \in \mathcal{Q}$, we define a $\{0, 1\}$-valued function $q_\omega : \mathbb{R} \to \{0, 1\}$ as

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

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

The concept of adversary was introduced in the literature (see, e.g. Kwiakowska [11]) 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 *An adversary (or scheduler) of a probabilistic timed structure $\mathcal{M} = (\mathcal{Q}, Steps, L)$ 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 $\mathcal{Q}$. Let $\mathcal{A}$ be the set of all adversaries of $\mathcal{M}$.*

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

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

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

From Defition 4, all ω in $Path^A_{fin}$ and $Path^A_{inf}$ start from the same state defined by $A(\epsilon)$.

A sequential Markov chain $MC^A = (Path^A_{fin}, \mathbf{P}^A)$ is associated with an adversary A, where $\mathbf{P}^A$ is defined as

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

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 $\{\omega \mid \omega \in Path^A_{inf} \text{ and } \omega' \text{ is a prefix of } \omega\}$. 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 adversaries under consideration are divergent in the probabilistic sense, i.e. we assume that for any adversary A,

$$Prob^A(\{\omega \mid \omega \in Path^A_{inf} \text{ and } \omega \text{ is divergent}\}) = 1.$$

We now define the behavior of probabilistic timed automata by associating every probabilistic timed automaton with a probabilistic timed structure.

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

- $\mathcal{Q}_G = \{\langle s, \nu\rangle \mid s \in \mathcal{S}, \nu \in \mathbb{R}^{\mathcal{C}}\}$
- *The function $Steps_G : \mathcal{Q}_G \to 2^{\mathbb{R}\times\mu(\mathcal{Q}_G)}$ assigns to each state in $\mathcal{Q}_G$ a set of transitions, each of which takes the form $(t, \bar{p})$ 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 $\langle s', \nu'\rangle \in \mathcal{Q}_G$: $\bar{p}(\langle s', \nu'\rangle) = \sum_{X\subseteq\mathcal{C}\wedge(\nu+t)[X:=0]=\nu'} p(s', X)$. For convenience, we refer to $\bar{p}$ as having type p, denoted by $\texttt{type}(\bar{p}) = p$.*
  - *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 $\langle s', \nu'\rangle \in \mathcal{Q}_G$: $\bar{p}(\langle s', \nu'\rangle) = 1$ 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. $\texttt{type}(\bar{p}) = \top$.*
- *The labeling function $L : \mathcal{Q} \to 2^{AP}$ is defined as: $L_G(\langle s, \nu\rangle) = \mathcal{L}(s)$ for all $\langle s, \nu\rangle \in \mathcal{Q}_G$.*

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

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 $P_\omega : \mathbb{R} \to \{0, 1\}$ by $P_\omega(t) = \max\{q_\omega(t) \mid q = \langle s, \nu\rangle \in \mathcal{Q}_G$ 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 [7]. 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.*

$$\begin{array}{lll}
\Phi & ::= & \Psi \mid [\Psi]_{\sqsupseteq\lambda} \mid \neg\Phi \mid \Phi \wedge \Phi, \\
\Psi & ::= & R(\eta, \dots, \eta) \mid \neg\Psi \mid \Psi \wedge \Psi \mid \Psi ; \Psi, \\
\eta & ::= & \int S \mid F(\eta, \dots, \eta), \\
S & ::= & \mathit{1} \mid P \mid \neg S \mid S \wedge S,
\end{array}$$

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 $\mathbb{R}$.

Given a path ω of $\mathcal{M}_G$ according to an adversary A. The interpretation of state expression S is a $\{0, 1\}$-valued function $I^\omega_S : \mathbb{R} \to \{0, 1\}$ defined inductively as: $I^\omega_{\mathbf{1}}(t) = 1$ for all $t \in \mathbb{R}$, $I^\omega_P = P_\omega$ where P_ω is defined as in the previous section, $I^\omega_{\neg S} = 1 - I^\omega_S$, 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 $I^\omega_\eta : Intv \to \mathbb{R}$ defined as $I^\omega_{\int S}([a, b]) = \int_a^b I^\omega_S(t)dt$, and $I^\omega_{f(\eta1,\dots,\eta k)}([a, b]) = f(I^\omega_{\eta1}([a, b]), \dots, I^\omega_{\eta k}([a, b]))$ 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.

- $(\omega, [a, b]) \models R(\eta1, \dots, \eta k)$ iff $R(I^\omega_{\eta1}([a, b]), \dots, I^\omega_{\eta k}([a, b]))$,
- $(\omega, [a, b]) \models \neg\Psi$ iff $(\omega, [a, b]) \not\models \Psi$,
- $(\omega, [a, b]) \models \Psi1 \wedge \Psi2$ iff $(\omega, [a, b]) \models \Psi1$ and $(\omega, [a, b]) \models \Psi2$,

- $(\omega, [a,b]) \models \Psi1; \Psi2$ iff $(\omega, [a,m]) \models \Psi1$ and $(\omega, [m,b]) \models \Psi2$ 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 an adversary 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 $(A,t) \not\models_{PDC} \Phi$
- $(A,t) \models_{PDC} \Phi1 \wedge \Phi2$ iff $(A,t) \models_{PDC} \Phi1$ and $(A,t) \models_{PDC} \Phi2$.

As usual in DC, we use the following abbreviations: $\ell \widehat{=} \int \mathbf{1}$, $True \widehat{=} \ell \geq 0$, $\Diamond\Psi \widehat{=} True; \Psi; True$, $\Box\Psi \widehat{=} \neg\Diamond\neg\Psi$, $\lceil S \rceil \widehat{=} \int S = \ell \wedge \ell > 0$.

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 2 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 time units, the accumulated leakage time is not longer than 4% of the length of the observation interval. This requirement is formalized as a DC formula $R \widehat{=} \Box(\ell \geq 60 \Rightarrow \int leak \leq 4\% * \ell)$.

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 [5]). 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 $t \in \mathbb{R}$, where $\mathcal{A}$ is the set af all integral adversaries of a timed automaton G.

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) [4], where M, a and b are integers, b could be ∞. For simplicity and as motivated by the discretisability of LDI [13], we restrict ourselves to those adversaries 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.

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 [1]. In this subsection we recall this standard notions restricted to the set $\mathbb{N}^{\mathcal{C}}$ of integral clock valuations. Let c be the max of integers occurring in clock constraints in G.

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

1. *$\forall x \in \mathcal{C}$, either $\nu(x) = \nu'(x)$, or both $\nu(x) > c$ and $\nu'(x) > c$,*
2. *$\forall x, x' \in \mathcal{C}$, 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 class satisfy the same set of clock constraints as formulated as the following lemma (taken from [1, 13]):

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

1. *$\nu[X := 0] \cong \nu'[X := 0]$*
2. *for any zone $\zeta \in \mathbf{Z}_{\mathcal{C}}(G)$ 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 $\zeta \in \mathbf{Z}_{\mathcal{C}}(G)$ iff ν satisfies ζ for some $\nu \in \alpha$. From the item 2 of Lemma 1, it follows that α satisfies 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 $d_\alpha = \sup\{t \in \mathbb{N} \mid \nu \in \alpha$ and $\nu + t \in succ(\alpha)$ and $\nu + t' \in \alpha \cup \beta$ for all $t' \leq t$ 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 $x \in \mathcal{C}$.

Definition 8 *The region graph $R(G)$ is defined to be 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* $\langle s,\nu\rangle \in V^*$*:*

  1. *If the invariant condition* $inv(s)$ *is satisfied by* $succ(\alpha)$ *then for any* $\langle s',\beta\rangle \in V^*$*, let*

     $$p^{s,\alpha}_{succ}(\langle s',\beta\rangle) = \begin{cases} 1 & \textit{if } \langle s',\beta\rangle = \langle s, succ(\alpha)\rangle, \\ 0 & \textit{otherwise.} \end{cases}$$

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

  2. $(0, p^{s,\alpha}_{p'}) \in Steps^*(\langle s,\alpha\rangle)$ *if there exists* $p' \in prob(s)$ *and* α *satisfies the enabling condition* $\tau_s(p')$ *such that for any* $\langle s',\beta\rangle \in V^*$*:*

     $$p^{s,\alpha}_{p'}(\langle s',\beta\rangle) = \sum_{X\subseteq\mathcal{C},\alpha[X:=0]=\beta} p'(s',X)$$

     In this case, we say $\mathtt{type}(p^{s,\alpha}_{p'}) = p'$.

Definition 9 *An adversary* 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 item 1 of the definition of transition function $Steps^*$, the number of the 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, an adversary 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. The relation between $R(G)$ and $\mathcal{M}_G$ is expressed formally as:

Lemma 2 *Let* A *be an integral adversary of probabilistic timed automaton* G *(i.e. an integral adversary of* $\mathcal{M}_G$*). Then, there exists an adversary* 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* $\mathbb{R}$ *for all* $\omega \in Path^A_{inf}$

Proof: See [9].

Item 2 of Lemma 2 follows 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 $[a,b]$. Combining with Item 1 implies that $A, t \models_{PDC} \Phi$ if and only if $A^*, t \models_{PDC} \Phi$ for any PDC formula Φ and $t \in \mathbb{R}$.

Depending on how integral adversary A of G is given, the corresponding adversary 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 $t \in \mathbb{R}$. Now consider the following case for PDC formula Φ:

$$\Phi \ = [\Psi]_{\sqsupseteq\lambda}, \quad \Psi \ = \Box\Psi1 \qquad (1)$$

where $\Psi1$ is a DC formula (to be more general Ψ is not necessary to be LDI). We have that

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

Because the set sequence

$$\{\omega \mid \omega \in Path^{A^*}_{inf} \text{and } \omega \text{ is divergent and } (\omega,[0,n]) \models \Psi\}$$

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 $\Psi1$. 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 $\Psi1$, 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 computed. 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 $\Psi1$, (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 [13, 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 $\omega \in Path_{inf}^{A^*}$. 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 adversaries A^* of the region graph $R(G)$. An adversary 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 adversary A^* of $R(G)$ for the case $k = 1$ is called simple adversary. Such a finitely representable adversary will be represented by a graph with no nondeterminism, complete probabilistic choices, and fully embedded in $R(G)$.

Definition 10 *Given a finitely representable adversary 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 $(t, \rho_p) \in Steps^*(\rho(v))$.*

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

Lemma 4 *Given a graph representation of a finitely representable adversary 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 adversary 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. Let $\mathcal{P}_\omega$ be $\{\omega' \in \mathcal{P} | \omega' = xz$ 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 $Prob^{A_n^*}$, $n \in V_{A^*}$ it follows

$$\begin{aligned}
&Prob^{A^*_{last(e)}}(\Delta(last(e)) \setminus \omega\Delta(last(\omega))) \\
&= Prob^{A^*_{last(e)}}(\Delta(last(e))) - \\
&\quad Prob^{A^*_{last(e)}}(\omega\Delta(last(\omega))) + \\
&\quad Prob^{A^*_{last(e)}}(\cup_{\omega' \in \mathcal{P}_\omega}(\omega \ominus \omega')\omega'\Delta(last(\omega')))
\end{aligned}$$

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

$$\begin{aligned}
&Prob^{A_v^*}(\Delta(v)) \\
&= \textstyle\sum_{e \in v^+} Prob_{fin}^{A^*}(e) Prob^{A^*_{last(e)}}(\Delta(last(e))) \\
&\quad - \textstyle\sum_{e\omega \in \mathcal{P}(v)} Prob_{fin}^{A^*}(e\omega) Prob^{A^*_{last(\omega)}}(\Delta(last(\omega)))) \\
&+ \textstyle\sum_{e\omega \in \mathcal{P}(v)} \sum_{\omega' \in \mathcal{P}_\omega} (Prob_{fin}^{A^*}(e(\omega \ominus \omega')\omega') \times \\
&\quad Prob^{A^*_{last(\omega')}}(\Delta(last(\omega'))))
\end{aligned}$$

This means that $P(v)$, $v \in V_{A^*}$ satisfy:

$$\begin{aligned}
&P(v) = \\
&= \textstyle\sum_{e \in v^+} Prob_{fin}^{A^*}(e) * P(last(e)) - \\
&\quad \textstyle\sum_{\omega \in \mathcal{P}(v)} Prob_{fin}^{A^*}(\omega) * P(last(\omega)) + \\
&\quad \textstyle\sum_{e\omega \in \mathcal{P}(v)} \sum_{\omega' \in \mathcal{P}_\omega} Prob_{fin}^{A^*}(e(\omega \ominus \omega')\omega') P(last(\omega'))
\end{aligned}$$

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 $P(v_0)$. □

The following theorem follows immediately from these lemmas.

Theorem 1 *For a PDC formula Φ of the form (1) where Ψ is a linear duration invariant, it is decidable whether a finitely representable integral adversary A of probabilistic timed automaton G satisfies Φ at any time point t.*

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 adversaries 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 an adversary A^*

is represented as a tree, and is embedded in the graph $R(G)$ in the same way as 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 adversary 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 subtree 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:

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

Let $k = 1 + \max\{1, 2|\omega| \ |\omega \in \mathcal{P}\}$. 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 adversary 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 adversary 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 adversaries 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 2 *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 adversaries of a probabilistic timed automaton G at any time point.*

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 $[\Psi]_{\sqsupset\lambda}$ 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.

Acknowledgement We are grateful to anonymous reviewers for their helpful comments and suggestions.

References

[1] R. Alur and D. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, pages 183–235, 1994.

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

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

[4] Z. Chaochen, Z. Jingzhong, Y. Lu, and L. Xiaoshan. Linear Duration Invariants. LNCS 863, 1994.

[5] Z. Chaochen, H. M. R., and S. P. Decidability and Undecidability Results in Duration Calculus. LNCS 665, Springer Verlag, 1993.

[6] L. X. Dong and D. V. Hung. Checking Linear Duration Invariants by Linear Programming. *Concurrency and Parallelism, Programming, Networking, and Security*. LNCS 1179, Springer, 1996, pp. 321–332.

[7] D. P. Guelev. Probabilistic Neighbourhood Logic. Proceedings FTRTFT'00, LNCS 1926, pp. 264–275.

[8] D. V. Hung and Z. Chaochen. Probabilistic Duration Calculus for Continuous Time. *Formal Aspects of Computing* (1999) 11: 21–44.

[9] D. V. Hung and Z. Miaomiao. On Verification of Probabilistic Timed Automata against Probabilistic Duration Properties. Technical Report 326, UNU-IIST, P.O.Box 3058, Macau, June 2005.

[10] Z. Jianhua and D. V. Hung. Checking Timed Automata for Some Discretisable Duration Properties. "Journal of Computer Science and Technology", Volume 15, Number 5, September 2000, pp. 423–429.

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

[12] P. H. Thai and D. V. Hung. Checking a Regular Class of Duration Calculus Models for Linear Duration Invariants. Proceedings of PDSE'98, IEEE Computer Society Press, 1998, pp. 61 – 71.

[13] P. H. Thai and D. V. Hung. Verifying Linear Duration Constraints of Timed Automata. LNCS 3407, Springer 2005, pp. 295-309.

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

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