J. SPROSTON. Discrete-Time Verification and Control for Probabilistic Rectangular Hybrid Automata, in: Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (QEST 2011), IEEE Computer Society Press, 2011, 9781457709739, pp: 79-88.

Discrete-Time Verification and Control for Probabilistic Rectangular Hybrid Automata

Jeremy Sproston
University of Turin – Italy
sproston@di.unito.it

***Abstract*—Hybrid automata provide a modeling formalism for systems characterized by a combination of discrete and continuous components. Probabilistic rectangular automata generalize the class of rectangular hybrid automata with the possibility of representing random behavior of the discrete components of the system. We consider the following two problems regarding probabilistic rectangular automata: verification concerns the computation of the maximum probability with which the system can satisfy a certain ω-regular specification; control concerns the computation of a strategy which guides certain choices of the system in order to maximize the probability of satisfying a certain ω-regular specification. Our main contribution is to give algorithms for the verification and control problems for probabilistic rectangular automata in a semantics in which discrete control transitions can occur only at integer points in time. Additionally, we give algorithms for verification of ω-regular specifications of probabilistic timed automata, a subclass of probabilistic rectangular automata, with the usual dense-time semantics.**

I. Introduction

Systems that are characterized by the interplay between discrete and continuous components are called hybrid systems. Examples of hybrid systems include digital controllers embedded in analog environments, and can be found in a wide variety of contexts, such as manufacturing processes, automotive or aeronautic applications, and domestic appliances. The critical nature of such systems, both from a social and economic viewpoint, has lead to interest in formal techniques to support their correct development. For this purpose, formalisms for hybrid systems, such as hybrid automata [1], have been introduced, along with their associated analysis techniques. A hybrid automaton consists of a finite control graph equipped with a finite set of real-valued variables. As time passes while control remains within a node of the graph, the values of the variables change continuously according to differential equations associated with the node. At certain points in time, control can instantaneously jump from one node to another, and the variables either retain their current value or change discontinuously with the jump. Analysis techniques for hybrid automata generally belong to two categories: *verification* approaches, such as those based on model checking (see, for example, [1], [2]), consist of determining whether the hybrid automaton satisfies some correctness property; *controller-synthesis* approaches involve the computation of a control strategy for (some of) the digital components of the system such that the application of this strategy guides the system in such a way as to guarantee the satisfaction of some correctness property, no matter how the environment behaves (see, for example, [3], [4], [5], [6]).

The basic hybrid automaton formalism does not take into account the relative likelihood of system events. Consider that, for example, in a manufacturing process a physical component may break, or in an aeronautic application an exceptional weather condition may present itself, in both cases with low probability. We may wish to represent such events within our hybrid automaton model, together with the information about their probability of occurrence. This has lead to interest in numerous probabilistic extensions of hybrid automata, where probabilistic information is added in a number of different ways [7], [8], [9], [10], [11], [12], [13], [14]. In this paper, we consider *probabilistic hybrid automata*, as considered in [8], [9], [12], which extend hybrid automata with probabilistic choices over the discrete part of the system. This formalism permits the modeling of events such as faults and message losses, in addition to randomized choices made by the digital components.

A practically relevant subclass of hybrid automata is that of *rectangular automata* [15], in which the continuous dynamics are governed by inclusions of the form $\dot{x} \in I$, where $\dot{x}$ is the first derivative of the variable x and I is an interval. The motivation for such inclusions is that they can over-approximate complex continuous dynamics [16], [17]. However, even simple verification problems for rectangular automata, such as determining whether an error state is reachable, are undecidable, leading to the study of two orthogonal restrictions. In [15], the assumption of *initialization* requires that if a jump between nodes involves a change in a variable's continuous dynamics then the variable is discontinuously reset to a new value by the jump. In [4], a *discrete-time* assumption requires that jumps between nodes can only occur at evenly-spaced points in time. Both assumptions lead to certain verification problems becoming decidable. Initialization and discrete-time have also been used to provide the basis of controller-synthesis algorithms in [5] and [4], respectively. With regard to probabilistic rectangular automata, the initialization assumption has been applied in [8], [9] to obtain approximate probability of the satisfaction of reachability properties. An approach based on approximation is also given in [12], but for probabilistic hybrid automaton models with a form of non-rectangular continuous dynamics incomparable to that of rectangular automata.

In this paper we consider the application of the discrete-time assumption to probabilistic rectangular automata. After introducing some preliminary concepts in Section II and probabilistic rectangular automata in Section III, we give the main result of the paper in Section IV. This result consists of a method for the computation both of the maximum probability with which a controller of the discrete part of the system can satisfy a certain property, no matter how the environment of the system behaves, and involves a reduction to a finite-state $2\frac{1}{2}$-player game. We consider the class of ω-regular properties, modeled here as deterministic Rabin or Streett automata, which allow us to specify a wide variety of safety and liveness requirements. As a side result, we give a method for solving the verification problem, which in this context concerns the computation of the maximum probability with which a property is satisfied. This method involves a reduction to finite-state Markov decision processes. We observe that these results are in contrast to the currently obtained results concerning initialization [8], [9], which do not give exact results even for simple reachability properties. In Section V, we consider a subclass of probabilistic rectangular automata, namely probabilistic timed automata [18], [19], which extend timed automata [20] with discrete probabilistic choice. Given that we have considered ω-regular properties in the context of discrete-time probabilistic rectangular automata, we also show that ω-regular properties of probabilistic timed automata can be verified, even when the usual dense-time semantics is used. In order to rule out unrealistic behavior in which time converges, we consider only executions of the system in which time diverges with probability 1, following [19], [21].

II. Preliminaries

We use $\mathbb{R}$ to denote the set of real numbers, $\mathbb{R}_{\geq 0}$ to denote the set of non-negative real numbers, $\mathbb{N}$ to denote the set of natural numbers, $\mathbb{Z}$ to denote the set of integers, $\mathbb{Q}$ to denote the set of rational numbers, and AP to denote a set of atomic propositions. Given a set Q and a function $\mu : Q \to \mathbb{R}_{\geq 0}$, we define $\mathsf{support}(\mu) = \{q \in Q \mid \mu(q) > 0\}$. A (discrete) probability *distribution* over a countable set Q is a function $\mu : Q \to [0,1] \cap \mathbb{Q}$ such that $\sum_{q \in Q} \mu(q) = 1$. Let $\mathsf{Dist}(Q)$ be the set of distributions over Q. If Q is an uncountable set, we define $\mathsf{Dist}(Q)$ to be the set of functions $\mu : Q \to [0,1]$, such that $\mathsf{support}(\mu)$ is a countable set and μ restricted to $\mathsf{support}(\mu)$ is a (discrete) probability distribution. Occasionally we use notation $\{q_1 \mapsto \lambda_1, ..., q_n \mapsto \lambda_n\}$ to denote a distribution μ for which $\mu(q_1) = \lambda_1$, ..., $\mu(q_n) = \lambda_n$.

A *probabilistic game graph* (or $2\frac{1}{2}$*-player game graph*) $\mathsf{G} = (S, \to, Lab)$ comprises the following components: a (possibly uncountable) set of *states* S; a (possibly uncountable) *probabilistic, game-based transition relation* $\to \subseteq S \times 2^{\mathsf{Dist}(S)} \setminus \emptyset$; and a *labeling function* $Lab : S \to 2^{AP}$. The transitions from state to state of a $2\frac{1}{2}$-player game are performed in three steps: given that the current state is s, the first step concerns a nondeterministic selection by player 1 of $(s, \Lambda) \in \to$; the second step comprises a nondeterministic selection by player 2 of some $\mu \in \Lambda$; the third step comprises a probabilistic choice, made according to the distribution μ, as to which state to make the transition to (that is, we then make a transition to a state $s' \in S$ with probability $\mu(s')$). Note that, in this paper, we assume that turns of the game are played in a cyclic manner, where each cycle consists first of the turn of player 1, then that of player 2, followed by that of the probabilistic player. This suffices for our purposes, but is in contrast to the usual presentation of $2\frac{1}{2}$-player games (see, for example, [22]), in which the order of the turns of the game does not follow a fixed cycle. A $2\frac{1}{2}$-player game is *total* if, for each state $s \in S$, there exists at least one transition $(s, \_) \in \to$. We generally consider total $2\frac{1}{2}$-player games in this paper. Occasionally we omit the labeling function Lab for $2\frac{1}{2}$-player games.

An *infinite path* of a $2\frac{1}{2}$-player game G is an infinite sequence $r = s_0\Lambda_0\mu_0 s_1\Lambda_1\mu_1 \cdots$ such that $(s_i, \Lambda_i) \in \to$, $\mu_i \in \Lambda_i$ and $\mu_i(s_{i+1}) > 0$ for each $i \in \mathbb{N}$. Similarly, a *finite path* of G is a finite sequence $r = s_0\Lambda_0\mu_0 s_1\Lambda_1\mu_1 \cdots \Lambda_{n-1}\mu_{n-1}s_n$ such that $(s_i, \Lambda_i) \in \to$, $\mu_i \in \Lambda_i$ and $\mu_i(s_{i+1}) > 0$ for each $i < n$. If r is finite, the length of r, denoted by $|r|$, is equal to the number of transitions (subsequences of the form $s\Lambda\mu$) along r. If r is infinite, we let $|r| = \infty$. We use $Path^{\mathsf{G}}_{ful}$ to denote the set of infinite paths of G, and $Path^{\mathsf{G}}_{fin}$ to denote the set of finite paths of G. When clear from the context we omit the superscript G. If r is a finite path, we denote by $last(r)$ the last state of r. For any path r and $i \leq |r|$, let $r(i) = s_i$ be the $(i+1)$th state along r, and let $step(r, i) = \mu_i$ be the $(i+1)$th distribution taken along r. Let $Path^{\mathsf{G}}_{ful}(s)$ and $Path^{\mathsf{G}}_{fin}(s)$ refer to the sets of infinite and finite paths of G, respectively, commencing in state $s \in S$.

Let $\mathsf{G} = (S, \to, Lab)$ be a $2\frac{1}{2}$-player game. A *player 1 strategy* on G is a function σ mapping every finite path $r \in Path_{fin}$ to a transition $(last(r), \Lambda) \in \to$. Similarly, a *player 2 strategy* on G is a function π mapping every sequence $r \cdot \Lambda$, such that $r \in Path_{fin}$ and $(last(r), \Lambda) \in \to$, to a distribution $\mu \in \Lambda$. We write Σ_{G} and Π_{G} for the set of strategies of player 1 and player 2, respectively, on G. A pair $(\sigma, \pi) \in \Sigma_{\mathsf{G}} \times \Pi_{\mathsf{G}}$ is called a *strategy profile*. For any strategy profile (σ, π), let $Path^{\sigma,\pi}_{ful}$ and $Path^{\sigma,\pi}_{fin}$ denote the sets of infinite and finite paths, respectively, resulting from the choices of (σ, π). For a state $s \in S$, let $Path^{\sigma,\pi}_{ful}(s) = Path^{\sigma,\pi}_{ful} \cap Path_{ful}(s)$ and $Path^{\sigma,\pi}_{fin}(s) = Path^{\sigma,\pi}_{fin} \cap Path_{fin}(s)$. Given a strategy profile $(\sigma, \pi) \in \Sigma_{\mathsf{G}} \times \Pi_{\mathsf{G}}$ and a state $s \in S$, we define the probability measure $Prob^{\sigma,\pi}_{s}$ over $Path^{\sigma,\pi}_{ful}(s)$ in the standard way [23]. Note that, following the usual terminology for games on graphs, we generally consider pure strategies (that is, strategies that do not make randomized choices), the choices of which may depend on the history of the system. The cases in which randomized strategies (which, for player 1, map from finite paths r to $\mathsf{Dist}(\to)$ and, for player 2, map from $r \cdot \Lambda$ to $\mathsf{Dist}(\Lambda)$) are considered will be signalled in the text.

Given an infinite path $r = s_0\Lambda_0\mu_0 s_1\Lambda_1\mu_1 \cdots$ of a $2\frac{1}{2}$-player game $\mathsf{G} = (S, \to, Lab)$, the *trace of* r, denoted by $\mathsf{trace}(r)$, is defined to be the infinite sequence $Lab(s_0)Lab(s_1)\cdots$. Let $\mathsf{Trace}(\mathsf{G})$ be the set of all traces of G (i.e., $\mathsf{Trace}(\mathsf{G}) = \{\mathsf{trace}(r) \in (2^{AP})^{\omega} \mid r \in Path^{\mathsf{G}}_{ful}\}$). An *objective* φ for G is a set of traces of G (i.e., $\varphi \subseteq \mathsf{Trace}(\mathsf{G})$). In this paper, we will consider the class of ω-regular objectives. Given the ω-regular objective φ, a state $s \in S$ and a strategy profile (σ, π), we note that $\{r \in \mathit{Path}_{\mathit{ful}}^{\sigma,\pi}(s) \mid \mathsf{trace}(r) \in \varphi\}$ is measurable, and for simplicity we write $\mathit{Prob}_s^{\sigma,\pi}(\varphi)$ instead of $\mathit{Prob}_s^{\sigma,\pi}(\{r \in \mathit{Path}_{\mathit{ful}}^{\sigma,\pi}(s) \mid \mathsf{trace}(r) \in \varphi\})$. The *value function* (for player 1 and the property φ) is defined as the function $\mathit{Val}^{\mathsf{G}}(\varphi)$ such that, for each state $s \in S$:

$$\mathit{Val}^{\mathsf{G}}(\varphi)(s) = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \mathit{Prob}_s^{\sigma,\pi}(\varphi) \ .$$

A *Markov decision process* (MDP) is a $2\frac{1}{2}$-player game $\mathsf{M} = (S, \rightarrow, \mathit{Lab})$ for which $|\Lambda| = 1$ for each $(s, \Lambda) \in \rightarrow$. Usually we write the transition relation $\rightarrow$ of an MDP as $\rightarrow \subseteq S \times \mathsf{Dist}(S)$. In contrast to $2\frac{1}{2}$-player games, the transitions from state to state of an MDP are performed in two steps: given that the current state is s, the first step concerns a nondeterministic selection of $(s, \mu) \in \rightarrow$; the second step comprises a probabilistic choice made according to the distribution μ. A (finite or infinite) path of an MDP is defined as for a $2\frac{1}{2}$-player game, with only minor notational differences (for example, an infinite path of an MDP is denoted by $s_0 \mu_0 s_1 \mu_1 \cdots$, where $(s_i, \mu_i) \in \rightarrow$ and $\mu_i(s_{i+1}) > 0$ for each $i \in \mathbb{N}$). In the case of MDPs, player 2 has a trivial choice over a single element, and hence has only one strategy (i.e., $|\Pi| = 1$): therefore we use the term strategy to refer both to player 1 strategies and strategy profiles. Similarly, we omit the notation referring to the player 2 strategy, and write, for example, $\mathit{Path}_{\mathit{ful}}^{\sigma}(s)$ and Prob_s^{σ}. The value function for the MDP M is defined as $\mathit{Val}^{\mathsf{M}}(\varphi)(s) = \sup_{\sigma \in \Sigma} \mathit{Prob}_s^{\sigma}(\varphi)$ for each state $s \in S$.

A sub-MDP $(S', \rightarrow', \mathit{Lab}|_{S'})$ of M is a MDP such that $S' \subseteq S$, $\rightarrow' \subseteq \rightarrow$, and $\mathit{Lab}|_{S'}$ is equal to Lab restricted to S'. Let $T \subseteq S$. The *sub-MDP of* M *induced by* T is the sub-MDP $(T, \rightarrow|_T, \mathit{Lab}|_T)$ of M, where $\rightarrow|_T = \{(s, \nu) \in \rightarrow \mid s \in T \wedge \mathsf{support}(\nu) \subseteq T\}$.

The graph of an MDP $(S, \rightarrow)$ is the pair (S, Edges) where $(s, s') \in \mathit{Edges}$ if and only if there exists $(s, \mu) \in \rightarrow$ such that $s' \in \mathsf{support}(\mu)$. An *end component* (EC) of an MDP M is a sub-MDP $(A, B) \in 2^S \times 2^{\rightarrow}$ such that (1) if $(s, \mu) \in B$, then $s \in A$ and $\mathsf{support}(\mu) \subseteq A$, and (2) the graph of (A, B) is strongly connected [24]. An EC (A, B) of M is *maximal* if there does not exist any EC (A', B') of M such that $(A, B) \neq (A', B')$, $A \subseteq A'$ and $B \subseteq B'$.

A *probabilistic timed labeled transition system* (PTLTS) $\mathsf{T} = (S, \mathit{Events}, \Rightarrow, \mathit{Lab})$ comprises the following components: a (possibly uncountable) set of states S; a (possibly uncountable) set of events *Events*; a (possibly uncountable) timed probabilistic, nondeterministic transition relation $\Rightarrow \subseteq S \times (\mathbb{R}_{\geq 0} \cup \mathit{Events}) \times \mathsf{Dist}(S)$; and a labeling function $\mathit{Lab} : S \rightarrow 2^{AP}$. The transitions from state to state of a PTLTS contain information about time duration of the event corresponding to the transition. The notions of totality and paths of PTLTSs are adapted in a straightforward way from $2\frac{1}{2}$-player games: for example, an infinite path of a PTLTS is denoted by $r = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \cdots$ where $a_i \in \mathbb{R}_{\geq 0} \cup \mathit{Events}$ for each $i \in \mathbb{N}$. We can interpret PTLTSs in two ways, depending on the type of analysis. The *MDP interpretation of* T is an MDP $\mathsf{M}(\mathsf{T}) = (S, \rightarrow, \mathit{Lab})$ where $\rightarrow$ is the smallest set such that $(s, a, \mu) \in \Rightarrow$ implies $(s, \mu) \in \rightarrow$. The MDP interpretation of a PTLTS is used for verification problems, in which all of the nondeterministic choices of which transitions to take are under the control of a single player. Next, we introduce a $2\frac{1}{2}$-player game interpretation of a PTLTS, which is used for control problems: in this interpretation, the control problems involve player 1 (the controller) choosing which event should be taken and player 2 (the environment) choosing the exact transition that is then taken, provided that it corresponds to the event chosen by player 1; alternatively, player 1 can choose that time should elapse, in which case player 2 chooses the exact time duration and time-elapse transition. Formally, the $2\frac{1}{2}$*-player game interpretation of* T is a $2\frac{1}{2}$-player game $\mathsf{G}(\mathsf{T}) = (S, \rightarrow, \mathit{Lab})$ where $\rightarrow$ is the smallest set such that, for each $s \in S$:

- *(Time transitions)* $(s, \{\mu \mid (s, d, \mu) \in \Rightarrow \text{ and } d \in \mathbb{R}_{\geq 0}\}) \in \rightarrow$ if there exists $d' \in \mathbb{R}_{\geq 0}$ such that $(s, d', \_) \in \Rightarrow$;
- *(Event transitions)* $(s, \{\mu \mid (s, e, \mu) \in \Rightarrow\}) \in \rightarrow$ for each $e \in \mathit{Events}$ such that there exists $(s, e, \_) \in \Rightarrow$.

In subsequent sections, we usually simplify the notation for choices of the player 1 strategies: for a finite path $r \in \mathit{Path}_{\mathit{fin}}$, we write $\sigma(r) = \mathit{time}$ if $\sigma(r)$ corresponds to a transition $(\mathit{last}(r), \Lambda)$ obtained by the time transition rule, and we write $\sigma(r) = e$ if $\sigma(r)$ corresponds to a transition $(\mathit{last}(r), \Lambda)$ obtained by the event transition rule for event $e \in \mathit{Events}$.

III. Probabilistic Rectangular Automata

A. Definition of probabilistic rectangular automata

Let $\mathcal{X}$ be a finite set of real-valued variables. A *valuation* $v : \mathcal{X} \rightarrow \mathbb{R}$ is a function that assigns a real-value to each variable of $\mathcal{X}$. A *rectangular inequality* over $\mathcal{X}$ is defined as a formula of the form $x \sim c$, where $x, y \in \mathcal{X}$, $\sim \in \{<, \leq, >, \geq\}$, and $c \in \mathbb{Z}$. A *rectangular constraint* over $\mathcal{X}$ is a conjunction of rectangular inequalities over $\mathcal{X}$. The set of all rectangular constraints over $\mathcal{X}$ is denoted by $\mathit{Rect}(\mathcal{X})$. Given a rectangular constraint Φ and valuation v, we say that v *satisfies* Φ if Φ is true after substituting $v(x)$ in place of x for all $x \in \mathcal{X}$. The set of valuations that satisfy Φ is denoted by $[\![\Phi]\!]$. Let $k \in \mathbb{N}$ be a non-negative integer. Then the rectangular constraint Φ is *k-definable* if $|c| \leq k$ for every conjunct $x \sim c$ of Φ.

A *probabilistic rectangular automaton* (PRA) $\mathcal{R} = (L, \mathcal{X}, \mathit{Events}, \mathit{inv}, \mathit{flow}, \mathit{prob}, \mathcal{L})$ consists of the following components:

- a finite set L of *locations*;
- a finite set $\mathcal{X}$ of variables;
- a finite set *Events* of events;
- a function $\mathit{inv} : L \rightarrow \mathit{Rect}(\mathcal{X})$ associating an *invariant condition* with each location;
- a function $\mathit{flow} : L \rightarrow \mathit{Rect}(\dot{\mathcal{X}})$ associating an *flow condition* with each location, where $\dot{\mathcal{X}} = \{\dot{x} \mid x \in \mathcal{X}\}$ is the set of first derivatives of variables in $\mathcal{X}$;
- a finite set $\mathit{prob} \subseteq L \times \mathit{Rect}(\mathcal{X}) \times \mathit{Events} \times \mathsf{Dist}(\mathit{Upd}(\mathcal{X}) \times L)$ of *probabilistic edges*, where $\mathit{Upd}(\mathcal{X})$ is the set of formulae of the form $\phi' \wedge \bigwedge_{x \in X}(x' = x)$,

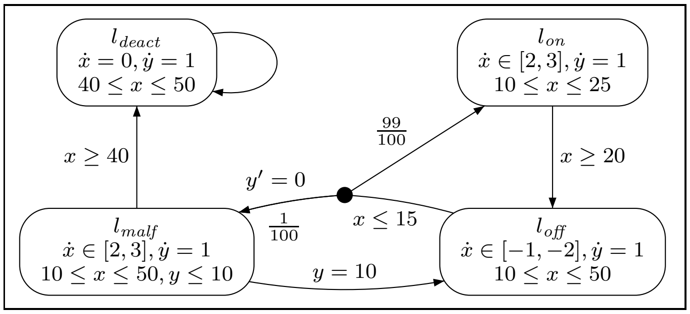

Fig. 1. A PRA modeling a faulty thermostat

for $X \subseteq \mathcal{X}$, $\phi' \in \mathit{Rect}(\mathcal{X}' \setminus X')$, where a primed variable x' refers to the value of x after traversing the probabilistic edge, and where $\mathcal{X}' = \{x' \mid x \in \mathcal{X}\}$ and $X' = \{x' \mid x \in X\}$;

- a *labeling function* $\mathcal{L} : L \rightarrow 2^{AP}$.

A probabilistic edge $(l, g, e, p) \in \mathit{prob}$ comprises (1) a source location l, (2) a rectangular constraint g, called a *guard*, (3) an event e, and (4) a probability distribution p that assigns probability to pairs of the form (ϑ, l'), where $\vartheta \in \mathit{Upd}(\mathcal{X})$ is a constraint describing the manner in which variables are reset and $l' \in L$ is a location. A constraint $\vartheta \in \mathit{Upd}(\mathcal{X})$ is said to be satisfied by a pair (v, v') of valuations if ϑ is true after substituting $v(x)$ for x and $v'(x)$ for x'.

The behavior of a probabilistic rectangular automaton takes a similar form to that of a rectangular automaton [15]. If the PRA is currently in location l, time can advance as long as the current values of the variables in $\mathcal{X}$ satisfy the invariant condition $inv(l)$. As time passes, the value of the variables in $\mathcal{X}$ change according to a differential trajectory satisfying the flow condition $\mathit{flow}(l)$. If the current values of the variables satisfy the guard g of a probabilistic edge (l, g, a, p), then the probabilistic edge can be taken, which involves a probabilistic choice according to the distribution p: if the pair (ϑ, l') is chosen, then the PRA goes to location l', resetting the variables according to the constraint ϑ. More precisely, if ϑ is the constraint $\phi' \wedge \bigwedge_{x \in X}(x' = x)$, then variables in X retain the same value, whereas variables in $\mathcal{X} \setminus X$ are reset to a value satisfying the rectangular constraint ϕ'. The following choices made by the PRA are nondeterministic: the amount of time to let advance in the current location l; the differential trajectory used to describe the change of the variables as time passes; the probabilistic edge taken (after time has finished elapsing, and provided that the guard of the probabilistic edge is satisfied by the current variable valuation); and, finally, the values to which the variables are reset (only for those variables that do not retain their previous value). Instead, the only probabilistic choice featured in the model concerns the choice of pair (ϑ, l') once a probabilistic edge has been chosen.

In Figure 1 we give an example of a PRA modeling a faulty thermostat. We use a number of the usual conventions for illustrating hybrid automata, such as flow and invariant conditions shown within locations. The temperature is represented by the variable x, and variable y is used to measure elapsed time. The system passes from the heater being on (location l_{on}) to being off (location l_{off}) when the temperature is between 20 and 25. The system passes from the heater being off (location l_{off}) to being on (location l_{on} or location l_{malf}) when the temperature is between 10 and 15. The location l_{on} corresponds to non-faulty behavior, and is reached with probability $\frac{99}{100}$ from l_{off}. Instead, the location l_{malf} corresponds to the heater being on in the presence of a fault in the temperature sensor, and is reached with probability $\frac{1}{100}$. The sensor fault means that the temperature can increase to a higher level than in l_{on}. After a malfunction, either the system is deactivated if the temperature reaches an excessive level (location l_{deact}), or the system times-out exactly 10 time units after the location l_{malf} was entered. All probabilistic edges of the PRA correspond to reaching a certain location with probability 1, apart from the probabilistic edge from l_{off}.

We now introduce formally the semantics of PRA in terms of PTLTSs. The *dense-time semantics of the PRA* $\mathcal{R} = (L, \mathcal{X}, \mathit{Events}, inv, \mathit{flow}, \mathit{prob}, \mathcal{L})$ is the PTLTS $\mathsf{T}^{\mathcal{R}}_{dense} = (S, \mathit{Events}, \Rightarrow, \mathit{Lab})$ defined in the following way. The set of states of $\mathsf{T}^{\mathcal{R}}_{dense}$ is defined as $S = \{(l, v) \in L \times \mathbb{R}^{\mathcal{X}} \mid v \text{ satisfies } inv(l)\}$. To define the transition relation $\Rightarrow$, we first define a transition relation for each time duration and event.

- *(Flows)* Let $d \in \mathbb{R}_{\geq 0}$. Then $\xRightarrow{d} \subseteq S \times \mathsf{Dist}(S)$ is the smallest set such that $((l, v), d, \{(l', v') \mapsto 1\}) \in \xRightarrow{d}$ implies that (1) $l = l'$, and (2) there exists a differentiable function $f : [0, d] \rightarrow [\![inv(l)]\!]$ such that $f(0) = v$, $f(d) = v'$ and $\dot{f}(\varepsilon) \in [\![\mathit{flow}(l)]\!]$ for all reals $\varepsilon \in (0, d)$, where $\dot{f}$ is the first derivative of f.
- *(Jumps)* Let $e \in \mathit{Events}$. Then $\xRightarrow{e} \subseteq S \times \mathsf{Dist}(S)$ is the smallest set of transitions such that $((l, v), e, \mu) \in \xRightarrow{e}$ implies that there exists a probabilistic edge $(l, g, e, p) \in \mathit{prob}$ satisfying the following conditions:
  1) v satisfies g,
  2) given $\mathsf{support}(p) = \{(\vartheta_1, l'_1), ..., (\vartheta_n, l'_n)\}$, there exists a vector $[v'_1, ..., v'_n]$ of valuations over $\mathcal{X}$ such that,
     a) (v, v'_i) satisfies ϑ_i for each $1 \leq i \leq n$, and
     b) for each $(l', v') \in S$:

$$\mu(l', v') = \sum_{1 \leq i \leq n \text{ s.t. } l' = l'_i \text{ and } v' = v'_i} p(\vartheta_i, l'_i) \ .$$

Then we define $\Rightarrow$ as the transition relation $(\bigcup_{d \in \mathbb{R}_{\geq 0}} \xRightarrow{d}) \cup (\bigcup_{e \in \mathit{Events}} \xRightarrow{e})$. Finally, the labeling function Lab is such that $\mathit{Lab}(l, v) = \mathcal{L}(l)$ for each state $(l, v) \in S$.

We note that the summation in the definition of jump transitions is necessary for the case in which the same state can be obtained by more than one element (ϑ, l) in the support set of the distribution of a probabilistic edge. We say that (l, v) is a state of $\mathcal{R}$ if (l, v) is a state of $\mathsf{T}^{\mathcal{R}}_{dense}$.

In addition to the usual dense-time semantics, we also consider a discrete-time semantics for PRA in which only flow transitions of duration 1 are included, following the precedent of [4] for rectangular automata. Formally, the *discrete-time* *semantics of the PRA* $\mathcal{R} = (L, \mathcal{X}, \mathit{Events}, \mathit{inv}, \mathit{flow}, \mathit{prob}, \mathcal{L})$ is the PTLTS $\mathsf{T}^{\mathcal{R}}_{\mathit{discrete}} = (S, \mathit{Events}, \Rightarrow, \mathit{Lab})$ defined as for the dense-time semantics except for $\Rightarrow$, which is defined as $\overset{1}{\Rightarrow} \cup (\bigcup_{e \in \mathit{Events}} \overset{e}{\Rightarrow})$.

We restrict our attention to PRA $\mathcal{R}$ with a semantic PTLTS $\mathsf{T}^{\mathcal{R}}_{\mathit{discrete}}$ that is total. This can be guaranteed by a syntactic condition similar to that which has been presented for PTA in [25], which guarantees that a guard of at least one probabilistic edge is enabled when the invariant of the current location is not satisfied by letting time elapse.

Let $\mathcal{R}$ be a PRA with the set L of locations and the set $\mathcal{X}$ of variables. We say that $\mathcal{R}$ is *k-definable* if every rectangular constraint in the definition of $\mathcal{R}$ is k-definable. Given $x \in \mathcal{X}$ and $\Phi \in \mathit{Rect}(\mathcal{X})$, we denote by $[\![\Phi]\!]_x$ the interval $\{v(x) \in \mathbb{R} \mid v \in [\![\Phi]\!]\}$. The variable $x \in \mathcal{X}$ is *nondecreasing* if both $[\![\mathit{inv}(l)]\!]_x \subseteq \mathbb{R}_{\geq 0}$ and $[\![\mathit{flow}(l)]\!]_x \subseteq \mathbb{R}_{\geq 0}$ for all locations $l \in L$. The variable $x \in \mathcal{X}$ is *bounded* if $[\![\mathit{inv}(l)]\!]_x$ is a bounded set, for all locations $l \in L$. The PRA $\mathcal{R}$ has *nondecreasing or bounded* variables if all variables in $\mathcal{X}$ are either nondecreasing or bounded. In the PRA of Figure 1, the variable x is bounded, whereas the variable y is nondecreasing.

A *probabilistic timed automaton (PTA)* [18], [19] is a PRA for which:

- $\mathit{flow}(l) = \bigwedge_{x \in \mathcal{X}} (\dot{x} = 1)$ for each location $l \in L$.
- for each $(l, g, e, p) \in \mathit{prob}$ and each $(\vartheta, l') \in \mathsf{support}(p)$, the variable-update constraint ϑ is of the form $\bigwedge_{x \in \mathcal{X} \setminus X} (x' = 0) \wedge \bigwedge_{x \in X} (x' = x)$ for some $X \subseteq \mathcal{X}$.

The variables of a PTA are referred to as *clocks*.

B. Verification and control problems

Both the dense-time and discrete-time semantics of a PRA can have an MDP interpretation or a $2\frac{1}{2}$-player game interpretation. Let φ be an ω-regular objective. For $\mathsf{Int} \in \{\mathsf{M}, \mathsf{G}\}$ and $\star \in \{\mathit{dense}, \mathit{discrete}\}$, we use $\mathit{Val}^{\mathsf{Int}(\mathcal{R})}_{\star}(\varphi)$ to denote $\mathit{Val}^{\mathsf{Int}(\mathsf{T}_{\star}(\mathcal{R}))}(\varphi)$. The *dense-time verification value function of $\mathcal{R}$ and φ* is the value function $\mathit{Val}^{\mathsf{M}(\mathcal{R})}_{\mathit{dense}}(\varphi)$. The corresponding discrete-time versions of the verification and control value functions are $\mathit{Val}^{\mathsf{M}(\mathcal{R})}_{\mathit{discrete}}(\varphi)$ and $\mathit{Val}^{\mathsf{G}(\mathcal{R})}_{\mathit{discrete}}(\varphi)$, respectively.

Let s be a state of $\mathsf{T}^{\mathcal{R}}_{\mathit{dense}}$ and $\lambda > 0$ be a rational. For $\star \in \{\mathit{dense}, \mathit{discrete}\}$, the *$\star$-time verification problem for $\mathcal{R}$, s and φ* consists of deciding whether $\mathit{Val}^{\mathsf{M}(\mathcal{R})}_{\star}(\varphi)(s) \geq \lambda$. The *discrete-time control problem for $\mathcal{R}$, s and φ* consists of deciding whether $\mathit{Val}^{\mathsf{G}(\mathcal{R})}_{\mathit{discrete}}(\varphi)(s) \geq \lambda$. The *discrete-time controller synthesis problem for $\mathcal{R}$, s and φ* consists of the construction of a strategy for player 1 that witnesses $\mathit{Val}^{\mathsf{G}(\mathcal{R})}_{\mathit{discrete}}(\varphi)(s) \geq \lambda$, for the case in which the corresponding control decision problem returns a positive answer.

Previous work in the field of PRA has mainly considered verification problems for the subclass of PTA, and with respect to properties expressed in the probabilistic temporal logic PTCTL [19]. The most important subroutine of the PTCTL model-checking algorithm of [19] concerns the computation of the dense-time verification value function of the PTA with regard to particular examples of ω-regular objectives, namely eventually and always objectives, written as $\Diamond \mathsf{a}$ and $\Box \mathsf{a}$, respectively, in Linear Temporal Logic (LTL) notation (see [26]), where a is an atomic proposition. These results are summarized in the following theorem[1].

Theorem 1 ([19], [27]): The dense-time verification problem for PTA with eventually or always objectives is EXPTIME-complete.

We also recall that [28] considers a dense-time controller synthesis problem concerning the computation of controllers of PTA that optimize the expected time to reach a state set.

IV. Discrete-Time Control for Probabilistic Rectangular Automata

In this section, we consider the discrete-time verification, control and controller-synthesis problems for PRA with ω-regular objectives under the discrete-time semantics. The key tool that we use to obtain solutions to these problems is that of probabilistic bisimulation [29], [30], in the same way that bisimulation was used to obtain algorithms for non-probabilistic rectangular automata in [4]. After describing how probabilistic bisimulation can be used to obtain a finite-state $2\frac{1}{2}$-player game or MDP from a PRA, we then consider the resulting computation of value functions for ω-regular properties modeled as deterministic Rabin or Streett automata.

A. Probabilistic bisimulation and finite quotient

Consider the set $\mathcal{X}$ of variables. Let $\approx^k \subseteq (\mathbb{R}^{\mathcal{X}})^2$ be the equivalence relation on valuations defined in the following way: $v \approx^k w$ if and only if either (1) $\lfloor v(x) \rfloor = \lfloor w(x) \rfloor$ and $\lceil v(x) \rceil = \lceil w(x) \rceil$, (2) $v(x), w(x) > k$, or (3) $v(x), w(x) < -k$, for all $x \in \mathcal{X}$. We note that every equivalence class of $\approx^k$ corresponds to the set of valuations that satisfy some k-definable rectangular constraint. Vice versa, every k-definable rectangular constraint defines a union of $\approx^k$-equivalence classes.

For a PRA $\mathcal{R}$ whose (dense- or discrete-time) semantics has the state set S, let $\cong^k_{\mathcal{R}} \subseteq (S)^2$ be the equivalence relation defined in the following way: $(l, v) \cong^k_{\mathcal{R}} (m, w)$ if and only if $l = m$ and $v \approx^k w$.

We now apply the notion of probabilistic bisimulation [29], [30] to PTLTSs. Let $\mathsf{T} = (S, \mathit{Events}, \Rightarrow, \mathit{Lab})$ be a PTLTS. For any two distributions $\mu, \nu \in \mathsf{Dist}(S)$ and for any equivalence relation $\equiv \subseteq (S)^2$, we denote by $\mu \equiv \nu$ the condition that, for each equivalence class C of $\equiv$, the equality $\sum_{s \in C} \mu(s) = \sum_{s \in C} \nu(s)$ holds. A *probabilistic bisimulation* on T is an equivalence relation $\simeq \subseteq (S)^2$ such that $s \simeq t$ implies:

1) $\mathit{Lab}(s) = \mathit{Lab}(t)$,
2) if $(s, a, \mu) \in \Rightarrow$, then there exists $(t, a, \nu) \in \Rightarrow$ such that $\mu \simeq \nu$.

Theorem 2: Let $\mathcal{R}$ be a k-definable PRA that has nondecreasing or bounded variables. Then $\cong^k_{\mathcal{R}}$ is a probabilistic bisimulation of the discrete-time semantics $\mathsf{T}^{\mathcal{R}}_{\mathit{discrete}}$ of $\mathcal{R}$.

[1] We assume that the size of the PTA is described in the usual way: constants used in the conditions of the PTA are encoded in binary, and probabilities are expressed as a ratio between two natural numbers, each written in binary.

Theorem 2 allows us to obtain the following proposition, which states that $\cong^k_{\mathcal{R}}$-equivalent states have the same values, for any ω-regular objective.

Proposition 1: Let φ be an ω-regular objective, and let s, t be states of the discrete-time semantics $\mathsf{T}^{\mathcal{R}}_{discrete}$ of the k-definable PRA $\mathcal{R}$ with nondecreasing or bounded variables such that $s \cong^k_{\mathcal{R}} t$. Then:

$$\begin{aligned} Val^{\mathsf{M}(\mathcal{R})}_{discrete}(\varphi)(s) &= Val^{\mathsf{M}(\mathcal{R})}_{discrete}(\varphi)(t) \\ Val^{\mathsf{G}(\mathcal{R})}_{discrete}(\varphi)(s) &= Val^{\mathsf{G}(\mathcal{R})}_{discrete}(\varphi)(t)\ . \end{aligned}$$

Let $\Lambda, \Lambda' \in 2^{\mathsf{Dist}}(S) \setminus \emptyset$. We write $\Lambda \cong^k_{\mathcal{R}} \Lambda'$ if (1) for each $\mu \in \Lambda$, there exists $\nu \in \Lambda'$ such that $\mu \cong^k_{\mathcal{R}} \nu$, and, conversely, (2) for each $\nu \in \Lambda'$, there exists $\mu \in \Lambda$ such that $\mu \cong^k_{\mathcal{R}} \nu$. Let $r = s_0\Lambda_0\mu_0 \cdots s_{n-1}\Lambda_{n-1}\mu_{n-1}s_n$ and $r' = t_0\Lambda'_0\nu_0 \cdots t_{n-1}\Lambda'_{n-1}\nu_{n-1}t_n$ be paths such that (1) $s_i \cong^k_{\mathcal{R}} t_i$ for all $i \leq n$, and (2) $\Lambda_i \cong^k_{\mathcal{R}} \Lambda'_i$ and $\mu_i \cong^k_{\mathcal{R}} \nu_i$ for all $i < n$. Then a player 1 strategy $\sigma \in \Sigma$ is $\cong^k_{\mathcal{R}}$*-oblivious* if $\sigma(r) = \sigma(r')$. Similarly, a player 2 strategy $\pi \in \Pi$ is $\cong^k_{\mathcal{R}}$-oblivious if $\pi(r \cdot \Lambda) \cong^k_{\mathcal{R}} \pi(r' \cdot \Lambda')$ where $\Lambda \cong^k_{\mathcal{R}} \Lambda'$. Let Σ^{obl} and Π^{obl} be the sets of $\cong^k_{\mathcal{R}}$-oblivious strategies of player 1 and player 2, respectively. The next proposition shows that $\cong^k_{\mathcal{R}}$-oblivious strategies suffice for determining the value function.

Proposition 2: Let $\mathcal{R}$ be a k-definable PRA $\mathcal{R}$ with nondecreasing or bounded variables, and let s be a state of $\mathsf{T}^{\mathcal{R}}_{discrete}$. Then:

$$\sup_{\sigma\in\Sigma}\inf_{\pi\in\Pi} Prob^{\sigma,\pi}_s(\varphi) = \sup_{\sigma'\in\Sigma^{\mathsf{obl}}}\inf_{\pi'\in\Pi^{\mathsf{obl}}} Prob^{\sigma',\pi'}_s(\varphi)\ .$$

Following [4], we observe that the number of equivalence classes of $\cong^k_{\mathcal{R}}$ equals $|L| \cdot (4k+3)^{|\mathcal{X}|}$. This, together with Proposition 2 suggests using probabilistic bisimulation to define a finite-state PTLTS on which verification, control and controller synthesis problems may be solved.[2] The $\cong^k_{\mathcal{R}}$*-quotient* of the k-definable PRA $\mathcal{R}$ with nondecreasing or bounded variables $\mathcal{R} = (L, \mathcal{X}, Events, inv, flow, prob, \mathcal{L})$ is the PTLTS $\mathfrak{P}^k(\mathcal{R}) = (\mathfrak{C}, Events, \rightarrowtail, \mathfrak{Lab})$ defined in the following way.

- $\mathfrak{C}$ is the set of equivalence classes of $\cong^k_{\mathcal{R}}$.
- $\rightarrowtail$ is the set $\overset{1}{\rightarrowtail} \cup (\bigcup_{e\in Events} \overset{e}{\rightarrowtail})$ defined as follows. First, $\overset{1}{\rightarrowtail}$ is the smallest set of transitions such that, for each $C, C' \in \mathfrak{C}$ for which there exist $s \in C$, $s' \in C'$ with $(s, 1, \{s' \mapsto 1\}) \in \Rightarrow$, we have $(C, 1, \{C' \mapsto 1\}) \in \overset{1}{\rightarrowtail}$. Second, for each $e \in Events$, $\overset{e}{\rightarrowtail}$ is the smallest set of transitions such that, for each $C \in \mathfrak{C}$ and each $(s, e, \mu) \in \Rightarrow$, we have $(C, e, \nu) \in \rightarrowtail$, where $\nu(C') = \sum_{s'\in C'} \mu(s')$ for each $C' \in \mathfrak{C}$.
- $\mathfrak{Lab}$ is defined by $\mathfrak{Lab}(C) = Lab(s)$, for each $C \in \mathfrak{C}$ and an arbitrary $s \in C$.

In the following, given PTLTSs $\mathsf{T} = (S, Events, \Rightarrow, Lab)$ and $\mathsf{T}' = (S', Events', \Rightarrow', Lab')$, let the *union PTLTS* be defined by $\mathsf{T} \uplus \mathsf{T}' = (S \uplus S', Events \cup Events', \Rightarrow \uplus \Rightarrow', Lab'')$, where $Lab''(s) = Lab(s)$ if $s \in S$ and $Lab''(s) = Lab'(s)$ if $s \in S'$.

Proposition 3: Let $\mathcal{R}$ be a k-definable PRA $\mathcal{R}$ with nondecreasing or bounded variables, let C be an equivalence class of $\cong^k_{\mathcal{R}}$, and let s be a state of $\mathsf{T}^{\mathcal{R}}_{discrete}$ such that $s \in C$. Then s and C are probabilistically bisimilar in $\mathsf{T}^{\mathcal{R}}_{discrete} \uplus \mathfrak{P}^k(\mathcal{R})$, and hence:

$$\begin{aligned} Val^{\mathsf{M}(\mathcal{R})}_{discrete}(\varphi)(s) &= Val^{\mathsf{M}(\mathfrak{P}^k(\mathcal{R}))}(\varphi)(C) \\ Val^{\mathsf{G}(\mathcal{R})}_{discrete}(\varphi)(s) &= Val^{\mathsf{G}(\mathfrak{P}^k(\mathcal{R}))}(\varphi)(C)\ . \end{aligned}$$

Proposition 3 suggests the following approach for computing the value functions $Val^{\mathsf{M}(\mathcal{R})}_{discrete}(\varphi)$ and $Val^{\mathsf{G}(\mathcal{R})}_{discrete}(\varphi)$: construct the $\cong^k_{\mathcal{R}}$-quotient of the PRA: then compute the value functions $Val^{\mathsf{M}(\mathfrak{P}^k(\mathcal{R}))}(\varphi)$ and $Val^{\mathsf{G}(\mathfrak{P}^k(\mathcal{R}))}(\varphi)$ using methods for the computation of value functions on finite-state MDPs and $2\frac{1}{2}$-player games (see, for example, [26], [22]).

B. Deterministic Rabin and Streett automata

In this section, we recall basic concepts concerning Rabin and Streett automata, which we use for the specification of ω-regular properties. Our notation is adapted from [26], [32].

A *deterministic ω-automaton* $\mathcal{A} = (\mathsf{Q}, \mathsf{Alph}, \delta, q_{\mathsf{init}}, \mathsf{Acc})$ consists of a set Q of automaton states, an alphabet Alph, a transition function $\delta : \mathsf{Q} \times \mathsf{Alph} \to \mathsf{Q}$, an initial state $q_{\mathsf{init}} \in \mathsf{Q}$ and an acceptance condition $\mathsf{Acc} \subseteq 2^{\mathsf{Q}} \times 2^{\mathsf{Q}}$. Let $\mathsf{Acc} = \{(H_1, K_1), ..., (H_n, K_n)\}$. A set $\mathsf{Q}' \subseteq \mathsf{Q}$ is called *Rabin accepting* if there exists $1 \leq i \leq n$ such that $\mathsf{Q}' \cap H_i = \emptyset$ and $\mathsf{Q}' \cap K_i \neq \emptyset$. The set Q' is called *Streett accepting* if for each $1 \leq i \leq n$ we have $\mathsf{Q}' \cap H_i \neq \emptyset$ or $\mathsf{Q}' \cap K_i = \emptyset$.

Let $\varsigma = \upsilon_1\upsilon_2\upsilon_3 \cdots$ be an infinite word over Alph. The *run* for ς is the infinite sequence $\rho_\varsigma = q_0q_1q_2\cdots$ such that $q_0 = q_{\mathsf{init}}$ and $q_i = \delta(q_{i-1}, \upsilon_i)$ for each $i \geq 1$. Let $\inf(\rho_\varsigma)$ be the set of states that occur infinitely often along ρ_ς. Then the *Rabin-accepted language* of $\mathcal{A}$ is $\mathsf{Lang}_{\mathsf{Rabin}}(\mathcal{A}) = \{\varsigma \in \mathsf{Alph}^\omega \mid \inf(\rho_\varsigma) \text{ is Rabin accepting}\}$. Similarly, the *Streett-accepted language* of $\mathcal{A}$ is defined by $\mathsf{Lang}_{\mathsf{Streett}}(\mathcal{A}) = \{\varsigma \in \mathsf{Alph}^\omega \mid \inf(\rho_\varsigma) \text{ is Streett accepting}\}$. A *deterministic Rabin automaton* is a deterministic ω-automaton for which Rabin acceptance is used to define its language. Similarly, a *deterministic Streett automaton* is a deterministic ω-automaton for which Streett acceptance is used to define its language. In the following we use the alphabet $\mathsf{Alph} = 2^{AP}$.

Let $\mathcal{R} = (L, \mathcal{X}, Events, inv, flow, prob, \mathcal{L})$ be a PRA and $\mathcal{A} = (\mathsf{Q}, \mathsf{Alph}, \delta, q_{\mathsf{init}}, \mathsf{Acc})$ be a deterministic ω-automaton. We define the *product PRA* $\mathcal{R} \otimes \mathcal{A} = (L \times \mathsf{Q}, \mathcal{X}, Events, \widehat{inv}, \widehat{flow}, \widehat{prob}, \widehat{\mathcal{L}})$ as the PRA defined in the following way:

- $\widehat{inv}(l, q) = inv(l)$ and $\widehat{flow}(l, q) = flow(l)$ for each $(l, q) \in L \times \mathsf{Q}$;
- $\widehat{prob}$ is the smallest set of probabilistic edges such that $((l, q), g, e, \widehat{p}) \in \widehat{prob}$ if there exists $(l, g, e, p) \in prob$ such that:

$$\widehat{p}(\vartheta, (l', q')) = \begin{cases} p(\vartheta, l') & \text{if } q' = \delta(q, \mathcal{L}(l')) \\ 0 & \text{otherwise.} \end{cases}$$

[2] We choose to define a finite-state PTLTS, rather than computing symbolically directly on the equivalence classes using a value iteration algorithm (see, for example, [31]), as done for non-probabilistic rectangular automata in [4], because this allows us to obtain a more precise complexity analysis.

- $\widehat{\mathcal{L}}(l, q) = \{q\}$ for each $(l, q) \in L \times \mathsf{Q}$.

In the following, we consider the $2\frac{1}{2}$-player game interpretation of the discrete-time semantics of $\mathcal{R} \otimes \mathcal{A}$, denoted in the usual way by $\mathsf{G}(\mathsf{T}^{\mathcal{R}\otimes\mathcal{A}}_{discrete})$. For $\star \in \{\mathsf{Rabin}, \mathsf{Streett}\}$, we let $\mathsf{accept}_\star$ be the set of traces of $\mathsf{G}(\mathsf{T}^{\mathcal{R}\otimes\mathcal{A}}_{discrete})$ defined by $\mathsf{accept}_\star = \{\rho \in (\mathsf{Q})^\omega \mid \inf(\rho) \text{ is } \star\text{-accepting}\}$.

Let (σ, π) be a strategy profile of $\mathsf{G}(\mathsf{T}^{\mathcal{R}}_{discrete})$. Then we define the strategy profile (σ^+, π^+) of $\mathsf{G}(\mathsf{T}^{\mathcal{R}\otimes\mathcal{A}}_{discrete})$ in the following way. First we note that, for any finite path $r = (l_0, v_0)\Lambda_0\mu_0(l_1, v_1)\Lambda_1\mu_1 \cdots (l_{n-1}, v_{n-1})\Lambda_{n-1}\mu_{n-1}(l_n, v_n)$ of $\mathsf{G}(\mathsf{T}^{\mathcal{R}}_{discrete})$, there exists a unique path $r^+ = ((l_0, q_1), v_0)\Lambda'_0\nu_0 \cdots ((l_{n-1}, q_n), v_{n-1})\Lambda'_{n-1}\nu_{n-1}$ $((l_n, q_{n+1}), v_n)$ of $\mathsf{G}(\mathsf{T}^{\mathcal{R}\otimes\mathcal{A}}_{discrete})$. Vice versa, for any such r^+ of $\mathsf{G}(\mathsf{T}^{\mathcal{R}\otimes\mathcal{A}}_{discrete})$, there exists a unique r of $\mathsf{G}(\mathsf{T}^{\mathcal{R}}_{discrete})$. Then the strategy σ^+ after path r^+ mimics the choice of σ after the path r: more precisely, if $\sigma(r) = a$, then $\sigma(r^+) = a$, for $a \in Events \cup \{time\}$. Similarly, the strategy π^+ after path $r^+ \cdot \Lambda'$ mimics the choice of π after the path $r \cdot \Lambda$ if both Λ and Λ' both correspond to either the time transition rule or the event transition rule for the same event: more precisely, if $\pi(r \cdot \Lambda) = \mu$ and $last(r^+) = ((l, q), v)$, then $\pi^+(r^+ \cdot \Lambda') = \nu$, where we have $\nu((l', \delta(q, \mathcal{L}(l'))), v') = \mu(l', v')$ for each $(l', v') \in S$ (it can be verified that such a distribution exists by definition of $\mathcal{R} \otimes \mathcal{A}$).

The following result states the equality of the probability of a strategy profile (σ, π) exhibiting traces of $\mathsf{G}(\mathsf{T}^{\mathcal{R}}_{discrete})$ accepted by $\mathcal{A}$ with acceptance condition $\star \in \{\mathsf{Rabin}, \mathsf{Streett}\}$ and the probability of the strategy profile (σ^+, π^+) exhibiting traces of $\mathsf{G}(\mathsf{T}^{\mathcal{R}\otimes\mathcal{A}}_{discrete})$ that are $\star$-accepting.

Proposition 4: Let $\mathcal{R}$ be a PRA, let $\mathcal{A}$ be a deterministic ω-automaton with $\star \in \{\mathsf{Rabin}, \mathsf{Streett}\}$ acceptance, let $(l, v) \in S$ be a state of $\mathsf{T}^{\mathcal{R}}_{discrete}$, and let (σ, π) be a strategy profile of $\mathsf{G}(\mathsf{T}^{\mathcal{R}}_{discrete})$. Then:

$$Prob^{\sigma,\pi}_{(l,v)}(\mathsf{Lang}_\star(\mathcal{A})) = Prob^{\sigma^+,\pi^+}_{((l,\delta(q_{\mathsf{init}}, Lab(l))),v)}(\mathsf{accept}_\star) \ .$$

The proposition then implies that the problem of computing $Val^{\mathsf{G}(\mathcal{R})}_{discrete}(\mathsf{Lang}_\star(\mathcal{A}))(s)$ can be reduced to that of computing $Val^{\mathsf{G}(\mathcal{R}\otimes\mathcal{A})}_{discrete}(\mathsf{accept}_\star)(s)$. By Proposition 3, we have $Val^{\mathsf{G}(\mathcal{R}\otimes\mathcal{A})}_{discrete}(\mathsf{accept}_\star)(s) = Val^{\mathsf{G}(\mathfrak{P}^k(\mathcal{R}\otimes\mathcal{A}))}(\mathsf{accept}_\star)(C)$ for the unique $\cong^k_{\mathcal{R}\otimes\mathcal{A}}$-equivalence class C for which $s \in C$. The latter value can be computed using standard methods for computing value functions for Rabin and Streett acceptance conditions on finite-state $2\frac{1}{2}$-player games [33]. Given the computational complexity results of [33], together with the fact that the size of $\mathcal{R} \otimes \mathcal{A}$ is exponential in the size of $\mathcal{R}$, we have the following result.

Theorem 3: The discrete-time verification problem for PRA with nondecreasing or bounded variables is in EXPTIME for deterministic Rabin or Streett automata objectives. The discrete-time control and controller synthesis problems for PRA with nondecreasing or bounded variables can be solved in NEXPTIME for deterministic Rabin automata objectives, and in coNEXPTIME for deterministic Streett automata objectives.

We can derive from Theorem 1 EXPTIME-lower bounds for all the problems considered in Theorem 3. The solutions to the controller synthesis problems follow from the fact that, for Rabin and Streett acceptance conditions, either finite-memory or randomized strategies for player 1 can be obtained for finite-state $2\frac{1}{2}$-player games [33].

C. Sampling-controller synthesis

In this subsection, we extend the sampling-controller synthesis construction presented for discrete-time non-probabilistic rectangular automata in [4] to PRA. The motivation for the construction arises from the observation that, at some point in time, a controller can enforce the execution of an arbitrary number of jump transitions based on probabilistic edges. To avoid this problem, [4] suggests alternating control explicitly between the controller and the plant under control. In the following, we adapt this approach to PRA.

Let $\mathcal{R}$ be a k-definable PRA. Let S, $\Rightarrow$ and Lab denote the state set, transition relation and labeling function, respectively, of the discrete-time semantics of $\mathcal{R}$. The *sampling-control PTLTS* of $\mathcal{R}$ is $\mathsf{T}^{\mathcal{R}}_{sampling} = (S \times \{control, plant\}, Events, \Rightarrow', Lab')$, where $\Rightarrow'$ is the smallest set of transitions defined in the following way: let $((s, plant), 1, \mu) \in \Rightarrow'$ if $(s, 1, \nu) \in \Rightarrow$ and $\mu(t, control) = \nu(t)$ for each $t \in S$, and, for each $e \in Events$, let $((s, control), e, \mu) \in \Rightarrow'$ if $(s, e, \nu) \in \Rightarrow$ and $\mu(t, plant) = \nu(t)$ for each $t \in S$. Furthermore, we let $Lab'(s, \star) = Lab(s)$ for each state $s \in S$ and $\star \in \{control, plant\}$.

Let φ be an ω-regular objective, and write $Val^{\mathsf{G}(\mathcal{R})}_{sampling}(\varphi)$ to denote $Val^{\mathsf{G}(\mathsf{T}^{\mathcal{R}}_{sampling})}(\varphi)$. Let s be a state of $\mathsf{T}^{\mathcal{R}}_{sampling}$. The *discrete-time sampling-control problem for* $\mathcal{R}$ *and* φ consists of deciding whether $Val^{\mathsf{G}(\mathcal{R})}_{sampling}(\varphi)(s) \geq \lambda$. The *discrete-time sampling-controller synthesis problem for* $\mathcal{R}$ *and* φ consists of the construction of a strategy for player 1 that witnesses $Val^{\mathsf{G}(\mathcal{R})}_{sampling}(\varphi)(s) \geq \lambda$, for the case in which the corresponding control decision problem returns a positive answer.

As in [4], the sampling-control PTLTS can be reduced to a discrete-time PTLTS. Let $\mathcal{R} = (L, \mathcal{X}, Events, inv, flow, prob, \mathcal{L})$ be a k-definable PRA with nondecreasing or bounded variables. First, we transform $\mathcal{R}$ to the PRA $\widetilde{\mathcal{R}} = (L, \mathcal{X} \cup \{z\}, Events, \widetilde{inv}, \widetilde{flow}, \widetilde{prob}, \mathcal{L})$. For each $l \in L$, we have $\widetilde{inv}(l) = inv(l) \wedge (z \leq 1)$, $\widetilde{flow}(l) = flow(l) \wedge (\dot{z} = 1)$. For each probabilistic edge $(l, g, e, p) \in prob$, we have $(l, g \wedge (z = 1), e, \widetilde{p}) \in \widetilde{prob}$, where $\widetilde{p}((\vartheta \wedge (z' = 0)), l') = p(\vartheta, l')$ for each $(\vartheta, l') \in \mathsf{support}(p)$. Then there exists a probabilistic bisimulation between state (l, v) of $\mathsf{T}^{\widetilde{\mathcal{R}}}_{discrete}$, where $v(z) = 0$, and the state $((l, plant), v|_{\mathcal{X}})$ of $\mathsf{T}^{\mathcal{R}}_{sampling}$, where $v|_{\mathcal{X}}$ denotes the restriction of v to $\mathcal{X}$. Similarly, there exists a probabilistic bisimulation between state (l, v) of $\mathsf{T}^{\widetilde{\mathcal{R}}}_{discrete}$, where $v(z) = 1$, and the state $((l, control), v|_{\mathcal{X}})$ of $\mathsf{T}^{\mathcal{R}}_{sampling}$. Hence the discrete-time sampling-control and sampling-controller synthesis problems can be solved using the algorithms already presented for the discrete-time control and synthesis problems, with no blow-up in complexity.

V. Dense-Time Verification for Probabilistic Timed Automata

In this section, we consider the dense-time verification problem for PTAs and ω-regular properties. Recall that an important issue in the verification of timed automata is that of time divergence: paths of a model in which the accumulated time does not exceed a bound correspond to unrealizable behavior, and therefore should be discarded during analysis [34], [35]. In contrast to the discrete-time case featured in this paper, we choose not to impose syntactic restrictions (such as the alternation between jumps and flows as featured in Section IV-C, or the structural non-Zenoness requirement of [36]) to ensure the divergence of time along paths, but instead we consider only strategies that let accumulated time diverge with probability 1 [19], [21].

Let $\mathcal{P} = (L, \mathcal{X}, \mathit{Events}, \mathit{inv}, \mathit{flow}, \mathit{prob}, \mathcal{L})$ be a PTA. To reason about time divergence in the remainder of the paper, we construct a modified PTA in the following manner [37], [38]. First we add a new atomic proposition tick to AP. The *enlarged PTA of* $\mathcal{P}$, denoted by $\overline{\mathcal{P}} = (\overline{L}, \overline{\mathcal{X}}, \overline{\mathit{Events}}, \overline{\mathit{inv}}, \overline{\mathit{flow}}, \overline{\mathit{prob}}, \overline{\mathcal{L}})$ is constructed as follows. For each location $l \in L$, we introduce a new location $\overline{l}$. Let $\overline{L} = L \cup \{\overline{l} \mid l \in L\}$, let $\overline{\mathcal{X}} = \mathcal{X} \cup \{\mathfrak{z}\}$ and let $\overline{\mathit{Events}} = \mathit{Events} \cup \{\tau\}$. For each $l \in L$, let $\overline{\mathit{inv}}(l) = \overline{\mathit{inv}}(\overline{l}) = \mathit{inv}(l) \wedge (\mathfrak{z} \leq 1)$, and let $\overline{\mathit{flow}}(l) = \overline{\mathit{flow}}(\overline{l}) = \mathit{flow}(l) \wedge (\dot{\mathfrak{z}} = 1)$. Let $\overline{\mathit{prob}} = \mathit{prob} \cup \{(l, (\mathfrak{z} = 1), \tau, \{(\vartheta_\emptyset, \overline{l}) \mapsto 1\}), (\overline{l}, (\mathfrak{z} = 1), \tau, \{(\vartheta_{\mathfrak{z}}, l) \mapsto 1\}) \mid l \in L\}$, where $\vartheta_\emptyset = \bigwedge_{x \in \overline{\mathcal{X}}} (x' = x)$ and $\vartheta_{\mathfrak{z}} = (\mathfrak{z}' = 0) \wedge \bigwedge_{x \in \mathcal{X}} (x' = x)$. Finally, let $\overline{\mathcal{L}}(l) = \mathcal{L}(l)$ and $\overline{\mathcal{L}}(\overline{l}) = \mathcal{L}(l) \cup \{\mathit{tick}\}$ for each $l \in L$. Note that tick becomes true at all natural numbered time points after the start of execution of the PTA.

Let $\mathcal{P}$ be a PTA and $\mathcal{A}$ be a deterministic ω-automaton, which we assume to be fixed throughout this section. To simplify notation, we write $\mathcal{P} \otimes \mathcal{A} = (L, \mathcal{X}, \mathit{Events}, \mathit{inv}, \mathit{prob}, \mathcal{L})$ to denote the enlarged PTA obtained from the product of $\mathcal{P}$ and $\mathcal{A}$, and henceforth omit *flow* from the definition of PTA. As in Section IV-B, we can identify a one-to-one relationship between strategies σ and σ^+ of $\mathsf{M}(\mathsf{T}^{\mathcal{P}}_{\mathit{dense}})$ and $\mathsf{M}(\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}})$, respectively, and, analogously to Proposition 4, we can show that $\mathit{Prob}^{\sigma}_{(l,v)}(\mathsf{Lang}_\star(\mathcal{A})) = \mathit{Prob}^{\sigma^+}_{((l,\delta(q_{\mathsf{init}}, \mathit{Lab}(l))),v)}(\mathsf{accept}_\star)$. Furthermore, we henceforth use the notation l for locations to refer to locations of $\mathcal{P} \otimes \mathcal{A}$ (in contrast to using pairs (l, q), where q is a state of $\mathcal{A}$).

Let $r = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \cdots$ be an infinite path of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$. Let the path of $\mathsf{M}(\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}})$ *derived from* r be the path $s_0 \mu_0 s_1 \mu_1 \cdots$. We say that path $s_0 \mu_0 s_1 \mu_1 \cdots$ of $\mathsf{M}(\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}})$ is *divergent* if it can be derived from a path $s_0 a_0 \mu_0 s_1 a_1 \mu_1 \cdots$ of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$ such that $\lim_{k\to\infty}(\sum_{i \leq k \text{ s.t. } a_i \in \mathbb{R}_{\geq 0}} a_i) = \infty$. Let $\mathsf{Timediv}$ be the set of divergent paths of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$. A strategy $\sigma \in \Sigma_{\mathsf{M}(\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}})}$ is *divergent* if $\mathit{Prob}^{\sigma}_{s}(\mathsf{Timediv}) = 1$ for all states $s \in S$ of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$. The set of divergent strategies of $\mathsf{M}(\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}})$ is denoted by $\Sigma^{\mathrm{div}}_{\mathcal{P}\otimes\mathcal{A}}$, and will henceforth be referred to simply as the set of divergent strategies of $\mathcal{P} \otimes \mathcal{A}$. Our task in the remainder of the section concerns computing the *divergent value function* on $\mathsf{M}(\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}})$ with respect to $\mathsf{accept}_\star$, defined by $\mathit{Val}^{\mathcal{R}\otimes\mathcal{A}}_{\mathrm{div}}(\mathsf{accept}_\star)(s) = \sup_{\sigma \in \Sigma^{\mathrm{div}}_{\mathcal{P}\otimes\mathcal{A}}} \mathit{Prob}^{\sigma}_{s}(\mathsf{accept}_\star)$ for each state $s \in S$.

Our first task is to construct a finite-state MDP from $\mathcal{P} \otimes \mathcal{A}$ by using the standard region graph construction [20], [19]. For $\theta \in \mathbb{R}_{\geq 0}$, we let $\mathsf{frac}(\theta) = \delta - \lfloor \theta \rfloor$. For each clock $x \in \mathcal{X}$, we let c_x be the maximal constant to which x is compared in any of the guards of probabilistic edges or invariants of $\mathcal{P}$ (if x is not involved in any clock constraint of $\mathcal{P}$, we let $c_x = 1$). Two valuations $v, v' \in \mathbb{R}^{\mathcal{X}}_{\geq 0}$ are *clock equivalent* if the following conditions are satisfied: (1) for all clocks $x \in \mathcal{X}$, we have $v(x) \leq c_x$ if and only if $v'(x) \leq c_x$; (2) for all clocks $x \in \mathcal{X}$ with $v(x) \leq c_x$, we have $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$; (3) for all clocks $x, y \in \mathcal{X}$ with $v(x) \leq c_x$ and $v(y) \leq c_y$, we have $\mathsf{frac}(v(x)) \leq \mathsf{frac}(v(y))$ if and only if $\mathsf{frac}(v'(x)) \leq \mathsf{frac}(v'(y))$; and (4) for all clocks $x \in \mathcal{X}$ with $v(x) \leq c_x$, we have $\mathsf{frac}(v(x)) = 0$ if and only if $\mathsf{frac}(v'(x)) = 0$. We use α and β to refer to classes of clock equivalence.

Two states $(l, v), (l', v')$ are *region equivalent* if (1) $l = l'$, and (2) v and v' are clock equivalent. A *region* is an equivalence class of region equivalence. Let Regions be the set of regions of $\mathcal{P} \otimes \mathcal{A}$. The number of regions corresponding to the PTA $\mathcal{P} \otimes \mathcal{A}$ is bounded by $|L| \cdot \prod_{x \in \mathcal{X}}(c_x + 1) \cdot |\mathcal{X}|! \cdot 2^{|\mathcal{X}|}$.

The set of regions of a PTA $\mathcal{P} \otimes \mathcal{A}$ can be used to construct an untimed, finite-state MDP $\mathsf{Reg}[\mathcal{P} \otimes \mathcal{A}] = (\mathsf{Regions}, \rightarrow_{\mathsf{Reg}}, \mathit{Lab}_{\mathsf{Reg}})$ in the following way. The set of states of $\mathsf{Reg}[\mathcal{P} \otimes \mathcal{A}]$ is the set Regions of regions. The transition relation $\rightarrow_{\mathsf{Reg}} \subseteq \mathsf{Regions} \times \mathsf{Dist}(\mathsf{Regions})$ is the smallest set such that:

1) $((l, \alpha), \{(l, \beta) \mapsto 1\}) \in \rightarrow_{\mathsf{Reg}}$ if there exists $((l, v), d, \{(l, v') \mapsto 1\}) \in \Rightarrow$ where $v \in \alpha$, $d \in \mathbb{R}_{\geq 0}$ and $v' \in \beta$;
2) $((l, \alpha), \nu) \in \rightarrow_{\mathsf{Reg}}$ if there exists $((l, v), e, \mu) \in \Rightarrow$ such that:
   a) $v \in \alpha$,
   b) for each $(l', \beta) \in \mathsf{Regions}$ for which there exists $(l', v') \in \mathsf{support}(\mu)$ and $v' \in \beta$ (by definition, this (l', v') will be unique), we have $\nu(l', \beta) = \mu(l', v')$, otherwise $(l', \beta) = 0$.

For each region $(l, \alpha) \in \mathsf{Regions}$, we let $\mathit{Lab}_{\mathsf{Reg}}(l, \alpha) = \mathcal{L}(l)$.

Given a clock valuation v, the unique clock equivalence class to which v belongs is denoted by $[v]$. Given a state $(l, v) \in S$, the unique region to which (l, v) belongs is $(l, [v])$, and is denoted by $[(l, v)]$. An infinite path $r = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \cdots$ of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$ corresponds to a unique infinite path $[r] = [s_0] \xrightarrow{\nu_0} [s_1] \xrightarrow{\nu_1} \cdots$. Similarly, a finite path $r = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \cdots s_{n-1} a_{n-1} \mu_{n-1} s_n$ of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$ corresponds to a unique finite path $[r] = [s_0] \xrightarrow{\nu_0} [s_1] \xrightarrow{\nu_1} \cdots \xrightarrow{\nu_{n-1}} [s_n]$. Observe that paths and region paths related by $[\cdot]$ result in the same traces (formally, $\mathsf{trace}(r) = \mathsf{trace}([r])$ for any path r).

We now introduce a concept of divergent strategies on $\mathsf{Reg}[\mathcal{P} \otimes \mathcal{A}]$. In the following we use LTL notation, which is interpreted on paths of $\mathsf{Reg}[\mathcal{P} \otimes \mathcal{A}]$ in the standard way (see, for example, [26]). An infinite path r of $\mathsf{Reg}[\mathcal{P} \otimes \mathcal{A}]$ is *region divergent* if it satisfies the condition $\Box\Diamond \mathit{tick}$. Note that an infinite path r of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$ is divergent if and only if

$[r]$ is region divergent. Hence $[\mathsf{Timediv}] = \bigcup_{r \in \mathsf{Timediv}} [r] = \{\mathsf{r} \in \mathit{Path}_{\mathit{ful}}^{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]} \mid \mathsf{r} \models \Box\Diamond \mathit{tick}\}$ is the set of all region divergent runs (where $\models$ is the standard satisfaction relation for LTL properties). A strategy $\sigma \in \Sigma_{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}$ is *region divergent* if $\mathit{Prob}_R^\sigma(\Box\Diamond \mathit{tick}) = 1$ for all regions $R \in \mathsf{Regions}$. The set of all region divergent strategies of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ is denoted by $\Sigma^{\mathrm{div}}_{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}$.

We can check whether there exists a region divergent strategy of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ by computing the set $\mathsf{Regions}_{\Box\Diamond \mathit{tick}}$ of regions for which there exists a strategy satisfying $\Box\Diamond \mathit{tick}$ with probability 1. The computation of $\mathsf{Regions}_{\Box\Diamond \mathit{tick}}$ can be done in polynomial-time in the size of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ [39]. If $\mathsf{Regions} \setminus \mathsf{Regions}_{\Box\Diamond \mathit{tick}} \neq \emptyset$, then there does not exist a region divergent strategy. In such a case, we compute the sub-MDP of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ induced by $\mathsf{Regions}_{\Box\Diamond \mathit{tick}}$ and use it in the place of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$. This allows us to assume that $\mathsf{Regions} = \mathsf{Regions}_{\Box\Diamond \mathit{tick}}$ in the remainder of this section.

From the fact that region equivalence is a probabilistic bisimulation on $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$, we obtain the following proposition. We use $\mathsf{accept}'_\star$ to denote the set $\{\rho \in (\mathsf{Q} \cup \{\mathit{tick}\})^\omega \mid \inf(\rho) \cap \mathsf{Q} \text{ is } \star\text{-accepting}\}$ of traces of $\mathcal{P}\otimes\mathcal{A}$ such that the elements occurring infinitely often along the trace, restricted to states of $\mathcal{A}$, form a $\star$-accepting set.

Proposition 5: Let $\star \in \{\mathsf{Rabin}, \mathsf{Streett}\}$ and $s \in S$ be a state of $\mathsf{T}^{\mathcal{P}\otimes\mathcal{A}}_{\mathit{dense}}$. Then:

$$\sup_{\sigma \in \Sigma^{\mathrm{div}}_{\mathcal{P}\otimes\mathcal{A}}} \mathit{Prob}_s^\sigma(\mathsf{accept}_\star) = \sup_{\sigma \in \Sigma^{\mathrm{div}}_{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}} \mathit{Prob}_{[s]}^\sigma(\mathsf{accept}'_\star) \ .$$

Hence, to compute $\mathit{Val}^{\mathcal{R}\otimes\mathcal{A}}_{\mathrm{div}}(\mathsf{accept}_\star)$, it suffices to compute $\sup_{\sigma \in \Sigma^{\mathrm{div}}_{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}} \mathit{Prob}_R^\sigma(\mathsf{accept}'_\star)$ on $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ for each $R \in \mathsf{Regions}$. In the remainder of this section, we generally omit the subscript from the sets of strategies of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$, and write Σ for $\Sigma_{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}$, and Σ^{div} for $\Sigma^{\mathrm{div}}_{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}$.

We recall the notion of *time-divergent EC* from [21]. A time-divergent EC (A, B) is an EC of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ such that $\mathit{tick} \in \mathit{Lab}_{\mathsf{Reg}}(R)$ for some region $R \in A$. For an infinite path $\mathsf{r} \in \mathit{Path}_{\mathit{ful}}^{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}$, let $A_{\mathsf{r}} = \{R \mid \overset{\infty}{\exists}\, i \geq 0 . \mathsf{r}(i) = R\}$ and $B_{\mathsf{r}} = \{(R, \nu) \mid \overset{\infty}{\exists}\, i \geq 0 . R \in A_{\mathsf{r}} \wedge \mathit{step}(\mathsf{r}, i) = \nu\}$. Let $\mathrm{InfEC}(\mathsf{r}) = (A_{\mathsf{r}}, B_{\mathsf{r}})$. Note that a path $\mathsf{r} \in \mathit{Path}_{\mathit{ful}}^{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}$ of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ is region divergent if and only if $\mathrm{InfEC}(\mathsf{r})$ is a time-divergent EC. For $A \subseteq \mathsf{Regions}$ and $B \subseteq \rightarrow_{\mathsf{Reg}}$, let $\mathit{Path}_{\mathit{ful}}^{(A,B)}(R) = \{\mathsf{r} \in \mathit{Path}_{\mathit{ful}}^{\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]}(R) \mid \mathrm{InfEC}(\mathsf{r}) = (A, B)\}$. The next lemma states that a probabilistically region divergent strategy will be confined eventually to time-divergent ECs with probability 1.

Lemma 1 ([21]): Let $\mathcal{E}$ be the set of time-divergent ECs of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$, let $R \in \mathsf{Regions}$ and let $\sigma \in \Sigma^{\mathrm{div}}$. Then $\mathit{Prob}_R^\sigma(\bigcup_{(A,B)\in\mathcal{E}} \mathit{Path}_{\mathit{ful}}^{(A,B)}(R)) = 1$.

Next, recall that methods for computing value functions for Rabin and Streett acceptance over the full set of strategies of a finite-state MDPs rely on the computation of a set of maximal ECs, $\mathcal{E}_{\mathsf{Rabin}}$ and $\mathcal{E}_{\mathsf{Streett}}$, respectively; then the overall value function is obtained by calculating the maximum probability of reaching the computed maximal ECs. Algorithms for computing $\mathcal{E}_{\mathsf{Rabin}}$ and $\mathcal{E}_{\mathsf{Streett}}$ have been presented in [24], [40] and [33], respectively. For $\star \in \{\mathsf{Rabin}, \mathsf{Streett}\}$, we let $\mathcal{E}_\star^{\mathrm{div}} = \{(A, B) \in \mathcal{E}_\star \mid \exists R \in A \text{ s.t. } \mathit{tick} \in \mathit{Lab}_{\mathsf{Reg}}(R)\}$, and let $\mathsf{F}_\star = \bigcup_{(A,B)\in\mathcal{E}_\star^{\mathrm{div}}} A$. The next lemma follows from standard reasoning that, when in an EC, we can (using either finite-memory or randomized strategies) visit all regions of the EC infinitely often with probability 1, thereby satisfying the acceptance condition *and* the condition of region divergence with probability 1.

Lemma 2: Let $\star \in \{\mathsf{Rabin}, \mathsf{Streett}\}$ and $R \in \mathsf{F}_\star$. Then there exists a region divergent strategy $\sigma \in \Sigma^{\mathrm{div}}$ such that $\mathit{Prob}_R^\sigma(\mathsf{accept}'_\star) = 1$.

Proposition 6: Let $\star \in \{\mathsf{Rabin}, \mathsf{Streett}\}$ and $R \in \mathsf{Regions}$. Then:

$$\sup_{\sigma \in \Sigma^{\mathrm{div}}} \mathit{Prob}_R^\sigma(\mathsf{accept}'_\star) = \sup_{\sigma \in \Sigma^{\mathrm{div}}} \mathit{Prob}_R^\sigma(\Diamond \mathsf{F}_\star) \ .$$

Proof: (Sketch.) ($\leq$) We first show that $\mathit{Prob}_R^\sigma(\mathsf{accept}'_\star) \leq \mathit{Prob}_R^\sigma(\Diamond \mathsf{F}_\star)$ for any $\sigma \in \Sigma^{\mathrm{div}}$. By Lemma 1, the strategy σ confines itself eventually in time-divergent ECs with probability 1. The probability $\mathit{Prob}_R^\sigma(\mathsf{accept}'_\star)$ is obtained from the sum of the probability of being confined to ECs that are sub-MDPs of the maximal ECs in $\mathcal{E}_\star^{\mathrm{div}}$. Given that $\mathit{Prob}_R^\sigma(\Diamond \mathsf{F}_\star)$ is the probability of reaching such maximal ECs, the inequality follows.

($\geq$) We show that, for any $\sigma \in \Sigma^{\mathrm{div}}$, we can obtain $\sigma' \in \Sigma^{\mathrm{div}}$ such that $\mathit{Prob}_s^{\sigma'}(\mathsf{accept}'_\star) \geq \mathit{Prob}_s^\sigma(\Diamond \mathsf{F}_\star)$. The strategy σ' is obtained by copying the choices of σ, except for the paths in which a $\mathsf{F}_\star$ has been reached. For those paths, as soon as a region R in $\mathsf{F}_\star$ is reached, σ' switches from copying σ to behaving as a region divergent strategy σ'' for which $\mathit{Prob}_R^{\sigma''}(\mathsf{accept}'_\star) = 1$, which exists by Lemma 2. It can be observed that σ' is region divergent and that $\mathit{Prob}_s^{\sigma'}(\mathsf{accept}'_\star) \geq \mathit{Prob}_s^\sigma(\Diamond \mathsf{F}_\star)$. ■

From Proposition 6, we can reduce the computation of $\sup_{\sigma \in \Sigma^{\mathrm{div}}} \mathit{Prob}_R^\sigma(\mathsf{accept}'_\star)$ for each region $R \in \mathsf{Regions}$ to that of $\sup_{\sigma \in \Sigma^{\mathrm{div}}} \mathit{Prob}_R^\sigma(\Diamond \mathsf{F}_\star)$. The final question that remains is how to compute this value. The following lemma states that, for reachability properties, we can find an optimal region divergent strategy.

Lemma 3 ([21]): Let $\mathsf{F} \subseteq L$ be a set of locations of $\mathcal{P}\otimes\mathcal{A}$. Then, for any $R \in \mathsf{Regions}$ and any strategy $\sigma \in \Sigma$, there exists a region divergent strategy $\sigma' \in \Sigma^{\mathrm{div}}$ such that $\mathit{Prob}_R^\sigma(\Diamond \mathsf{F}) \leq \mathit{Prob}_R^{\sigma'}(\Diamond \mathsf{F})$.

Lemma 3 and the fact that $\Sigma^{\mathrm{div}} \subseteq \Sigma$ imply that:

$$\sup_{\sigma \in \Sigma^{\mathrm{div}}} \mathit{Prob}_R^\sigma(\Diamond \mathsf{F}_\star) = \sup_{\sigma \in \Sigma} \mathit{Prob}_R^\sigma(\Diamond \mathsf{F}_\star) \ .$$

In order to compute $\sup_{\sigma \in \Sigma} \mathit{Prob}_R^\sigma(\Diamond \mathsf{F}_\star)$, we can use standard computations for reachability properties on MDPs.

Noting that the size of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$ is exponential in the size of $\mathcal{P}$, that the computation of $\mathsf{F}_\star$ can be done in polynomial time in the size of $\mathsf{Reg}[\mathcal{P}\otimes\mathcal{A}]$, and the lower bound of Theorem 1, we have the following result.

Theorem 4: The dense-time verification problem for PTA is EXPTIME-complete for deterministic Rabin or Streett automata objectives.

VI. Conclusion

In this paper we have presented methods for discrete-time verification and control problems for PRA, and for dense-time verification problems for PTA. For all of the considered problems, the solution is obtained using a probabilistic bisimulation: $\cong^{k}_{\mathcal{R}}$-equivalence for the case of discrete-time PRA, and region equivalence for PTA. Note that the discrete-time approach used in this paper is different from that in [41], in which a PTA model with digital clocks in used as a correctness-preserving representation of the usual dense-time PTA model, rather than assuming from the outset that the PTA makes jump transitions only at integer points in time, as we do in the discrete-time semantics this paper. By considering ω-regular properties, we have enlarged the class of properties that can be considered in the PRA and PTA framework.

Acknowledgment

The author is supported in part by the MIUR-PRIN project PaCo - Performability-Aware Computing: Logics, Models and Languages. The author would like to thank Arnaud Sangnier for useful comments on a draft of this paper.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *TCS*, vol. 138, no. 1, pp. 3–34, 1995.

[2] T. A. Henzinger, "The theory of hybrid automata," in *Proc. LICS'96*. IEEE, 1996, pp. 278–292.

[3] H. Wong-Toi, "The synthesis of controllers for linear hybrid automata," in *Proc. CDC'97*. IEEE, 1997, pp. 4607–4612.

[4] T. A. Henzinger and P. W. Kopke, "Discrete-time control for rectangular hybrid automata," *TCS*, vol. 221, no. (1-2), pp. 369–392, 1999.

[5] T. A. Henzinger, B. Horowitz, and R. Majumdar, "Rectangular hybrid games," in *Proc. CONCUR'99*, ser. LNCS, vol. 1664. Springer, 1999, pp. 320–335.

[6] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli, "Effective synthesis of switching controllers for linear systems," *Proc. IEEE*, vol. 88, pp. 1011–1025, 2000.

[7] J. Hu, J. Lygeros, and S. Sastry, "Towards a theory of stochastic hybrid systems," in *Proc. HSCC'00*, ser. LNCS, vol. 1790. Springer, 2000, pp. 160–173.

[8] J. Sproston, "Decidable model checking of probabilistic hybrid automata," in *Proc. FTRTFT'00*, ser. LNCS, vol. 1926. Springer, 2000, pp. 31–45.

[9] ——, "Model checking for probabilistic timed and hybrid systems," Ph.D. dissertation, School of Computer Science, University of Birmingham, 2001.

[10] M. L. Bujorianu, "Extended stochastic hybrid systems and their reachability problem," in *Proc. HSCC'04*, ser. LNCS, vol. 2993. Springer, 2004, pp. 234–249.

[11] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008.

[12] L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E. M. Hahn, "Safety verification for probabilistic hybrid systems," in *Proc. CAV'10*, ser. Lecture Notes in Computer Science, vol. 6174. Springer, 2010, pp. 196–211.

[13] J. Assouramou and J. Desharnais, "Continuous time and/or continuous distributions," in *Proc. EPEW'10*, ser. LNCS, vol. 6342. Springer, 2010, pp. 99–114.

[14] M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang, "Measurability and safety verification for stochastic hybrid systems," in *Proc. HSCC'11*. ACM, 2011, to appear.

[15] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *J. Comput. Syst. Sci.*, vol. 57, no. 1, pp. 94–124, 1998.

[16] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, pp. 540–554, 1998.

[17] L. Doyen, T. A. Henzinger, and J.-F. Raskin, "Automatic rectangular refinement of affine hybrid systems," in *Proc. FORMATS'05*, ser. LNCS, vol. 3829. Springer, 2005, pp. 144–161.

[18] H. Gregersen and H. E. Jensen, "Formal design of reliable real time systems," Master's Thesis, Department of Mathematics and Computer Science, Aalborg University, 1995.

[19] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston, "Automatic verification of real-time systems with discrete probability distributions," *TCS*, vol. 286, pp. 101–150, 2002.

[20] R. Alur and D. L. Dill, "A theory of timed automata," *TCS*, vol. 126, no. 2, pp. 183–235, 1994.

[21] J. Sproston, "Strict divergence for probabilistic timed automata," in *Proc. CONCUR'09*, ser. LNCS, vol. 5710. Springer, 2009, pp. 620–636.

[22] K. Chatterjee and T. A. Henzinger, "A survey of stochastic omega-regular games," *J. Comput. Syst. Sci.*, 2011, to appear.

[23] J. G. Kemeny, J. L. Snell, and A. W. Knapp, *Denumerable Markov Chains*, 2nd ed., ser. Graduate Texts in Mathematics. Springer, 1976.

[24] L. de Alfaro, "Formal verification of probabilistic systems," Ph.D. dissertation, Stanford University, Department of Computer Science, 1997.

[25] M. Jurdziński, F. Laroussinie, and J. Sproston, "Model checking probabilistic timed automata with one or two clocks," *LMCS*, vol. 4, no. 3, pp. 1–28, 2008.

[26] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT Press, 2008.

[27] F. Laroussinie and J. Sproston, "State explosion in almost-sure probabilistic reachability," *IPL*, vol. 102, no. 6, pp. 236–241, 2007.

[28] V. Forejt, M. Kwiatkowska, G. Norman, and A. Trivedi, "Expected reachability-time games," in *Proc. FORMATS'10*, ser. LNCS, vol. 6246. Springer, 2010, pp. 122–136.

[29] K. G. Larsen and A. Skou, "Bisimulation through probabilistic testing," *I & C*, vol. 94, no. 1, pp. 1–28, 1991.

[30] R. Segala and N. A. Lynch, "Probabilistic simulations for probabilistic processes," *Nordic Journal of Computing*, vol. 2, no. 2, pp. 250–273, 1995.

[31] K. Chatterjee and T. A. Henzinger, "Value iteration," in *25 Years of Model Checking - History, Achievements, Perspectives*, ser. LNCS. Springer, 2008, vol. 5000, pp. 107–138.

[32] C. Baier, M. Größer, and F. Ciesinski, "Model checking linear-time properties of probabilistic systems," in *Handbook of Weighted Automata*, ser. EATCS Monographs in Theoretical Computer Science. Springer, 2009, pp. 519–570.

[33] K. Chatterjee, L. de Alfaro, and T. A. Henzinger, "The complexity of stochastic Rabin and Streett games," in *Proc. ICALP'05*, ser. LNCS, vol. 3580. Springer, 2005, pp. 878–890.

[34] R. Alur, C. Courcoubetis, and D. L. Dill, "Model-checking in dense real-time," *I & C*, vol. 104, no. 1, pp. 2–34, 1993.

[35] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine, "Symbolic model checking for real-time systems," *I & C*, vol. 111, no. 2, pp. 193–244, 1994.

[36] S. Tripakis, S. Yovine, and A. Bouajjani, "Checking timed Büchi automata emptiness efficiently," *FMSD*, vol. 26, no. 3, pp. 267–292, 2005.

[37] R. Alur and T. Henzinger, "Real-time system = discrete system + clock variables," *STTT*, vol. 1, pp. 86–109, 1997.

[38] L. de Alfaro, M. Faella, T. Henzinger, R. Majumdar, and M. Stoelinga, "The element of surprise in timed games," in *Proc. CONCUR'03*, ser. LNCS, vol. 2761. Springer, 2003, pp. 144–158.

[39] K. Chatterjee, M. Jurdziński, and T. Henzinger, "Simple stochastic parity games," in *Proc. CSL'03*, ser. LNCS, vol. 2803. Springer, 2003, pp. 100–113.

[40] C. Baier and M. Kwiatkowska, "Model checking for a probabilistic branching time logic with fairness," *Dist. Comp.*, vol. 11, no. 3, pp. 125–155, 1998.

[41] M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston, "Performance analysis of probabilistic timed automata using digital clocks," *FMSD*, vol. 29, pp. 33–78, 2006.