Decidable Model Checking of Probabilistic Hybrid Automata

Jeremy Sproston*

School of Computer Science, University of Birmingham,
Birmingham B15 2TT, United Kingdom. J.Sproston@cs.bham.ac.uk

Abstract. Hybrid automata offer a framework for the description of systems with both discrete and continuous components, such as digital technology embedded in an analogue environment. Traditional uses of hybrid automata express choice of transitions purely in terms of nondeterminism, abstracting potentially significant information concerning the relative likelihood of certain behaviours. To model such probabilistic information, we present a variant of hybrid automata augmented with discrete probability distributions. We concentrate on restricted subclasses of the model in order to obtain decidable model checking algorithms for properties expressed in probabilistic temporal logics.

1 Introduction

Many systems, such as embedded controllers, can be modelled in terms of interaction between discrete and continuous components. Examples of such *hybrid systems* include robots, medical equipment and manufacturing processes. Traditionally, formal techniques for the description of hybrid systems express the system model purely in terms of nondeterminism. However, it may be desirable to express the relative likelihood of the system exhibiting certain behaviour. This notion is particularly important when considering fault-tolerant systems, in which the occurrence of the discrete event *malfunction* is less likely than the event *correct_action*. Furthermore, it may be appropriate to model the likelihood of an event changing with respect to the continuous behaviour of the environment; for example, *malfunction* may become more likely if the system is operating at extreme temperatures or at high speeds. We may also wish to have a model checking algorithm for verifying automatically such hybrid systems against temporal logic properties referring explicitly to likelihoods. The feasibility of such verification methods is suggested by the successful development of model checking algorithms and tools both in the domain of hybrid systems [11] and that of discrete, finite-state probabilistic-nondeterministic systems [8].

Therefore, we extend the model of *hybrid automata* [1], a framework for the description of hybrid systems, with *discrete probability distributions*. This approach is inspired by the work of [16], which presents firstly a model of timed automata (a highly restricted subclass of hybrid automata) extended with such

* Supported in part by the EPSRC grant GR/N22960.

distributions, and secondly a decidable algorithm for verifying instances of this model against formulae of a probabilistic temporal logic. Our new model, *probabilistic hybrid automata*, differs from traditional hybrid automata in that the edge relation of the graph representing the system's discrete component is both nondeterministic and probabilistic in nature. More precisely, instead of making a purely nondeterministic choice over the set of currently enabled edges, we nondeterministically choose amongst the set of *enabled discrete probability distributions*, each of which is defined over a set of edges. Then, a probabilistic choice as to which edge to take according to the selected distribution is performed. Although probability is defined to affect directly only the *discrete* dynamics of the model, the proposed model would nevertheless be useful for the analysis of many systems, such as embedded technology operating according to randomised algorithms, or the aforementioned fault-tolerant systems, for which an appropriate probabilistic hybrid automaton may be obtained given appropriate failure specifications of the system's components.

A substantial body of work has been devoted to exploring notions of *decidability* of *non-probabilistic* hybrid automata, particularly with regard to problems which underly model checking procedures. These problems are usually addressed by utilising refinement relations such as simulation and bisimulation in order to introduce notions of equivalence and abstraction on the infinite state space of a hybrid automaton. It follows that model checking can then be performed not on the original, infinite state space, but on a quotient induced by an equivalence relation; therefore, if the number of equivalence classes for a given hybrid automaton is finite, then model checking is decidable. However, such finitary quotients exist only for certain classes of model [12, 18]. In particular, the *rectangular automata* of [12] feature differential inequalities which describe the continuous evolution of system variables taking place within piecewise-linear, convex envelopes, and can be used to state, for example, that a system variable increases between 1 and 3 units per second. Another such class is that of *o-minimal hybrid automata* [18], which permit expressive (albeit deterministic) continuous behaviour, and feature restricted discrete transitions. The remit of this paper is to extend these classes with discrete probability distributions, and to explore the way in which established model checking techniques for probabilistic-nondeterministic systems [7, 6] may be used to verify such models against probabilistic temporal logic specifications. This provides us with a means to verify probabilistic extensions of rectangular or o-minimal automata against properties such as 'soft deadlines' (for example, a response to a request will be granted within 5 seconds with probability at least 0.95), or those which refer to the probability of malfunction or component failure (such as, with probability 0.999 or greater, less than 1 litre of coolant leaks from the nuclear reactor before an alarm is sounded).

The paper proceeds by first introducing probabilistic hybrid automata in Section 2. Section 3 explains how their semantics can be presented in terms of infinite-state, nondeterministic-probabilistic transition systems. Strategies for model checking probabilistic rectangular automata and probabilistic o-minimal automata are presented in Section 4.

2 Probabilistic Hybrid Automata

The purpose of this section is to present a model for probabilistic hybrid systems using the framework of hybrid automata, based on the probabilistic timed automata of [16]. For a set Y, a (discrete probability) *distribution* on Y is a function $\mu : Y \to [0,1]$ such that $\mu(y) > 0$ for at most countably many $y \in Y$ and $\sum_{y \in Y} \mu(y) = 1$. We use $\mathsf{Dist}(Y)$ to denote the set of all distributions on Y. Given a distribution μ on a set Y, let $\mathsf{support}(\mu)$ be the support of μ; that is, the set of elements y of Y such that $\mu(y) > 0$. If Y contains one element, then a distribution over Y is called a *Dirac distribution*, and is denoted $\mathcal{D}(y)$ where $Y = \{y\}$. Next, let $\mathcal{X} = \{x_1, ..., x_n\}$ be a set of real-valued variables. We write $\mathbf{a} \in \mathbb{R}^n$ for a vector of length n which assigns a *valuation* $\mathbf{a}_i \in \mathbb{R}$ to each variable $x_i \in \mathcal{X}$. We fix a finite set AP of atomic propositions.

A *probabilistic hybrid automaton* $H = (\mathcal{X}, V, L, init, inv, flow, prob, \langle pre_v \rangle_{v \in V})$ comprises of the following components:

Variables. $\mathcal{X}$ is a finite set of real-valued variables.
Control modes. V is a finite set of *control modes*.
Labelling function. The function $L : V \to 2^{\mathrm{AP}}$ assigns a set of atomic propositions to each control mode.
Initial set. The function $init : V \to 2^{\mathbb{R}^n}$ maps every control mode to an *initial set* in $\mathbb{R}^n$.
Invariant set. The function $inv : V \to 2^{\mathbb{R}^n}$ maps every control mode to an *invariant set* in $\mathbb{R}^n$.
Flow inclusion. The partial function $flow : V \times \mathbb{R}^n \to 2^{\mathbb{R}^n}$ maps control modes and valuations to a *flow inclusion* in $\mathbb{R}^n$, and is such that, for each $v \in V$ and $\mathbf{a} \in inv(v)$, the set $flow(v, \mathbf{a})$ is defined.
Probability distributions. The function $prob : V \to \mathcal{P}_{fn}(\mathsf{Dist}(V \times 2^{\mathbb{R}^n} \times 2^{\mathcal{X}}))$ maps every control mode to a finite, non-empty set of distributions over the set of control modes, and the powersets of both $\mathbb{R}^n$ and $\mathcal{X}$. Therefore, each control mode $v \in V$ will be associated with a set of distributions denoted by $prob(v) = \{\mu_v^1, ..., \mu_v^m\}$ for some finite $m \geq 1$.
Pre-condition sets. For each $v \in V$, the function $pre_v : prob(v) \to 2^{\mathbb{R}^n}$ maps every probability distribution associated with a control mode to a *pre-condition set* in $\mathbb{R}^n$.

For simplicity, and without loss of generality, we assume that the initial point is unique; that is, for a control mode $v_0 \in V$, the initial condition $init(v_0)$ is a singleton in $\mathbb{R}^n$, and $init(v') = \emptyset$ for all other $v' \in V \setminus \{v_0\}$. Therefore, control of the model commences in a mode v_0 with the variable valuation given by $init(v_0)$. When control of a probabilistic rectangular automaton is in a given mode $v \in V$, the values of the real-valued variables in $\mathcal{X}$ change continuously with respect to time. Such continuous evolution is determined by the mode's flow inclusion $flow$; that is, $flow(v, \mathbf{a})$ gives the set of values that the first derivative with respect to time $\frac{dx_i}{dt}$ of each variable $x_i \in \mathcal{X}$ may take in the control mode v when the current value of the variables is given by $\mathbf{a}$. A discrete transition, henceforth called a *control switch*, from v to another mode, may take place if the

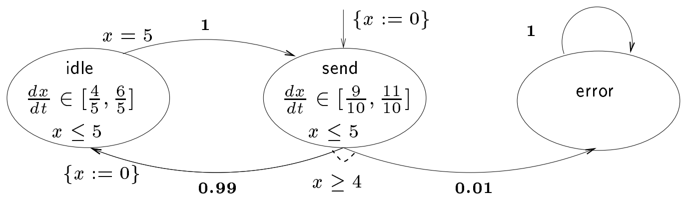

Fig. 1. The probabilistic hybrid automaton H_1.

pre-condition $pre_v(\mu)$ of a distribution $\mu \in prob(v)$ is satisfied by the current values of the variables. In such a case, we say that μ is *enabled*. Conversely, such a control switch *must* take place if the passage of some time would result in the current variable values leaving the invariant set $inv(v)$. For simplicity, the invariant and pre-condition sets are subject to the assumption that, if allowing any amount of time to elapse would result in the departure of the set $inv(v)$, then the current point in the continuous state space must be in the pre-condition set of at least one distribution in $prob(v)$.

Given that it has been decided to make a control switch via a particular enabled distribution $\mu \in prob(v)$, then a probabilistic choice as to the target mode of the switch, and to the discrete changes to the continuous variables, is performed. More precisely, with probability $\mu(w, post, X)$, a transition is made to mode $w \in V$ with the valuation $\mathbf{b}$, such that $\mathbf{b}$ is in the set $post \subseteq \mathbb{R}^n$ and $\mathbf{b}_i = \mathbf{a}_i$ for every variable $x_i \in \mathcal{X} \setminus X$. Sets such as *post* are referred to as *post-conditions*, and variable sets such as X are referred to as *reset sets*. Together, post-conditions and reset sets determine the effects that a probabilistic hybrid automaton's control switches have on its continuous variables. These sets are subject to the following simplifying assumption: for each $v \in V$, each $\mu \in prob(v)$, and each $(w, post, X) \in \mathsf{support}(\mu)$, we have $post \subseteq inv(w)$.

An example of a probabilistic hybrid automaton is given in Figure 1. A number of standard conventions concerning the diagrammatic representation of hybrid automata are used here (see, for example, [1]). The probabilistic hybrid automaton H_1 models a process in which data packets are repeatedly sent from a sender to a receiver. We explain only the action of a fragment of the model. The process measures time according to a drifting clock, which is represented by the variable x. When the process is ready to send a packet (control is in the mode send), the clock progresses at any rate between $\frac{9}{10}$ and $\frac{11}{10}$ units per millisecond, and when it is idle (control is in the mode idle), the clock progresses at between $\frac{4}{5}$ and $\frac{6}{5}$ units per millisecond. In the former case, the process waits until its clock is equal to or greater than 4 before transmitting data, and must transmit before the clock exceeds 5. However, when transmission takes place, there is a 1% chance of the occurrence of an unrecoverable error (control passes to the mode error), as represented by a distribution over the edges from send to idle, and from send to error. More precisely, the distribution $\mu \in prob(\mathsf{send})$ is such that $\mu(\mathsf{idle}, [0, 0], \{x\}) = 0.99$, $\mu(\mathsf{error}, \mathbb{R}, \emptyset) = 0.01$ and $pre_{\mathsf{send}}(\mu) = [4, \infty)$.

Subclasses of probabilistic hybrid automata. A *rectangular inequality* over $\mathcal{X}$ is of the form $x_i \sim k$, where $x_i \in \mathcal{X}$, $\sim \in \{<, \leq, =, \geq, >\}$ and $k \in \mathbb{Q}$. A *rectangular predicate* over $\mathcal{X}$ is a conjunction of rectangular inequalities over $\mathcal{X}$. For any rectangular predicate P, the set of valuations for which P is true when each $x_i \in \mathcal{X}$ is replaced by its corresponding valuation $\mathbf{a}_i$ is denoted by $[\![P]\!]$ (intuitively, $[\![P]\!]$ is the set of points in $\mathbb{R}^n$ that satisfy P). Furthermore, we call $[\![P]\!]$ a *rectangle*, and occasionally refer to such a set as *rectangular*. A closed and bounded rectangle is described as being *compact*. The set of rectangles over $\mathcal{X}$ is obtained from the set of rectangular inequalities over $\mathcal{X}$, and is denoted $\mathit{Rect}(\mathcal{X})$. The projection of the rectangle Z onto the axis of x_i is denoted by Z_i.

We now introduce a probabilistic extension of rectangular automata [12]. A *probabilistic rectangular automaton* R is a probabilistic hybrid automaton such that, for every $v \in V$, the sets $\mathit{inv}(v)$ and $\mathit{flow}(v, \cdot)$ are rectangles, for every $\mu \in \mathit{prob}(v)$, the set $\mathit{pre}_v(\mu)$ is a rectangle, and, for every $(w, \mathit{post}, X) \in \mathsf{support}(\mu)$, the set post is a rectangle. The probabilistic rectangular automaton R is *initialised* if, for every pair of modes $v, w \in V$, and every $x_i \in \mathcal{X}$ for which $\mathit{flow}(v, \cdot)_i \neq \mathit{flow}(w, \cdot)_i$, then if there exists a distribution $\mu \in \mathit{prob}(v)$ and a tuple $(w, \mathit{post}, X) \in \mathsf{support}(\mu)$, we have $x_i \in X$. Intuitively, if the execution of a control switch results in a variable x_i changing the condition on its continuous evolution, then the value of x_i must be reinitialised. The probabilistic rectangular automaton R has *deterministic jumps* if for every $v, w \in V$, $\mu \in \mathit{prob}(v)$, and $(w, \mathit{post}, X) \in \mathsf{support}(\mu)$, then the set post_i is a singleton for every $x_i \in X$. Intuitively, this requirement states that, for every control switch, each variable either remains unchanged or is deterministically reset to a new value. A *probabilistic multisingular automaton* M is an initialised probabilistic rectangular automaton with deterministic jumps such that, for each $v \in V$ and for each $x_i \in \mathcal{X}$, we have $\mathit{flow}(v, \cdot)_i = k$ for some $k \in \mathbb{N}$. [1]

Next, the o-minimal hybrid automata of [18] are extended to the probabilistic context. The definition is as in Theorem 5.7 of [3], except for the following alterations: naturally, we dispense with the notion of edges connecting control modes, and replace them with a set of distributions; also, for every $v \in V$ and all $\mu \in \mathit{prob}(v)$, the sets $\mathit{inv}(v)$ and $\mathit{pre}_v(\mu)$ are semi-algebraic with rational coefficients, and, for every $(w, \mathit{post}, X) \in \mathsf{support}(\mu)$, the set post is semi-algebraic with rational coefficients and $X = \mathcal{X}$. Finally, to obtain *probabilistic o-minimal hybrid automata*, we assume that the initial point of the model is unique.

3 Semantics of Probabilistic Hybrid Automata

3.1 Concurrent Probabilistic Systems

The underlying transition system of a probabilistic hybrid automaton will take the form of a *concurrent probabilistic system* [6]. These systems are based on

[1] Observe that a *probabilistic timed automaton* [16] is a probabilistic multisingular automaton such that $\mathit{flow}(v, \cdot)_i = 1$ for each $v \in V$ and for each $x_i \in \mathcal{X}$.

Markov decision processes, and are a state-labelled variant of the "simple probabilistic automata" of [21]. Formally, a *concurrent probabilistic system* $\mathcal{S}$ is a tuple $(Q, q^0, \mathcal{L}, \Sigma, \mathit{Steps})$, where Q is a (possibly infinite) set of *states*, $q^0 \in Q$ is the *initial state*, $\mathcal{L} : Q \to 2^{\mathrm{AP}}$ is a function assigning a finite set of atomic propositions to each state, Σ is a set of *events*, and $\mathit{Steps} \subseteq \Sigma \times \mathsf{Dist}(Q)$ is a function which assigns to each state a non-empty set $\mathit{Steps}(q)$ of pairs comprising of an event σ and a distribution ν on Q.

A *transition* of $\mathcal{S}$ from state q comprises of a nondeterministic choice of an event-distribution pair $(\sigma, \nu) \in \mathit{Steps}(q)$, followed by a probabilistic choice of a next-state q' according to ν such that $\nu(q') > 0$, and is denoted by $q \xrightarrow{\sigma,\nu} q'$. A *path* of $\mathcal{S}$ is a non-empty finite or infinite sequence of transitions of the form $\omega = q_0 \xrightarrow{\sigma_0,\nu_0} q_1 \xrightarrow{\sigma_1,\nu_1} q_2 \xrightarrow{\sigma_2,\nu_2} \cdots$. The special case $\omega = q$, for some $q \in Q$, is also a path. The following notation is employed when reasoning about paths. For a path ω, the first state of ω is denoted by $\mathit{first}(\omega)$, and, if ω is finite, the last state of ω is denoted by $\mathit{last}(\omega)$. If ω is infinite, then $\mathit{step}(\omega, i)$ is the event-distribution pair associated with the i-th transition for each $i \in \mathbb{N}$. We denote by $\mathit{Path}_{\mathit{ful}}$ the set of infinite paths, and by $\mathit{Path}_{\mathit{ful}}(q)$ the set of paths ω in $\mathit{Path}_{\mathit{ful}}$ such that $\mathit{first}(\omega) = q$.

An *adversary* of a concurrent probabilistic system $\mathcal{S}$ is a function A mapping every finite path ω of $\mathcal{S}$ to an event-distribution pair (σ, ν) such that $(\sigma, \nu) \in \mathit{Steps}(\mathit{last}(\omega))$. Intuitively, an adversary resolves all of the nondeterministic choices of $\mathcal{S}$. For an adversary A of $\mathcal{S}$, we define $\mathit{Path}^A_{\mathit{ful}}$ to be the set of paths in $\mathit{Path}_{\mathit{ful}}$ such that $\mathit{step}(\omega, i) = A(\omega^{(i)})$ for all $i \in \mathbb{N}$. Furthermore, $\mathit{Path}^A_{\mathit{ful}}(q)$ is defined to be the set of paths of $\mathit{Path}^A_{\mathit{ful}}$ such that $\mathit{first}(\omega) = q$ for all $\omega \in \mathit{Path}_{\mathit{ful}}$. For each adversary, we can define a probability measure Prob^A on infinite paths in the standard manner (see, for example, [6]).

We introduce two state relations for concurrent probabilistic systems, namely *probabilistic bisimulation and simulation.* In the standard manner, the concept of weight functions [15] is used to provide the basis of the definition of simulation, and bisimulation is defined as a symmetric simulation [21]. Let $\mathcal{R} \subseteq Q_1 \times Q_2$ be a relation between the two sets Q_1, Q_2, and ν_1, ν_2 distributions such that $\nu_1 \in \mathsf{Dist}(Q_1)$ and $\nu_2 \in \mathsf{Dist}(Q_2)$. A *weight function* for (ν_1, ν_2) with respect to $\mathcal{R}$ is a function $w : Q_1 \times Q_2 \to [0, 1]$ such that, for all $q_1 \in Q_1$, $q_2 \in Q_2$:

1. if $w(q_1, q_2) > 0$, then $(q_1, q_2) \in \mathcal{R}$, and
2. $\sum_{q' \in Q_2} w(q_1, q') = \nu_1(q_1)$, and $\sum_{q' \in Q_1} w(q', q_2) = \nu_2(q_2)$.

We write $\nu_1 \mathcal{R} \nu_2$ if there exists a weight function for (ν_1, ν_2) with respect to $\mathcal{R}$. For example, if $Q_1 = \{q_1, q_1'\}$, $Q_2 = \{q_2, q_2'\}$, $\nu_1(q_1) = \nu_1(q_1') = \frac{1}{2}$, $\nu_2(q_2) = \frac{1}{3}$, $\nu_2(q_2') = \frac{2}{3}$, and $\mathcal{R} = \{(q_1, q_2), (q_1, q_2'), (q_1', q_2')\}$, then a weight function w for (ν_1, ν_2) with respect to $\mathcal{R}$ is $w(q_1, q_2) = \frac{1}{3}$, $w(q_1, q_2') = \frac{1}{6}$, $w(q_1', q_2') = \frac{1}{2}$.

The following definitions, which follow immediately from the probabilistic simulations and bisimulations of [15, 21], are with respect to the concurrent probabilistic system $\mathcal{S} = (Q, q^0, \mathcal{L}, \Sigma, \mathit{Steps})$. We write $q \xrightarrow{\sigma,\nu}$ if there exists a transition $q \xrightarrow{\sigma,\nu} q'$ for some $q' \in Q$. A *simulation of* $\mathcal{S}$ is a relation $\mathcal{R} \subseteq Q \times Q$ such that, for each $(q_1, q_2) \in \mathcal{R}$:

1. $\mathcal{L}(q_1) = \mathcal{L}(q_2)$, and
2. if $q_1 \xrightarrow{\sigma,\nu_1}$, then $q_2 \xrightarrow{\sigma,\nu_2}$ for some distribution ν_2 such that $\nu_1 \mathcal{R} \nu_2$.

We say that q_2 *simulates* q_1, denoted by $q_1 \preceq q_2$, iff there exists a simulation which contains (q_1, q_2). A *bisimulation of* $\mathcal{S}$ is a simulation of $\mathcal{S}$ which is symmetric. Two states q_1, q_2 are called *bisimilar*, denoted by $q_1 \simeq q_2$, iff there exists a bisimulation which contains (q_1, q_2). As any simulation is a preorder, a bismulation is an equivalence relation. We can define simulation and bisimulation with respect to the composition of concurrent probabilistic systems in the standard manner [19, 5] in order to obtain a notion of relation between two such systems.

If an equivalence relation $\mathcal{R}$ on (a finite- or infinite-state) concurrent probabilistic system $\mathcal{S}$ contains a finite number of classes, we can define a finite-state *quotient concurrent probabilistic system* $\mathcal{S}_{\mathit{fin}}$, the states of which are the equivalence classes of $\mathcal{R}$, and the transitions of which are derived from those of $\mathcal{S}$, such that the initial state of $\mathcal{S}$ is related to the initial state of $\mathcal{S}_{\mathit{fin}}$ by $\mathcal{R}$. We omit details for reasons of space.

3.2 Probabilistic Temporal Logic

We now present a probabilistic temporal logic which can be used to specify properties of probabilistic hybrid automata. In brief, PBTL (Probabilistic Branching Time Logic) [6] is an extension of the temporal logic CTL in which the until operator includes a bound on probability. For example, the property of Section 1 regarding component failure is represented by the PBTL formula $[(\mathit{coolant})\forall\mathcal{U}(\mathit{alarm})]_{\geq 0.999}$, where *coolant* and *alarm* are atomic propositions labelling the appropriate states. Note that PBTL is essentially identical to the logics PCTL and pCTL presented in [9] and [4, 7] respectively, and that PBTL model checking of finite-state concurrent probabilistic systems may be performed using the algorithms of [7, 6]. The syntax of PBTL is defined as follows:

$$\Phi ::= \texttt{true} \;\mid\; a \;\mid\; \Phi \wedge \Phi \;\mid\; \neg\Phi \;\mid\; [\Phi \,\exists\mathcal{U}\, \Phi]_{\sqsupseteq\lambda} \;\mid\; [\Phi \,\forall\mathcal{U}\, \Phi]_{\sqsupseteq\lambda}$$

where $a \in \mathrm{AP}$, $\lambda \in [0,1]$, and $\sqsupseteq \in \{\geq, >\}$. The satisfaction relation for $\texttt{true}$, $a \in \mathrm{AP}$, $\wedge$ and $\neg$ are standard for temporal logic. In the following definition of the semantics of the probabilistic operators $[\Phi_1 \,\exists\mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda}$ and $[\Phi_1 \,\forall\mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda}$ we make use of the 'path formula' $\Phi_1 \,\mathcal{U}\, \Phi_2$, the interpretation of which is also standard; that is, $\Phi_1 \,\mathcal{U}\, \Phi_2$ is true of a path ω if and only if Φ_2 is true at some point along ω, and Φ_1 is satisfied at all preceding points (for a formal description, see, e.g., [6]). For a concurrent probabilistic system $\mathcal{S}$, a set $\mathcal{A}$ of adversaries on $\mathcal{S}$, and a state q of $\mathcal{S}$, the satisfaction relation for $q \models_{\mathcal{A}} [\Phi_1 \,\exists\mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda}$ is as follows:

$$q \models_{\mathcal{A}} [\Phi_1 \,\exists\mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda} \Leftrightarrow \mathit{Prob}^A(\{\omega \mid \omega \in \mathit{Path}^A_{\mathit{ful}}(q) \;\&\; \omega \models_{\mathcal{A}} \Phi_1 \,\mathcal{U}\, \Phi_2\}) \sqsupseteq \lambda$$
$$\text{for some adversary } A \in \mathcal{A}\,.$$

The semantics for $[\Phi_1 \,\forall\mathcal{U}\, \Phi_2]_{\sqsupseteq\lambda}$ is obtained by substituting "all adversaries $A \in \mathcal{A}$" for "some adversary $A \in \mathcal{A}$" in the above equivalence. The concurrent probabilistic system $\mathcal{S} = (Q, q^0, \mathcal{L}, \Sigma, \mathit{Steps})$ satisfies the PBTL formula Φ iff $q^0 \models_{\mathcal{A}} \Phi$.

We now introduce ∀PBTL as a fragment of PBTL involving only *universal* quantification over adversaries.The syntax of ∀PBTL is defined as follows:

$$\Phi ::= \texttt{true} \;\mid\; \texttt{false} \;\mid\; a \;\mid\; \neg a \;\mid\; \Phi \wedge \Phi \;\mid\; \Phi \vee \Phi \;\mid\; [\Phi \,\forall\, \mathcal{U}\, \Phi]_{\sqsupseteq \lambda}$$

where a, λ and $\sqsupseteq$ are as in the definition of PBTL. The following theorem states that simulation and bisimulation preserve certain (∀)PBTL formulae, and is inspired by a conjecture in [5]. The proof follows from similar results of Segala [21], which are defined for an action-based, rather than a state-based logic such as PBTL, and of Aziz et al. [4], which concern fully probabilistic systems (that is, Markov chains, or, equivalently, concurrent probabilistic systems for which $|Steps(q)| = 1$ for all states $q \in Q$).

Theorem 1. *Let $\mathcal{S}$ be a concurrent probabilistic system, a set $\mathcal{A}$ of adversaries of $\mathcal{S}$, let $\Phi_\forall, \Phi$ be formulae of ∀PBTL and PBTL respectively, and let $q_1, q_2 \in Q$.*

- *If $q_1 \preceq q_2$, then $q_2 \models_{\mathcal{A}} \Phi_\forall$ implies $q_1 \models_{\mathcal{A}} \Phi_\forall$.*
- *If $q_1 \simeq q_2$, then $q_1 \models_{\mathcal{A}} \Phi$ iff $q_2 \models_{\mathcal{A}} \Phi$.*

Naturally, for the two concurrent probabilistic systems $\mathcal{S}_1 = (Q_1, q_1^0, \mathcal{L}_1, \Sigma_1, Steps_1)$ and $\mathcal{S}_2 = (Q_2, q_2^0, \mathcal{L}_2, \Sigma_2, Steps_2)$, if $q_1^0 \preceq q_2^0$, then $\mathcal{S}_2 \models_{\mathcal{A}} \Phi_\forall$ implies $\mathcal{S}_1 \models_{\mathcal{A}} \Phi_\forall$. Similarly, if $q_1^0 \simeq q_2^0$, then $\mathcal{S}_1 \models_{\mathcal{A}} \Phi$ if and only if $\mathcal{S}_2 \models_{\mathcal{A}} \Phi$. Observe that Theorem 1 implies a decidable PBTL model checking procedure for any infinite-state concurrent probabilistic system with a finitary bisimilarity relation via reduction to the quotient concurrent probabilistic system, which can then be verified using the techniques of [7, 6].

3.3 Semantics of Probabilistic Hybrid Automata

The semantics of a given probabilistic hybrid automaton H can be represented in terms of a concurrent probabilistic system in the following way. The subsequent notation is used to reason about the target states of the probabilistic transitions of H. Let $\mathbf{a} \in \mathbb{R}^n$, $Z \in Rect(\mathcal{X})$ be a rectangle and $X \subseteq \mathcal{X}$. Then $\mathbf{a}[X := Z]$ denotes the set of valuations such that $\mathbf{a}' \in \mathbf{a}[X := Z]$ iff $\mathbf{a}' \in Z$ and $\mathbf{a}'_i = \mathbf{a}_i$ for all $x_i \in \mathcal{X} \setminus X$. Now, consider the valuation $\mathbf{a} \in \mathbb{R}^n$ and the m-vector $\langle \eta \rangle = [(Z^1, X^1), ..., (Z^m, X^m)]$, where, for each $j \in \{1, ..., m\}$, the set Z^j is a rectangle and $X^j \subseteq \mathcal{X}$ is a variable set. Then we generate the m-vector of valuations $\langle \mathbf{b} \rangle = [\mathbf{b}^1, ..., \mathbf{b}^m]$ in the following way: for each $j \in \{1, ..., m\}$, we choose a valuation $\mathbf{b}^j \in \mathbb{R}^n$ such that $\mathbf{b}^j \in \mathbf{a}[X^j := Z^j]$. Observe that, for any $i, j \in \{1, ..., m\}$ such that $i \neq j$, it may be the case that $\mathbf{a}[X^i := Z^i]$ and $\mathbf{a}[X^j := Z^j]$ have a non-empty intersection, and therefore it is possible that $\mathbf{b}^i = \mathbf{b}^j$. Let $\mathsf{Combinations}(\mathbf{a}, \langle \eta \rangle)$ be the set of all such vectors $\langle \mathbf{b} \rangle$. In the sequel, we use exclusively vectors of the form of $\langle \eta \rangle$ which comprise of post-conditions and variable sets in the support of a distribution. For the distribution μ, we let the vector $\mathsf{extract}(\mu) = [(post^1, X^1), ..., (post^m, X^m)]$ if $\mathsf{support}(\mu) = \{(w^1, post^1, X^1), ..., (w^m, post^m, X^m)\}$.

The (time-abstract) *concurrent probabilistic system* $\mathcal{S}_H = (Q_H, q_H^0, \mathcal{L}_H, \Sigma_H, Steps_H)$ *of the probabilistic hybrid automaton* $H = (\mathcal{X}, V, L, init, inv, flow, prob, \langle pre \rangle_{v \in V})$ is defined as follows:

- $Q_H \subseteq V \times \mathbb{R}^n$ such that $(v, \mathbf{a}) \in Q_H$ iff $\mathbf{a} \in \mathit{inv}(v)$;
- $q_H^0 \in Q_H$ such that $q_H^0 = (v, \mathit{init}(v))$ for $v \in V$ such that $\mathit{init}(v) \neq \emptyset$;
- for each $(v, \mathbf{a}) \in Q_H$, we have $\mathcal{L}(v, \mathbf{a}) = L(v)$;
- $\Sigma_H = \{\theta, \tau\}$;
- for each $(v, \mathbf{a}) \in Q_H$, we have $\mathit{Steps}_H(v, \mathbf{a}) = \mathit{Cts}_H(v, \mathbf{a}) \cup \mathit{Disc}_H(v, \mathbf{a})$, where:
  - for each $\delta \in \mathbb{R}_{\geq 0}$, there exists the pair $(\tau, \mathcal{D}(v, \mathbf{b})) \in \mathit{Cts}_H(v, \mathbf{a})$ iff $\mathbf{b} \in \mathit{inv}(v)$, and there exists a differentiable function $f : [0, \delta] \to \mathbb{R}^n$ with $\dot{f} : (0, \delta) \to \mathbb{R}^n$ such that $f(0) = \mathbf{a}$, $f(\delta) = \mathbf{b}$, and $\dot{f} \in \mathit{flow}(q, f(\epsilon))$ and $f(\epsilon) \in \mathit{inv}(v)$ for all $\epsilon \in (0, \delta)$;
  - for each $\mu \in \mathit{prob}(v)$, if $\mathbf{a} \in \mathit{pre}_v(p)$, then there exists the pair $(\theta, \nu_{\langle \mathbf{b} \rangle}) \in \mathit{Disc}_H(v, \mathbf{a})$, for each $\langle \mathbf{b} \rangle \in \mathsf{Combinations}(\mathbf{a}, \mathsf{extract}(\mu))$, iff there exists $\mu \in \mathit{prob}(v)$ such that:

$$\nu_{\langle \mathbf{b} \rangle}(w, \mathbf{c}) = \sum_{i \in \{1, \ldots, |\mathsf{support}(\mu)|\} \,\&\, \mathbf{c} = \mathbf{b}^i} \mu(w, \mathit{post}^i, X^i).$$

For a state $(v, \mathbf{a}) \in Q_H$, the definition of the continuous transitions in $\mathit{Cts}_H(v, \mathbf{a})$ is identical to the analogous definition for non-probabilistic hybrid automata, except that we require them to be made according to Dirac distributions (that is, with probability 1). The definition of the transitions in $\mathit{Disc}_H(v, \mathbf{a})$ reflects the intuition that H performs a control switch in the following manner: by (a) choosing an enabled distribution nondeterministically; (b) selecting a target mode and post-condition set probabilistically; and (c) choosing a successor state within the post-condition set nondeterministically. It is easy to verify that combining the two nondeterministic choices that comprise the first and third steps of the transition into a *single* nondeterministic selection, in the manner of the definition of $\mathcal{S}_H$, results in an equivalent transition. Naturally, if the post-condition set of at least one tuple in the support of a distribution μ is uncountable, then the set of vectors of the form $\langle \mathbf{b} \rangle$ associated with this set will also be uncountable, as will the set of transitions in $\mathit{Disc}_H(.)$ corresponding to μ. As a further note, observe that the definitions of (bi)simulation are applicable to concurrent probabilistic systems of probabilistic hybrid automata. Finally, a notion of *time divergence* can be associated with adversaries of the concurrent probabilistic systems of probabilistic hybrid automata; we omit details for reasons of space.

4 Model Checking Subclasses of Probabilistic Hybrid Automata

4.1 Probabilistic Multisingular and O-Minimal Hybrid Automata

The results of [1] and [18], which state the existence of finite bisimulation quotients of non-probabilistic multisingular and o-minimal hybrid automata respectively, can be extended to the probabilistic context in the following way. Firstly, the *region equivalence* of [2, 1] can be used to subdivide the infinitary state space of a probabilistic multisingular automaton M into a finite number of equivalence classes. Without loss of generality (see [3]), let all endpoints of rectangles used in the description of M be non-negative integers, with the maximal such integer denoted by c. For any $t \in \mathbb{R}$, let $\lfloor t \rfloor$ denote its integral part and $\mathit{frac}(t)$ its fractional part. For a vector $\mathbf{a}$, let $\lfloor \mathbf{a} \rfloor$ denote the vector whose ith coordinate is $\lfloor \mathbf{a}_i \rfloor$, and $\mathit{frac}(\mathbf{a})$ the vector whose ith coordinate is $\mathit{frac}(\mathbf{a}_i)$. For each mode $v \in V$, let $\langle \zeta^v \rangle = [\zeta_1^v, ..., \zeta_n^v]$ be the n-vector such that the ith element of $\langle \zeta^v \rangle$ is $\mathit{flow}(v)_i$ if $\mathit{flow}(v)_i \neq 0$, and is 1 otherwise. Let $\equiv^{\zeta^v}$ be the equivalence relation on $\mathbb{R}^n$ such that $\mathbf{a} \equiv^{\zeta^v} \mathbf{b}$ iff, for each $x_i, x_j \in \mathcal{X}$, (1) $\lfloor \zeta_i^v \mathbf{a}_i \rfloor = \lfloor \zeta_i^v \mathbf{b}_i \rfloor$, (2) $\mathit{frac}(\zeta_i^v \mathbf{a}_i) = \mathit{frac}(\zeta_i^v \mathbf{b}_i)$, and (3) $\mathit{frac}(\zeta_i^v \mathbf{a}_i) = \mathit{frac}(\zeta_j^v \mathbf{a}_j)$ iff $\mathit{frac}(\zeta_i^v \mathbf{b}_i) = \mathit{frac}(\zeta_j^v \mathbf{b}_j)$. Two states $(v, \mathbf{a})$ and $(w, \mathbf{b})$ are *region equivalent*, written $(v, \mathbf{a}) \equiv^R (w, \mathbf{b})$, if (1) $v = w$, (2) for each $x_i \in \mathcal{X}$, either $\lfloor \mathbf{a}_i \rfloor = \lfloor \mathbf{b}_i \rfloor$, or both $\mathbf{a}_i > c$ and $\mathbf{b}_i > c$, and (3) $\mathit{frac}(\mathbf{a}) \equiv^{\zeta^v} \mathit{frac}(\mathbf{b})$ (our notation is adapted from [10]). Intuitively, for each control mode $v \in V$, region equivalence subdivides $\mathbb{R}^n$ into a finite grid of unit hypercubes, which are in turn subdivided according to the flow gradients $\mathit{flow}(v, \cdot)_i = k_i$ of each $x_i \in \mathcal{X}$.

Lemma 1. *Let M be a probabilistic multisingular automaton. Region equivalence $\equiv^R$ is a finite bisimulation of $\mathcal{S}_M$.*

Proof. Clearly $\equiv^R$ has a finite number of equivalence classes; therefore, it remains to show that $\equiv^R$ is a bisimulation. The case for the continuous transitions in the sets $\mathit{Cts}_M(.)$ is similar to that in the non-probabilistic context, and therefore we concentrate on the discrete transitions in $\mathit{Disc}_M(.)$.

Observe that the set of valuations in a given region equivalence class is either contained within any rectangle Z used in the description of M, or is disjoint from Z. In particular, all valuations within such a class must be in the same pre-condition sets of M, and therefore enable the same distributions for choice. That is, if two states $(v, \mathbf{a}), (v, \mathbf{b}) \in Q_M$ are such that $(v, \mathbf{a}) \equiv^R (v, \mathbf{b})$, then, for any $\mu \in \mathit{prob}(v)$, we have $\mathbf{a} \in \mathit{pre}_v(\mu)$ if and only if $\mathbf{b} \in \mathit{pre}_v(\mu)$. Therefore, there exists an event-distribution pair $(\theta, \nu^1) \in \mathit{Steps}_M(v, \mathbf{a})$ if and only if there exists $(\theta, \nu^2) \in \mathit{Steps}_M(v, \mathbf{b})$, such that both ν^1 and ν^2 are derived from μ. Now we show that $\nu^1 \equiv^R \nu^2$. A standard fact is that, given $(v, \mathbf{a}) \equiv^R (v, \mathbf{b})$, for any tuple $(w, \mathit{post}, X) \in V \times 2^{\mathbb{R}^n} \times 2^{\mathcal{X}}$ such that post is a singleton, we have $(w, \mathbf{a}[X := \mathit{post}]) \equiv^R (w, \mathbf{b}[X := \mathit{post}])$. Furthermore, if the tuples $(w, \mathit{post}, X), (w, \mathit{post}', X') \in V \times 2^{\mathbb{R}^n} \times 2^{\mathcal{X}}$ are such that $(w, \mathbf{a}[X := \mathit{post}]) = (w, \mathbf{a}[X' := \mathit{post}'])$, then it must be the case that $(w, \mathbf{b}[X := \mathit{post}]) = (w, \mathbf{b}[X' := \mathit{post}'])$. The combination of these facts then gives us that, for $(w, \mathbf{c}), (w, \mathbf{d}) \in Q_M$ which are such that $(w, \mathbf{c}) \equiv^R (w, \mathbf{d})$:

$$\nu^1(w, \mathbf{c}) = \sum_{\substack{i \in \{1,\ldots,k\} \\ \& \mathbf{c} = \mathbf{a}[X^i := \mathit{post}^i]}} \mu(w, \mathit{post}^i, X^i) = \sum_{\substack{i \in \{1,\ldots,k\} \\ \& \mathbf{d} = \mathbf{b}[X^i := \mathit{post}^i]}} \mu(w, \mathit{post}^i, X^i) = \nu^2(w, \mathbf{d}),$$

where $k = |\mathsf{support}(\mu)|$. The fact that $\nu^1 \equiv^R \nu^2$ then follows. We can repeat this process for all region equivalence classes, and all distributions enabled in these classes, to conclude that $\equiv^R$ satisfies the properties of bisimulation. $\square$

Secondly, we show that the model checking results for o-minimal hybrid automata of [18, 3] transfer to the probabilistic context. Observe that the previous decidability results for this class necessitate the *decoupling* of discrete and continuous behaviour of the hybrid automata; more precisely, all variables are reset to a new value at every discrete transition. The result of [18] then shows that, for each control mode $v \in V$, the associated continuous state space of v has a finite bisimulation quotient. This quotient is obtained after an initial subdivision of the continuous space of v according to the invariant set of v, the pre-condition sets of all the outgoing discrete transitions of v, and the post-condition sets of all incoming discrete transitions of v. Similarly, in our context, we require that such an initial subdivision is made according to $pre_v(\mu)$, for all $\mu \in prob(v)$, in addition to $inv(v)$ and all post-condition sets $post$ appearing in tuples of the form $(v, post, \mathcal{X}) \in \mathsf{support}(\mu')$, for all $\mu' \in prob(v')$ and $v' \in V$.

Consider the probabilistic o-minimal hybrid automaton O, and let $(v, \mathbf{a})$, $(v, \mathbf{b}) \in Q_O$ be two states such that $\mathbf{a}, \mathbf{b} \in pre_v(\mu)$ for some $\mu \in prob(v)$. Because the reset set for all tuples $(w, post, \mathcal{X}) \in \mathsf{support}(\mu)$ is the full variable set $\mathcal{X}$, it follows that $\mathsf{Combinations}(\mathbf{a}, \mathsf{extract}(\mu)) = \mathsf{Combinations}(\mathbf{b}, \mathsf{extract}(\mu))$. Intuitively, given that the distribution μ is enabled in $(v, \mathbf{a})$ and $(v, \mathbf{b})$, the distinction between the valuations $\mathbf{a}$ and $\mathbf{b}$ is lost after taking μ. Therefore, the sets of concurrent probabilistic system distributions corresponding to the choice of μ is the same for $(v, \mathbf{a})$ and $(v, \mathbf{b})$. Now consider the case in which $(v, \mathbf{a})$ and $(v, \mathbf{b})$ lie in the intersection of the pre-condition sets of multiple distributions $\mu_1, ..., \mu_l \in prob(v)$, where $l \in \{1, ..., |prob(v)|\}$. Then, extending our intuition from the single distribution μ to the set $\{\mu_1, ..., \mu_l\}$, we have the strong characteristic that $Disc_O(v, \mathbf{a}) = Disc_O(v, \mathbf{b})$. Such intersections of pre-conditions are further subdivided with respect to continuous transitions using the methodology of [18]. Given that $(v, \mathbf{a})$ and $(v, \mathbf{b})$ lie in the same portion of the state space according to this subdivision, we conclude that $(v, \mathbf{a})$ and $(v, \mathbf{b})$ are bisimilar.

Lemma 2. *Let O be a probabilistic o-minimal hybrid automata. O has a finite bisimulation quotient.*

Corollary 1. *The PBTL model checking problems for probabilistic multisingular automata and probabilistic o-minimal hybrid automata are decidable.*

4.2 Probabilistic Rectangular Automata

We now introduce a model checking strategy for initialised probabilistic rectangular automata, based on similar results in the non-probabilistic context of [20, 12]. From an initialised probabilistic rectangular automaton R, we construct a probabilistic multisingular automaton M_R, such that M_R is a *sufficient abstraction* of R which can subsequently be verified. More precisely, each variable $x_i \in \mathcal{X}$ of R is represented by two variables $y_{l(i)}, y_{u(i)} \in \mathcal{Y}$ of M_R, with the intuition that $y_{l(i)}$ tracks the least possible value of x_i, whereas $y_{u(i)}$ tracks its greatest possible value. Therefore, singular flow conditions for $y_{l(i)}$ (respectively, $y_{u(i)}$) are derived from the minimal (respectively, maximal) slopes that x_i may take in R. Furthermore, the probabilistic edge relation of M_R updates $y_{l(i)}$ and $y_{u(i)}$ so that the interval $[y_{l(i)}, y_{u(i)}]$ represents the possible values of x_i. For example, consider Figure 2(a); say the current control mode is v, and that the flow

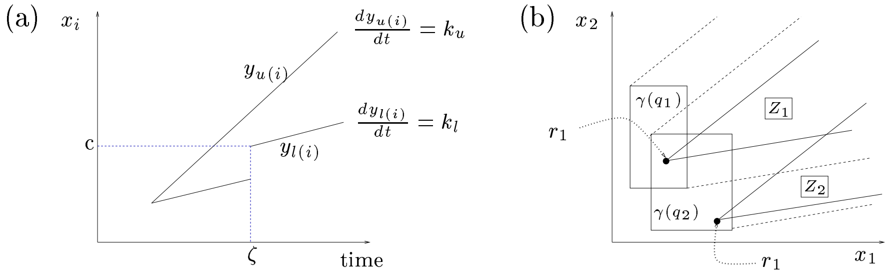

Fig. 2. (a) Updating the value of x_i. (b) M_R simulates R.

condition $\mathit{flow}(v)_i$ of the variable x_i is the rectangle $[k_l, k_u]$. As time passes, the possible values of x_i are contained within an envelope, the lower (respectively, upper) bound of which is represented by $y_{l(i)}$ (respectively, $y_{u(i)}$). If a control switch occurs, the values of $y_{l(i)}$ and $y_{u(i)}$ must continue to represent the possible values of x_i; this may involve resetting $y_{l(i)}$ or $y_{u(i)}$, or both, even if x_i is not reset by the corresponding control switch. In Figure 2(a), at time ζ a distribution μ is chosen, where $\mathit{pre}_v(\mu) = [c, \infty)$, and say a tuple $(w, \mathit{post}, X) \in \mathsf{support}(\mu)$ is probabilistically selected for which $x_i \notin X$. Then, for $y_{l(i)}$ to correctly represent the lower bound on x, it must be updated to c when emulating this control switch, as its value is below c when the distribution μ was selected. This reflects the standard intuition in the non-probabilistic case of [12].

Let $R = (\mathcal{X}, V, L, \mathit{init}^R, \mathit{inv}^R, \mathit{flow}^R, \mathit{prob}^R, \langle \mathit{pre}_v^R \rangle_{v \in V})$ be an initialised probabilistic rectangular automaton subject to the following simplifying assumptions. For all control modes $v \in V$, we have $\mathit{inv}^R(v) = \mathbb{R}^n$, and the rectangle $\mathit{flow}^R(v, \cdot)$ is compact; for all $\mu \in \mathit{prob}^R(v)$, the rectangle $\mathit{pre}_v^R(\mu)$ is compact, and for each $(w, \mathit{post}, X) \in \mathsf{support}(\mu)$, the rectangle post is compact. [2] Then $M_R = (\mathcal{Y}, V, L, \mathit{init}^{M_R}, \mathit{inv}^{M_R}, \mathit{flow}^{M_R}, \mathit{prob}^{M_R}, \langle \mathit{pre}_v^{M_R} \rangle_{v \in V})$ is the probabilistic multisingular automaton constructed in the following way.

Variables $\mathcal{Y} = \{y_1, ..., y_{2n}\}$, where the $l(i)$-th variable $y_{l(i)}$ represents the lower bound on the i-th variable $x_i \in \mathcal{X}$ of R, the $u(i)$-th variable $y_{u(i)}$ represents the upper bound on $x_i \in \mathcal{X}$, and $l(i) = 2i - 1$, $u(i) = 2i$.

Initial and invariant sets. For each control mode $v \in V$ and each $x_i \in \mathcal{X}$, we have $\mathit{init}^{M_R}(v)_{l(i)} = \mathit{init}^{M_R}(v)_{u(i)} = \mathit{init}^R(v)_i$, and $\mathit{inv}^{M_R}(v) = \mathbb{R}^{2n}$.

Flow inclusion. For each control mode $v \in V$ and $x_i \in \mathcal{X}$, if $\mathit{flow}^R(v, \cdot)_i = [l, u]$ then $\mathit{flow}^{M_R}(v, \cdot)_{l(i)} = l$ and $\mathit{flow}^{M_R}(v, \cdot)_{u(i)} = u$.

Probability distributions. For each control mode $v \in V$ and $\mu^R \in \mathit{prob}^R(v)$, there exists a corresponding set $\{\mu_1^{M_R}, ..., \mu_4^{M_R}\} \subseteq \mathit{prob}^{M_R}(v)$, which takes the following form. For each tuple $(w, \mathit{post}^R, X) \in \mathsf{support}(\mu^R)$, and for all $j \in \{1, ..., 4\}$, a corresponding tuple $(w, \mathit{post}_j^{M_R}, Y_j) \in \mathsf{support}(\mu_j^{M_R})$ exists,

[2] Note that it follows from [12] that *all* of these assumptions may be dropped; however, in such a case, the construction of M_R requires significant book-work that is independent of probabilistic concerns.

such that $\mu_j^{M_R}(w, post_j^{M_R}, Y_j) = \mu^R(w, post_j^R, X)$. Let $[l_i, u_i] = pre_v^R(\mu^R)_i$ and $[l'_i, u'_i] = (post^R)_i$ for each $i \in \{1, ..., n\}$. If $x_i \in X$, then $(post_j^{M_R})_{l(i)} = l'_i$, $(post_j^{M_R})_{u(i)} = u'_i$, and $y_{l(i)}, y_{u(i)} \in Y_j$ for each $j \in \{1, ..., 4\}$. However, if $x_i \notin X$, then $(post_j^{M_R})_{l(i)}$, $(post_j^{M_R})_{u(i)}$, and Y_j are defined as follows:

$$\begin{array}{lll}
(post_1^{M_R})_{l(i)} = l_i, & (post_1^{M_R})_{u(i)} = u'_i, & y_{l(i)} \in Y_1; \\
(post_2^{M_R})_{l(i)} = l_i, & (post_2^{M_R})_{u(i)} = u_i, & y_{l(i)}, y_{u(i)} \in Y_2; \\
(post_3^{M_R})_{l(i)} = l'_i, & (post_3^{M_R})_{u(i)} = u'_i; & \text{no requirement on } Y_3; \\
(post_4^{M_R})_{l(i)} = l'_i, & (post_4^{M_R})_{u(i)} = u_i, & y_{u(i)} \in Y_4.
\end{array}$$

Pre-condition sets. For every $v \in V$ and $\mu^R \in prob^R(v)$, we define the pre-condition sets for $\{\mu_1^{M_R}, ..., \mu_4^{M_R}\} \subseteq prob^{M_R}(v)$ in the following way. For every $i \in \{1, ..., n\}$, if $pre_v^R(\mu^R)_i = [l, u]$, let:

$$\begin{array}{ll}
pre_v^{M_R}(\mu_1^{M_R})_{l(i)} = (-\infty, l), & pre_v^{M_R}(\mu_1^{M_R})_{u(i)} = [l, u]; \\
pre_v^{M_R}(\mu_2^{M_R})_{l(i)} = (-\infty, l), & pre_v^{M_R}(\mu_2^{M_R})_{u(i)} = (u, \infty); \\
pre_v^{M_R}(\mu_3^{M_R})_{l(i)} = \ [l, u], & pre_v^{M_R}(\mu_3^{M_R})_{u(i)} = [l, u]; \\
pre_v^{M_R}(\mu_4^{M_R})_{l(i)} = \ [l, u], & pre_v^{M_R}(\mu_4^{M_R})_{u(i)} = (u, \infty).
\end{array}$$

The function $\gamma : Q_{M_R} \to 2^{Q_R}$ of [12], where $\gamma(v, \mathbf{a}) = \{v\} \times \Pi_{i=1}^n [\mathbf{a}_{l(i)}, \mathbf{a}_{u(i)}]$, can also be used in the probabilistic context to relate a set of states of R to a state of M_R. We now propose a strategy for model checking R via the construction of M_R. More precisely, R is simulated by M_R (that is, the initial state of R is simulated by the initial state of M_R), and, by Theorem 1, if a $\forall$PBTL formula Φ is satisfied by M_R, then this is sufficient for concluding that Φ is also satisfied by R. Observe that the simulation is obtained by viewing γ as a relation.

Lemma 3. *Let R be a probabilistic rectangular automaton, and M_R be the probabilistic multisingular automaton constructed from R. Let $q \in Q_{M_R}$ be a state of $\mathcal{S}_{M_R}$, and let $r \in \gamma(q)$ be a state of $\mathcal{S}_R$. Then $r \preceq q$.*

We omit the proof of Lemma 3 for reasons of space. An example of the way in which M_R forward simulates R is shown in Figure 2(b). Let the variable set of R contain two variables, x_1 and x_2. Consider a state $r \in Q_R$ of R, and a state $q \in Q_{M_R}$ of M_R which are such that $r \in \gamma(q)$. Say R nondeterministically selects an enabled distribution μ^R for choice, which, by construction of M_R, can be matched in q by one (and because the pre-conditions of each distribution are disjoint, only one) of the four distributions $\{\mu_1^{M_R}, ..., \mu_4^{M_R}\}$; let $\mu_j^{M_R}$ be this distribution, for some $j \in \{1, ..., 4\}$. Let $\mu^R(w, post_1^R, X_1) = \frac{1}{3}$, $\mu^R(w, post_2^R, X_2) = \frac{1}{6}$, and $\mu^R(w, post_3^R, X_3) = \frac{1}{2}$. From the construction of M_R, $\mu_j^{M_R}(w, post_{j,1}^{M_R}, Y_1) = \frac{1}{3}$, $\mu_j^{M_R}(w, post_{j,2}^{M_R}, Y_2) = \frac{1}{6}$, $\mu_j^{M_R}(w, post_{j,3}^{M_R}, Y_3) = \frac{1}{2}$. Say $post_2^R, X_2$ and $post_3^R, X_3$ are such that, when applied to r, they result in the same target sets of states. Then the maximal and minimal values of x_1, x_2 encoded in M_R will be the same for $post_2^R, X_2$ and $post_3^R, X_3$, and therefore the probability of M_R making a transition to the state q_2, which encodes these values, will be $\nu^{M_R}(q_2) = \mu_j^{M_R}(w, post_{j,2}^{M_R}, Y_2) + \mu_j^{M_R}(w, post_{j,3}^{M_R}, Y_3) = \frac{1}{6} + \frac{1}{2} =$

$\frac{2}{3}$. We encode the maximal and minimal values reached by M_R from q after the probabilistic choice of $(w, post_{j,1}^{M_R}, Y_1)$ to be q_1; therefore, $\nu^{M_R}(q_1) = \mu_j^{M_R}(w, post_{j,1}^{M_R}, Y_1) = \frac{1}{3}$. Say the rectangular sets encoded by q_1 and q_2 via γ overlap (see Figure 2(b)). Consider the case in which, after probabilistically choosing either of the tuples $(w, post_1^R, X_1)$ and $(w, post_2^R, X_2)$, the *same* target state r_1 is selected by R, which, naturally, is in the intersection of the state sets defined by $\gamma(q_1)$ and $\gamma(q_2)$. We let the choice of target state after the probabilistic choice of $(w, post_3^R, X_3)$ to be r_2, which is in $\gamma(q_2)$ but not $\gamma(q_1)$. Then, from our view of γ as inducing the simulation, we have $r_1 \preceq q_1$ and $r_1, r_2 \preceq q_2$. Say ν^R is the distribution of $\mathcal{S}_R$ corresponding to the states r_1 and r_2; then $\nu^R(r_1) = \mu^R(w, post_1^R, X_1) + \mu^R(w, post_2^R, X_2) = \frac{1}{3} + \frac{1}{6} = \frac{1}{2}$, and $\nu^R(r_2) = \mu^R(w, post_3^R, X_3) = \frac{1}{2}$. To show that $\nu^R \preceq \nu^{M_R}$, the weight function w, which relates ν^R and ν^{M_R} via $\preceq$, is defined: let $w(r_1, q_1) = \frac{1}{3}, w(r_1, q_2) = \frac{1}{6}$, $w(r_2, q_2) = \frac{1}{2}$. Note that the weight function is obtained from the probabilities assigned by μ^R to tuples in its support; this fact can be used to derive a weight function for any probabilistic transition of R and M_R.

The argument that γ induces a simulation with regard to *continuous* transitions follows from the non-probabilistic precedent of [20, 12]. In Figure 2(b), both q_1 and q_2 can simulate all of the transitions from any state lying in the intersection $\gamma(q_1) \cap \gamma(q_2)$, such as r_1. For example, the distribution with the pre-condition Z_1 is enabled in r_1, q_1 and q_2, after some time has elapsed. However, as r_2 is in $\gamma(q_2)$ but not $\gamma(q_1)$, the state q_2, but not q_1, can simulate the choice of the distribution with the pre-condition Z_2 by r_2.

Proposition 1. *Let R, M_R be defined as in Lemma 3. For any $\forall$PBTL formula Φ, if $\mathcal{S}_{M_R} \models_{\mathcal{A}} \Phi$ then $\mathcal{S}_R \models_{\mathcal{A}} \Phi$.*

5 Conclusions

Model checking for hybrid systems is well known to be expensive, and the strategies presented in this paper are no exception. For multisingular automata, the size of the region quotient is exponential in the number of variables used and the magnitude of the upper bounds used in the description of the sets of the model. Furthermore, the verification algorithm for PBTL [7, 6] is polynomial in the size of this quotient and linear in the size of the formula. Therefore, further work could address the inefficiencies of this method, for example exploiting model checking methods of [13] for rectangular automata. Formalisms which admit continuous probabilistic behaviour, such as stochastic hybrid systems [14], are also of interest, and could be subject to a variant of the model checking technique for timed automata with continuously distributed delays of [17].

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. *Theoretical Computer Science*, 138:3–34, 1995.

2. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
3. R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. To appear in *Proceedings of the IEEE*, 2000.
4. A. Aziz, V. Singhal, F. Balarin, R. Brayton, and A. Sangiovanni-Vincentelli. It usually works: the temporal logic of stochastic systems. In *Proc. 7th CAV*, volume 939 of *Lecture Notes in Computer Science*, pages 155–165. Springer-Verlag, 1995.
5. C. Baier. On algorithmic verification methods for probabilistic systems, 1998. Habilitation thesis, University of Mannheim.
6. C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11:125–155, 1998.
7. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. FST&TCS'95*, volume 1026 of *LNCS*, pages 499–513. Springer-Verlag, 1995.
8. L. de Alfaro, M. Kwiatkowska, G. Norman, D. Parker, and R. Segala. Symbolic model checking of concurrent probabilistic processes using MTBDDs and the Kronecker representation. In *Proc. TACAS'00*, volume 1785 of *LNCS*, pages 395–410. Springer-Verlag, 2000.
9. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
10. T. A. Henzinger, B. Horowitz, and R. Majumdar. Rectangular hybrid games. In *Proc. CONCUR'99*, volume 1664 of *LNCS*, pages 320–335. Springer-Verlag, 1999.
11. T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond HyTech: hybrid systems analysis using interval numerical methods. In *Proc. HSCC'00*, volume 1790 of *LNCS*, pages 130–144. Springer-Verlag, 2000.
12. T. A. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, 1998.
13. T. A. Henzinger and R. Majumdar. Symbolic model checking for rectangular hybrid systems. In *Proc. TACAS'00*, volume 1785 of *LNCS*, pages 142–156. Springer-Verlag, 2000.
14. J. Hu, J. Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. In *Proc. HSCC'00*, volume 1790 of *LNCS*. Springer-Verlag, 2000.
15. B. Jonsson and K. G. Larsen. Specification and refinement of probabilistic processes. In *Proc. 6th LICS*, pages 266–279. IEEE Computer Society Press, 1991.
16. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. To appear in *Theoretical Computer Science*, special issue on *ARTS'99: Formal Methods for Real-time and Probabilistic Systems*, 2000.
17. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In *Proc. CONCUR'00*, *LNCS*. Springer-Verlag, 2000.
18. G. Lafferriere, G. Pappas, and S. Yovine. A new class of decidable hybrid systems. In *Proc. HSCC'99*, volume 1569 of *LNCS*, pages 137–151. Springer-Verlag, 1999.
19. R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.
20. A. Olivero, J. Sifakis, and S. Yovine. Using abstractions for the verification of linear hybrid systems. In *Proc. 6th CAV*, volume 818 of *LNCS*, pages 81–94. Springer-Verlag, 1994.
21. R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.