Electronic Communications of the EASST
Volume 70 (2014)

Proceedings of the
14th International Workshop on
Automated Verification of Critical Systems (AVoCS 2014)

Exact and Approximate Abstraction for Classes of Stochastic Hybrid Systems

Jeremy Sproston

15 pages

Guest Editors: Marieke Huisman, Jaco van de Pol
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Exact and Approximate Abstraction for Classes of Stochastic Hybrid Systems

Jeremy Sproston[*]

Dipartimento di Informatica, University of Turin, Italy

Abstract: A stochastic hybrid system contains a collection of interacting discrete and continuous components, subject to random behaviour. The formal verification of a stochastic hybrid system often comprises a method for the generation of a finite-state probabilistic system which either represents exactly the behaviour of the stochastic hybrid system, or which approximates conservatively its behaviour. We extend such abstraction-based formal verification of stochastic hybrid systems in two ways. Firstly, we generalise previous results by showing how bisimulation-based abstractions of non-probabilistic hybrid automata can be lifted to the setting of probabilistic hybrid automata, a subclass of stochastic hybrid systems in which probabilistic choices can be made with respect to finite, discrete alternatives only. Secondly, we consider the problem of obtaining approximate abstractions for discrete-time stochastic systems in which there are continuous probabilistic choices with regard to the slopes of certain system variables. We restrict our attention to the subclass of such systems in which the approximate abstraction of such a system, obtained using the previously developed techniques of Fränzle et al., results in a probabilistic rectangular hybrid automaton, from which in turn a finite-state probabilistic system can be obtained. We illustrate this technique with an example, using the probabilistic model checking tool PRISM.

Keywords: Probabilistic verification, stochastic hybrid systems, bisimulation

1 Introduction

Our increasing reliance on complex embedded and cyber-physical systems calls for the development of methods for the verification of hybrid systems, which are systems in which behaviour is described as an interplay between discrete and continuous components. In this paper, we consider a well-known formalism for the description of hybrid systems, namely hybrid automata [ACH$^+$95], which comprise a finite-state graph, to represent the discrete part of the system, and a finite set of real-valued variables, to represent the continuous part of the system. Interaction between the discrete and continuous parts of the system is represented by labelling the graph with conditions on the variables and their first derivatives. Hence we can express that, as time passes and the system resides in a particular node of the graph, the rate of change of the variables is described in a certain way; we can also describe how the system moves from node to node when the

[*] Supported by the project AMALFI (University of Turin/Compagnia di San Paolo), the MIUR-PRIN project CINA, and the EU ARTEMIS Joint Undertaking under grant agreement no. 332933 (HoliDes).

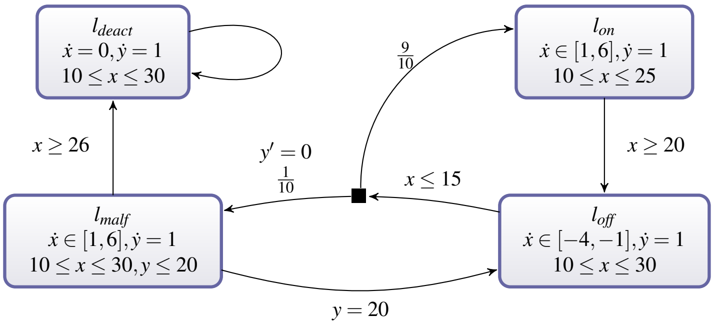

Figure 1: A probabilistic hybrid automaton modelling a faulty thermostat

value of the variables satisfies certain conditions. Hybrid automata can be subject to automatic verification procedures, which have been implemented in a number of tools [HHW97, Fre08].

In many contexts, a representation of a hybrid system without information regarding the likelihood of its various behaviours is insufficient. For example, a verification method for a particular hybrid system may show the existence of a behaviour corresponding to an error or to a decrease in performance, but without representing the fact that the behaviour is unlikely to occur. This observation has led to a number of formalisms for stochastic hybrid systems, in which the likelihood of behaviours is represented, for example [HLS00, Spr00, Buj04, APLS08, FHH+11, Hah13]. In particular, probabilistic hybrid automata [Spr00] extend hybrid automata by associating probabilities with edges of the graph. Stochastic hybrid automata [FHH+11, Hah13] extend probabilistic hybrid automata by allowing the values of the continuous variables to be reset according to continuous probability distributions.

An example of a probabilistic hybrid automaton modelling a faulty thermostat is shown in Figure 1 (the model is a modification of an example of [Spr11]). We use a number of the usual conventions for illustrating hybrid automata, and refer to the nodes of the graph as *locations*. The ambient temperature is represented by the variable x, and variable y is a timer. When the heater is on (location l_{on} or location l_{malf}), the temperature increases at a rate between 1 and 6; when the heater is off (location l_{off}), the temperature changes at a rate between -4 and -1. The locations l_{on} and l_{off} corresponds to non-faulty behaviour, whereas the location l_{malf} corresponds to the heater being on in the presence of a fault in the temperature sensor that means that the measurement of the temperature is temporarily unavailable. The system passes from l_{on} to l_{off}, with probability 1, when the temperature is between 20 and 25, and from l_{off} to l_{on}, with probability $\frac{9}{10}$, or to l_{malf}, with probability $\frac{1}{10}$, when the temperature is between 10 and 15. The sensor fault means that the temperature can increase to a higher level in l_{malf} 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 20 time units after the location l_{malf} was entered, in which case the heater is switched off. All edges of the probabilistic hybrid automaton correspond to reaching a certain location with probability 1, apart from the probabilistically branching edge from l_{off}.

Two main approaches for the verification of probabilistic and stochastic hybrid automata have been presented. On the one hand, *exact* methods generally consider the construction of a finite-state probabilistic system (more precisely, a probabilistic automaton or a Markov decision process [Put94]) that represents faithfully the behaviour of the original system. This approach has been adopted for restricted subclasses of probabilistic hybrid automata in [Spr00, Spr11] using probabilistic bisimulation [LS91, SL95] to obtain a finite number of equivalence classes from which a finite-state system can be constructed. On the other hand *approximate* methods consider generally the construction of a finite-state probabilistic system that overapproximates the behaviour of the original system. This approach has been adopted for probabilistic hybrid automata in [Spr01, ZSR$^+$12], and for stochastic hybrid automata in [FHH$^+$11, Hah13].

The contributions of this paper are twofold. First, in Section 2, we generalise the results of [Spr00, Spr11] concerning exact verification methods. Any probabilistic hybrid automaton can be translated into a non-probabilistic hybrid automaton in which information concerning probability distributions is encoded in labels on edges of the graph. Consider a probabilistic hybrid automaton $\mathscr{H}$: we show that if the non-probabilistic hybrid automaton counterpart of $\mathscr{H}$ has a finite bisimulation equivalence quotient, then $\mathscr{H}$ has a finite probabilistic bisimulation equivalence quotient. This result unifies and generalises previous results, and has the consequence that we can identify classes of probabilistic hybrid automata with a finite bisimulation equivalence quotient on the basis of whether members of the corresponding class of hybrid automata have finite bisimulation quotients. For example, we can conclude that the class of probabilistic STORMED hybrid automata, which are probabilistic hybrid automata to which the restrictions of STORMED hybrid automata [VPVD08] apply to the non-probabilistic characteristics of the system, have a finite probabilistic bisimulation quotient, because STORMED hybrid automata have finite bisimulation equivalence quotients. Any future results on the identification of classes of hybrid automata with finite bisimulation quotients will also imply that the corresponding class of probabilistic hybrid automata has finite probabilistic bisimulation quotients.

The second contribution, in Section 3, concerns approximate verification methods. In stochastic hybrid automata, the rate of change of continuous variables as time passes is not (directly) chosen probabilistically. However, the rate of change of a continuous variable x may be equal to the value of another continuous variable y (that is, $\dot{x} = y$), and y could be subject to a reset by a continuous probability distribution: hence, the rate of change of x is indirectly dependent on a continuous probabilistic choice. We consider this approach in the context of stochastic hybrid automata and the approximation method of [FHH$^+$11, Hah13], which uses probabilistic hybrid automata as approximate models of stochastic hybrid automata. We show that there exists a class of probabilistic hybrid automata that features dependence of variable's rate of change on the value of other variables (where these variables are constant as time passes) that is equivalent to probabilistic rectangular automata, a subclass of probabilistic hybrid automata which has a finite probabilistic bisimulation equivalence quotient when considering a discrete-time semantics [Spr11]. We apply this approach to the case of the faulty thermostat of Figure 1: for example, when the heater is on, rather than increase nondeterministically with a rate in $[1,6]$, we consider that the rate of increase of the temperature is chosen from the normal distribution with mean 3.5 and standard deviation 1, truncated to the interval $[1,6]$. This continuous distribution is approximated according to the approach of [FHH$^+$11, Hah13], which then results in an intermediate probabilistic hybrid automaton that we can show is equivalent to a probabilistic rectangular automaton, which (assuming a discrete-time semantics) can then be transformed into a finite-state probabilistic system, which is then analysed using the probabilistic model checking tool PRISM [KNP11]. We show that there is a significant difference in the results obtained from the probabilistic rectangular automaton in Figure 1 and from the probabilistic rectangular automaton obtained as an approximation of an automaton involving continuous distributions, as described above.

2 Exact Abstraction of Probabilistic Hybrid Automata

2.1 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, $\mathbb{Z}$ to denote the set of integers, $\mathbb{N}$ to denote the set of natural numbers and $\mathbb{Q}$ to denote the set of rational numbers. 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 $\{q_1 \mapsto \lambda_1, \dots, q_n \mapsto \lambda_n\}$ to denote a distribution μ for which $\mu(q_1) = \lambda_1$, ..., $\mu(q_n) = \lambda_n$. Given a set $Q' \subseteq Q$, we let $\mu[Q'] = \sum_{q \in Q'} \mu(q)$.

A *probabilistic automaton* (PA) $\mathscr{P} = (S, Act, \Rightarrow)$ comprises the following components: a set of states S; a set of actions Act; and a probabilistic, nondeterministic transition relation $\Rightarrow \subseteq S \times Act \times \mathsf{Dist}(S)$. Each of S, Act and $\Rightarrow$ may be uncountable.

An *infinite path* of a PA is an infinite sequence $r = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \cdots$ such that $(s_i, a_i, \mu_i) \in \Rightarrow$ and $\mu_i(s_{i+1}) > 0$ for each $i \in \mathbb{N}$. Similarly, a *finite path* of a PA is a finite sequence $r = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \cdots a_{n-1} \mu_{n-1} s_n$ such that $(s_i, a_i, \mu_i) \in \Rightarrow$ and $\mu_i(s_{i+1}) > 0$ for each $i < n$. We use $Path_{ful}^{\mathscr{P}}$ to denote the set of infinite paths of $\mathscr{P}$, and $Path_{fin}^{\mathscr{P}}$ to denote the set of finite paths of $\mathscr{P}$. When clear from the context we omit the superscript $\mathscr{P}$. If r is a finite path, we denote by $last(r)$ the last state of r. Let $Path_{ful}^{\mathscr{P}}(s)$ and $Path_{fin}^{\mathscr{P}}(s)$ refer to the sets of infinite and finite paths of $\mathscr{P}$, respectively, commencing in state $s \in S$.

Let $\mathscr{P} = (S, Act, \Rightarrow)$ be a PA. A *strategy* of $\mathscr{P}$ is a function σ mapping every finite path $r \in Path_{fin}$ to a transition $(last(r), a, \mu) \in \Rightarrow$. We write as $\Sigma_{\mathscr{P}}$ the set of strategies of $\mathscr{P}$. For any strategy σ, let $Path_{ful}^{\sigma}$ and $Path_{fin}^{\sigma}$ denote the sets of infinite and finite paths, respectively, resulting from the choices of σ. For a state $s \in S$, let $Path_{ful}^{\sigma}(s) = Path_{ful}^{\sigma} \cap Path_{ful}(s)$ and $Path_{fin}^{\sigma}(s) = Path_{fin}^{\sigma} \cap Path_{fin}(s)$. Given a strategy $\sigma \in \Sigma_{\mathscr{P}}$ and a state $s \in S$, we define the probability measure $Prob_s^{\sigma}$ over $Path_{ful}^{\sigma}(s)$ in the standard way [KSK76]. Note that 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 map from finite paths r to $\mathsf{Dist}(\Rightarrow)$) are considered will be signalled in the text.

Given state set $S' \subseteq S$, we write $Reach(S')$ for the set of paths that reach S', that is $Reach(S') = \{r \mid r \in Path_{ful} \land \exists i \in \mathbb{N}. r(i) \in S'\}$ where $r(i)$ is the $(i+1)$-th state along r. Then, given state $s \in S$, we write

$$MaxReach_s^{\mathscr{P}}(S') = \sup_{\sigma \in \Sigma_{\mathscr{P}}} Prob_s^{\sigma}(Reach(S')) \ , \ MinReach_s^{\mathscr{P}}(S') = \inf_{\sigma \in \Sigma_{\mathscr{P}}} Prob_s^{\sigma}(Reach(S'))$$

to be the maximum and minimum probability of reaching a state in S' from s, respectively.

Let $\mathscr{P} = (S, Act, \Rightarrow)$ be a PA. For distributions $\mu, \nu \in \mathsf{Dist}(S)$ and for an equivalence relation $\equiv \subseteq S \times S$, we denote by $\mu \equiv \nu$ the condition that, for each equivalence class C of $\equiv$, the equality $\mu[C] = \nu[C]$ holds. Let $\equiv \subseteq S \times S$ be an equivalence relation on S. A *probabilistic bisimulation respecting* $\equiv$ on $\mathscr{P}$ [LS91, SL95] is an equivalence relation $\simeq \subseteq S \times S$ such that $s \simeq t$ implies that (1) $s \equiv t$, and (2) if $(s, a, \mu) \in \Rightarrow$, then there exists $(t, a, \nu) \in \Rightarrow$ such that $\mu \simeq \nu$. States s and t are called probabilistically bisimilar with respect to $\equiv$ in $\mathscr{P}$ if there exists a probabilistic bisimulation $\simeq$ respecting $\equiv$ such that $s \simeq t$. Probabilistic bisimilar states exhibit the same maximum and minimum probabilities of reachability (or, more generally, ω-regular properties [BK08]), and satisfy the same properties of the probabilistic temporal logic PCTL* [SL95].

A PA $\mathscr{P} = (S, Act, \Rightarrow)$ for which all transitions $(s, a, \mu) \in \Rightarrow$ are such that μ is of the form $\{s' \mapsto 1\}$ for some $s' \in S$ is called a *nondeterministic automaton*. In the case of nondeterministic automata, we often write (s, a, s') to denote the transition $(s, a, \{s' \mapsto 1\})$. If $\mathscr{P}$ is a nondeterministic automaton, then we can simplify the definition of probabilistic bisimulation, which, in this context, is called simply *bisimulation*, in the following way: a bisimulation respecting $\equiv$ on a nondeterministic automaton is an equivalence relation $\approx \subseteq S \times S$ such that $s \approx t$ implies that (1') $s \equiv t$, and (2') if $(s, a, s') \in \Rightarrow$, then there exists $(t, a, t') \in \Rightarrow$ such that $s' \approx t'$. States s and t are called bisimilar with respect to $\equiv$ in the nondeterministic automaton $\mathscr{P}$ if there exists a bisimulation $\approx$ respecting $\equiv$ such that $s \approx t$.

2.2 Probabilistic Hybrid Automata

Let $\mathscr{X}$ be a finite set of real-valued variables. A *valuation* $v : \mathscr{X} \to \mathbb{R}$ is a function that assigns a value to each variable of $\mathscr{X}$. Let $\mathscr{V}(\mathscr{X})$ be the set of valuations of $\mathscr{X}$. When the set $\mathscr{X}$ is clear from the context, we generally write $\mathscr{V}$.

A *probabilistic hybrid automaton* (PHA) $\mathscr{H} = (L, \mathscr{X}, Events, post, prob)$ consists of the following components: a finite set L of *locations*; a finite set $\mathscr{X}$ of variables; a finite set *Events* of *events*; a *post operator* $post : L \times \mathscr{V} \times \mathbb{R}_{\geq 0} \to 2^{\mathscr{V}}$; a finite set $prob \subseteq L \times 2^{\mathscr{V}} \times Events \times \mathsf{Dist}(Upd(\mathscr{X}) \times L)$ of *probabilistic edges*, where $Upd(\mathscr{X})$ is the set of functions $\vartheta : \mathscr{V} \to 2^{\mathscr{V}}$. A probabilistic edge $(l, g, e, p) \in prob$ comprises (1) a source location l, (2) a set g of valuations, called a *guard*, (3) an event e, and (4) a probability distribution p that assigns probability to pairs of the form (ϑ, l'), where $\vartheta \in Upd(\mathscr{X})$ is a function describing the manner in which variables are updated and $l' \in L$ is a target location.

The behaviour of a PHA takes a similar form to that of a classical, non-probabilistic hybrid automaton [ACH$^+$95]. If the PHA is currently in location l, as time passes, the value of the variables in $\mathscr{X}$ change according to the post operator *post*: more precisely, if the current valuation is v and $\delta \in \mathbb{R}_{\geq 0}$ time units elapse, the subsequent valuation belongs to the set $post(l, v, \delta)$. If the current valuation of the variables belongs to the guard g of a probabilistic edge (l, g, e, p), then the probabilistic edge can be taken. This involves a probabilistic choice according to the distribution p: if the pair (ϑ, l') is chosen, then the PHA goes to location l', updating the variables according to the function ϑ. More precisely, if the current valuation of the variables is v and the pair (ϑ, l') is chosen, then the state after taking the probabilistic edge will be (l', v') for some v' that is chosen nondeterministically from the set $\vartheta(v)$. To summarise, the following choices made by the PHA are nondeterministic: the amount of time to let advance in the current location l; the valuation used to describe the values of the variables after time has elapsed, according to *post*; the probabilistic edge taken (provided that the guard of the probabilistic edge is satisfied by the current variable valuation); and, finally, the values to which the variables are updated when a probabilistic edge is taken. Instead, the only probabilistic choice featured in the model concerns the choice of pair (ϑ, l') once a probabilistic edge has been chosen.

We make a number of standard assumptions on the components of a PHA [Spr01, Hah13].

- *(Assumptions on post.)* For all locations $l \in L$ and valuations $v \in \mathcal{V}$, we require the following: (1) $post(l,v,0) = \{v\}$; (2) for all $\delta, \delta' \in \mathbb{R}_{\geq 0}$ such that $\delta \geq \delta'$, we have $post(l,v,\delta) = \bigcup_{v' \in post(l,v,\delta')} post(l,v',\delta - \delta')$; (3) there exists $\delta \in \mathbb{R}_{\geq 0}$ such that $\delta > 0$ and $post(l,v,\delta) = \emptyset$.

- *(Probabilistic edges can be taken when no more time can elapse.)* If $l \in L$ and $v \in \mathcal{V}$ are such that, for all $\delta \in \mathbb{R}_{\geq 0}$ such that $\delta > 0$, we have $post(l,v,\delta) = \emptyset$, then there must exist some probabilistic edge $(l,g,e,p) \in prob$ such that $v \in g$.

- *(Non-empty updates.)* For all probabilistic edges $(l,g,e,p) \in prob$ and each $(\vartheta, l') \in \mathsf{support}(p)$, we have $\vartheta(v) \neq \emptyset$ for all $v \in g$.

We now introduce formally the semantics of PHA in terms of PA. The *(dense-time) semantics of the PHA* $\mathcal{H} = (L, \mathcal{X}, Events, post, prob)$ is the PA $[\![\mathcal{H}]\!] = (S, Act, \Rightarrow)$ defined in the following way. The set of states of $[\![\mathcal{H}]\!]$ is defined as $S = L \times \mathcal{V}$. The set of actions of $[\![\mathcal{H}]\!]$ is $Act = \mathbb{R}_{\geq 0} \cup Events$. To define the transition relation $\Rightarrow$, we first define a transition relation for each time duration and event.

- *(Time elapse.)* Let $\delta \in \mathbb{R}_{\geq 0}$. Then $\overset{\delta}{\Rightarrow} \subseteq S \times \mathbb{R}_{\geq 0} \times \mathsf{Dist}(S)$ is the largest set such that $((l,v), \delta, \{(l',v') \mapsto 1\}) \in \overset{\delta}{\Rightarrow}$ implies that (1) $l = l'$ and (2) $v' \in post(l,v,\delta)$.

- *(Jumps.)* Let $e \in Events$. Consider a distribution $p \in \mathsf{Dist}(Upd(\mathcal{X}) \times L)$, where $\mathsf{support}(p) = \{(\vartheta_1, l_1), \ldots, (\vartheta_n, l_n)\}$. Then, for valuation v, we write $\mathsf{Bundle}(v,p) \subseteq \mathcal{V}^n$ to denote the largest set of vectors of valuations such that $[v_1, \ldots, v_n] \in \mathsf{Bundle}(v,p)$ implies $v_i \in \vartheta_i(v)$ for each $1 \leq i \leq n$. Then $\overset{e}{\Rightarrow} \subseteq S \times \mathsf{Dist}(S)$ is the largest set of transitions such that $((l,v), e, \mu) \in \overset{e}{\Rightarrow}$ implies that there exists a probabilistic edge $(l,g,e,p) \in prob$ such that (a) $v \in g$ and (b) there exists $[v_1, \ldots, v_n] \in \mathsf{Bundle}(v,p)$ such that, for each $(l',v') \in S$:

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

Then we define $\Rightarrow$ as the transition relation $(\bigcup_{\delta \in \mathbb{R}_{\geq 0}} \overset{\delta}{\Rightarrow}) \cup (\bigcup_{e \in Events} \overset{e}{\Rightarrow})$.

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.

In the next section, we generally consider a time-abstract semantics of $\mathcal{H}$, in which actions corresponding to durations of time-elapse transitions are replaced by a single action τ (where we assume that $\tau \notin Events$). Formally, the *time-abstract semantics of* $\mathcal{H}$ is the PA $[\![\mathcal{H}]\!]^{\mathrm{ta}} = (S, Act, \Rightarrow)$, where the set S of states is the same as for the dense-time semantics of $\mathcal{H}$, the set of actions is defined as $Act = \{\tau\} \cup Events$, and the transition relation $\Rightarrow$ is defined as $\overset{\tau}{\Rightarrow} \cup (\bigcup_{e \in Events} \overset{e}{\Rightarrow})$, where $\overset{\tau}{\Rightarrow} = \{(s, \tau, \mu) \mid \exists \delta \in \mathbb{R}_{\geq 0}. (s, \delta, \mu) \in \overset{\delta}{\Rightarrow}\}$.

2.3 From Non-Probabilistic to Probabilistic Bisimulation

A *hybrid automaton* (HA) is a PHA $(L,\mathscr{X},Events,post,prob)$ for which all probabilistic edges $(l,g,e,p) \in prob$ correspond to a trivial probabilistic choice over a single element; more precisely, each $(l,g,e,p) \in prob$ is such that p is of the form $\{(\vartheta,l') \mapsto 1\}$ for some $\vartheta \in Upd(\mathscr{X})$ and $l' \in L$). We refer to probabilistic edges of the above form as *edges*.

Consider an arbitrary PHA $\mathscr{H} = (L,\mathscr{X},Events,post,prob)$. We let $ind(prob)$ be the smallest set of edges such that, if $(l,g,e,p) \in prob$ then, for each $(\vartheta,l') \in \mathsf{support}(p)$, there exists the probabilistic edge $(l,g,(e,p,\vartheta),\{(\vartheta,l') \mapsto 1\}) \in ind(prob)$. Let $ind(\mathscr{H}) = (L,\mathscr{X},Events \times \mathsf{Dist}(Upd(\mathscr{X}) \times L) \times Upd(\mathscr{X}),post,ind(prob))$ be the HA *induced by* the PHA $\mathscr{H}$. Note that the location, variable and post sets are the same in $\mathscr{H}$ and $ind(\mathscr{H})$. The events of $ind(\mathscr{H})$ feature tuples comprising an event of $\mathscr{H}$, the distribution over updates and locations, and an update. The set of probabilistic edges of $ind(\mathscr{H})$ is $ind(prob)$, i.e., a set of edges in which the events encode information derived from probabilistic edges in $prob$.

We present our first result, namely that bisimilar states of the time-abstract semantics of $ind(\mathscr{H})$ are probabilistically bisimilar in the time-abstract semantics of $\mathscr{H}$. In this paper, we regard locations as being observable from the point of view of properties (such as PCTL* formulae or ω-regular objectives), and hence we consider (probabilistic) bisimulation with respect to an equivalence relation that considers as equivalent states with the same location. We define the equivalence relation $\equiv_{\mathsf{loc}} \subseteq S \times S$ in the following way: $(l,v) \equiv_{\mathsf{loc}} (m,w)$ if and only if $l = m$, for all states $(l,v),(m,w) \in S$.

Proposition 1 *Let $\approx$ be a bisimulation with respect to $\equiv_{\mathsf{loc}}$ on $[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}$. Then $\approx$ is a probabilistic bisimulation with respect to $\equiv_{\mathsf{loc}}$ on $[\![\mathscr{H}]\!]^{\mathsf{ta}}$.*

Proof. We use $[\![\mathscr{H}]\!]^{\mathsf{ta}} = (S,Act_{[\![\mathscr{H}]\!]^{\mathsf{ta}}},\Rightarrow_{[\![\mathscr{H}]\!]^{\mathsf{ta}}})$ and $[\![ind(\mathscr{H})]\!]^{\mathsf{ta}} = (S,Act_{[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}},\Rightarrow_{[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}})$ be the time-abstract semantics of $\mathscr{H}$ and $ind(\mathscr{H})$, respectively (note that $[\![\mathscr{H}]\!]^{\mathsf{ta}}$ and $[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}$ have the same set of states, $S = L \times \mathscr{V}$). Let $\approx$ be a bisimulation respecting $\equiv_{\mathsf{loc}}$ on $[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}$.

Consider states $(l,v),(m,w) \in S$, and assume that $(l,v) \approx (m,w)$. This implies that the two conditions in the definition of bisimulation are satisfied: more precisely, we have (1') $l = m$, and (2') if $((l,v),a,(l',v')) \in \Rightarrow_{[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}}$, then there exists $((m,w),a,(m',w')) \in \Rightarrow_{[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}}$ such that $(l',v') \approx (m',w')$.

Given that $l = m$, condition (1) in the definition of probabilistic bisimulation is satisfied. Therefore it remains to show condition (2) in the definition of probabilistic bisimulation. Recall the definition of the action sets $Act_{[\![\mathscr{H}]\!]^{\mathsf{ta}}} = \{\tau\} \cup Events$ and $Act_{[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}} = \{\tau\} \cup (Events \times \mathsf{Dist}(Upd(\mathscr{X}) \times L) \times Upd(\mathscr{X}))$. We first consider transitions of $[\![\mathscr{H}]\!]^{\mathsf{ta}}$ and $[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}$ that correspond to the action τ, i.e., the transitions corresponding to time elapsing. The definition of time-elapse transitions depends on $post$, which is identical in both $\mathscr{H}$ and $ind(\mathscr{H})$. Hence, for (l,v), the existence of a transition $((l,v),\tau,(l',v')) \in \Rightarrow_{[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}}$ implies the existence of a transition $((l,v),\tau,\{(l',v') \mapsto 1\}) \in \Rightarrow_{[\![\mathscr{H}]\!]^{\mathsf{ta}}}$. Similarly, for (m,w), the existence of a transition $((m,w),\tau,(m',w')) \in \Rightarrow_{[\![ind(\mathscr{H})]\!]^{\mathsf{ta}}}$ implies the existence of a transition $((m,w),\tau,\{(m',w') \mapsto 1\}) \in \Rightarrow_{[\![\mathscr{H}]\!]^{\mathsf{ta}}}$. Recalling that $(l',v') \approx (m',w')$, we conclude that, in the case of action τ, we have that condition (2') in the definition of bisimulation implies condition (2) in the definition of probabilistic bisimulation.

We now consider jump transitions. Consider an edge $(l,g,(e,p,\vartheta),\{(\vartheta,l') \mapsto 1\}) \in \mathit{ind}(\mathit{prob})$ such that $v \in g$. We write $\mathsf{support}(p) = \{(\vartheta_1,l_1),\dots,(\vartheta_n,l_n)\}$. From the definition of the edge set $\mathit{ind}(\mathit{prob})$, we have that there exist edges $(l,g,(e,p,\vartheta_i),\{(\vartheta_i,l_i) \mapsto 1\}) \in \mathit{ind}(\mathit{prob})$ for all $1 \leq i \leq n$, where $\vartheta = \vartheta_i$ and $l' = l_i$ for some $1 \leq i \leq n$. Consider some vector $\mathbf{a} = [v_1,\dots,v_n] \in \mathscr{V}^n$ such that $v_i \in \vartheta_i(v)$ for each $1 \leq i \leq n$. We then consider the set of transitions corresponding to $\mathbf{a}$, namely $T_{\mathbf{a}} = \{((l,v),(e,p,\vartheta_i),(l_i,v_i)) \mid v_i \in \vartheta_i(v) \wedge 1 \leq i \leq n\}$. We have that $T_{\mathbf{a}} \subseteq \Rightarrow_{[\![\mathit{ind}(\mathscr{H})]\!]^{\mathrm{ta}}}$ for the following reasons: first, we have assumed above that $v \in g$; second, for each $1 \leq i \leq n$, noting that $\mathsf{Bundle}(v,\{(\vartheta_i,l_i) \mapsto 1\})$ contain vectors of length 1, namely those valuations v' such that $v' \in \vartheta_i(v)$, we obtain that $v_i \in \mathsf{Bundle}(v,\{(\vartheta_i,l_i) \mapsto 1\})$, which then implies that $((l,v),(e,p,\vartheta_i),\{(l_i,v_i) \mapsto 1\}) \in \Rightarrow_{[\![\mathit{ind}(\mathscr{H})]\!]^{\mathrm{ta}}}$ (furthermore, recall that we simplify the notation of such transitions to $((l,v),(e,p,\vartheta_i),(l_i,v_i))$). Informally, (l_i,v_i) is the unique state which corresponds to the traversal of edge $(l,g,(e,p,\vartheta_i),\{(\vartheta_i,l_i) \mapsto 1\})$ from (l,v).

Now, by condition (2') of the definition of bisimulation, the existence of each transition in $T_{\mathbf{a}}$ implies the existence of an equally-labelled transition from (m,w) leading to a bisimilar state. Formally, we can obtain a set $U = \{((m,w),(e,p,\vartheta_i),(m_i,w_i)) \mid ((l,v),(e,p,\vartheta_i),(l_i,v_i)) \in T_{\mathbf{a}} \wedge (l_i,v_i) \approx (m_i,w_i)\}$, and $U \subseteq \Rightarrow_{[\![\mathit{ind}(\mathscr{H})]\!]^{\mathrm{ta}}}$.

Next, we show that the transition sets $T_{\mathbf{a}}$ and U imply the existence of probabilistic transitions from (l,v) and (m,w) in $[\![\mathscr{H}]\!]^{\mathrm{ta}}$. First note that $\mathbf{a} \in \mathsf{Bundle}(v,p)$, because $v_i \in \vartheta_i(v)$ for each $1 \leq i \leq n$. Then $\mathbf{a}$ induces the transition $((l,v),e,\mu_{\mathbf{a}}) \in \Rightarrow_{[\![\mathscr{H}]\!]^{\mathrm{ta}}}$ where the distribution $\mu_{\mathbf{a}}$ is defined as $\mu_{\mathbf{a}}(l',v') = \sum_{1 \leq i \leq n \wedge v' = \mathbf{a}[i]} p(\vartheta_i,l_i)$ for each $(l',v') \in S$.

Let $\mathbf{b} = [w_1,\dots,w_n]$. Note that, for each $1 \leq i \leq n$, we have $(l_i,\mathbf{a}[i]) \approx (m_i,\mathbf{b}[i])$ (because $(l_i,v_i) \approx (m_i,w_i)$). We also have $\mathbf{b} \in \mathsf{Bundle}(v,p)$ because, for each $1 \leq i \leq n$, the existence of the transition $((m,w),(e,p,\vartheta_i),(m_i,w_i))$ implies that $w_i \in \vartheta_i(w)$. In a similar manner to the case of $\mathbf{a}$, we have that $\mathbf{b}$ induces the transition $((m,w),e,\nu_{\mathbf{b}}) \in \Rightarrow_{[\![\mathscr{H}]\!]^{\mathrm{ta}}}$, where $\nu_{\mathbf{b}}(m',w') = \sum_{1 \leq i \leq n \wedge w' = \mathbf{b}[i]} p(\vartheta_i,m_i)$ for each $(m',w') \in S$.

We now show that $\mu_{\mathbf{a}}[C] = \nu_{\mathbf{b}}[C]$ for all equivalence classes C of $\approx$. Recall that:

$$\begin{aligned}
\mu_{\mathbf{a}}[C] &= \sum_{(l',v') \in C} \mu_{\mathbf{a}}(l',v') &= \sum_{(l',v') \in C} \; \sum_{1 \leq i \leq n \text{ s.t. } v' = \mathbf{a}[i]} p(\vartheta_i,l') \\
\nu_{\mathbf{b}}[C] &= \sum_{(m',w') \in C} \nu_{\mathbf{b}}(m',w') &= \sum_{(m',w') \in C} \; \sum_{1 \leq i \leq n \text{ s.t. } w' = \mathbf{b}[i]} p(\vartheta_i,m') \,.
\end{aligned}$$

Given that $\approx$ respects $\equiv_{\mathsf{loc}}$, then, for all $(l',v'),(l'',v'') \in C$, we have $l' = l''$. We use l_C to denote the location component of the states in C. Now consider the sets $\mathscr{I}_C^{\mathbf{a}} = \{i \in \mathbb{N} \mid 1 \leq i \leq n \wedge (l_C,\mathbf{a}[i]) \in C\}$ and $\mathscr{I}_C^{\mathbf{b}} = \{i \in \mathbb{N} \mid 1 \leq i \leq n \wedge (l_C,\mathbf{b}[i]) \in C\}$. Note that we can write:

$$\begin{aligned}
\sum_{(l_C,v') \in C} \; \sum_{1 \leq i \leq n \text{ s.t. } v' = \mathbf{a}[i]} p(\vartheta_i,l_C) &= \sum_{i \in \mathscr{I}_C^{\mathbf{a}}} p(\vartheta_i,l_C) \\
\sum_{(l_C,w') \in C} \; \sum_{1 \leq i \leq n \text{ s.t. } w' = \mathbf{b}[i]} p(\vartheta_i,l_C) &= \sum_{i \in \mathscr{I}_C^{\mathbf{b}}} p(\vartheta_i,l_C) \,.
\end{aligned}$$

Hence, to show that $\mu_{\mathbf{a}}[C] = \nu_{\mathbf{b}}[C]$, it suffices to show that $\sum_{i \in \mathscr{I}_C^{\mathbf{a}}} p(\vartheta_i,l') = \sum_{i \in \mathscr{I}_C^{\mathbf{b}}} p(\vartheta_i,m')$.

Given that we established above that, for each $1 \leq i \leq n$, we have $(l_C,\mathbf{a}[i]) \approx (l_C,\mathbf{b}[i])$, we also conclude that $(l_C,\mathbf{a}[i]) \in C$ if and only if $(l_C,\mathbf{b}[i]) \in C$. This implies that $\mathscr{I}_C^{\mathbf{a}} = \mathscr{I}_C^{\mathbf{b}}$. We then have that $\sum_{i \in \mathscr{I}_C^{\mathbf{a}}} p(\vartheta_i,l') = \sum_{i \in \mathscr{I}_C^{\mathbf{b}}} p(\vartheta_i,m')$. Hence $\mu_{\mathbf{a}}[C] = \nu_{\mathbf{b}}[C]$. We thus conclude that $\mu_{\mathbf{a}} \approx \nu_{\mathbf{b}}$. Condition (2) of the definition of probabilistic bisimulation has been satisfied. □

3 Approximate Abstraction of a Class of Stochastic Hybrid Automata

In this section, we consider the analysis of a restricted class of stochastic hybrid automata (SHA) by a reduction to probabilistic rectangular automata (PRA), which can then be transformed into a finite-state PA. We describe briefly SHA in Section 3.1 and PRA in Section 3.2. As an intermediate step between SHA and PRA, we introduce a class of PHA called *probabilistic slope-update hybrid automata* (PSUHA) in Section 3.3. Given certain assumptions, we show that PSUHA can be reduced to PRA. We illustrate the approach with respect to the example of Figure 1.

3.1 Stochastic Hybrid Automata: A Brief Overview

We now describe briefly the SHA formalism; for a more technical introduction, see [FHH$^+$11, Hah13]. A SHA features the same components as a PHA, but also features a set of *continuous probabilistic edges*. A continuous probabilistic edge $(l,g,e,\widetilde{p})$ comprises (1) a source location l, (2) a guard g, (3) an event e, and (4) a measurable function $\widetilde{p}$ mapping each state to a probability measure over locations and valuations. We assume that *post*, guards and update sets of standard probabilistic edges, and guards of continuous probabilistic edges are measurable in the sense of [FHH$^+$11, Hah13].

We informally describe the semantics of SHA. As in the case of PHA, from state (l,v) time can pass with the values of the continuous variables changing as described by the operator *post*. The conditions in which standard probabilistic edges can be taken, and their effects on the location and continuous variables, are also as in the case of *prob* of PHA. Similarly to standard probabilistic edges, a continuous probabilistic edge $(l,g,e,\widetilde{p})$ can be taken if the current state is (l,v) for $v \in g$, and the location and valuation is updated according to a probabilistic choice. However, for a continuous probabilistic edge, a probability measure (which depends on the current state) described by $\widetilde{p}$ is used; in particular, this allows us to update variables according to infinite-support probability distributions, such as the continuous normal or uniform distributions. In [FHH$^+$11, Hah13], continuous probabilistic edges have been used in the modelling of hybrid systems that are subject to measurement errors.

Note that, as with PHA, SHA do not feature probabilistic choice over elements of the post operator: therefore, choices regarding the amount of time to let elapse, and regarding the change to the continuous variables corresponding to time elapsing, are nondeterministic. However, the change to continuous variables corresponding to time elapsing can be made to be dependent on continuous probabilistic choice indirectly. Consider an example of a stochastic hybrid system in which the rate of change of a continuous variable x as time passes in location l is determined by a continuous probabilistic choice, say a uniform distribution over the interval $[1,3]$, made when location l is entered. Then we can model this system as a SHA with a continuous variable $\overline{x}$ that is updated by continuous probabilistic edges according to a uniform distribution over $[1,3]$ on entry to location l. The post operator then specifies that the rate of change of x as time passes in l is equal to the value of $\overline{x}$, and that the value of $\overline{x}$ does not change as time passes in l.

We now explain briefly the methodology of [FHH$^+$11, Hah13] for the construction of approximate abstractions of SHA. This approach concerns the construction of a PHA that is identical to the SHA except for the fact that each continuous probabilistic edge is replaced by a standard probabilistic edge with the same source location, guard and event, but for which the distribution ranges over update sets and target locations rather than over variable values and target locations. For example, consider the case of a continuous probabilistic edge $(l,g,e,\widetilde{p})$, where $\widetilde{p}$ corresponds to the choice of a single target location l', and updates the value of a variable x according to a uniform distribution over $[1,3]$, while leaving the values of the other variables unchanged. The continuous probabilistic edge can be replaced by a discrete probabilistic edge (l,g,e,p), where p is defined in the following manner. The interval $[1,3]$ is represented by a finite number of intervals, the union of which contains $[1,3]$, and each of which corresponds to a probability derived from the original distribution. For example, we can consider the intervals $[1,2]$ and $[2,3]$, which each correspond to probability $\frac{1}{2}$, in accordance with the original uniform distribution. Then we let $p = \{(\vartheta_1,l') \mapsto \frac{1}{2}, (\vartheta_2,l') \mapsto \frac{1}{2}\}$, where ϑ_1 (ϑ_2, respectively) corresponds to a nondeterministic choice of which value to update x to within the interval $[1,2]$ ($[2,3]$, respectively), while leaving the values of the other variables unchanged. It can be seen that this construction results in a conservative approximation of a SHA, in the sense that any strategy of the SHA can be emulated by a randomised strategy of the constructed PHA (see [Hah13], Theorem 4.22). Hence the minimum (maximum, respectively) probability of reaching a certain location in the PHA will be no greater than (no less than, respectively) the minimum (maximum, respectively) probability of reaching the location in the SHA. This approximate abstraction methodology then allows tools for the analysis for PHA to be applied to the analysis of SHA.

3.2 Probabilistic Rectangular Automata

Let $\mathscr{I}$ be the set of intervals $\{[a,b] \mid a,b \in \mathbb{Z}\} \cup \{(-\infty,a] \mid a \in \mathbb{Z}\} \cup \{[a,\infty) \mid a \in \mathbb{Z}\} \cup \{(-\infty,\infty)\}$. In the following, we describe a set V of valuations $\mathscr{V}(X)$ over $X \subseteq \mathscr{X}$ to be *rectangular over* X if, for each variable $x \in X$, we have that there exists an interval $I_x \in \mathscr{I}$ such that $v \in V$ if and only if $v(x) \in I_x$ for all $x \in X$. In this case, we often write $\prod_{x \in X} I_x$ to denote V. If a set of valuations is rectangular over $\mathscr{X}$, we simply describe the set as being rectangular.

In a PRA, all guards and updates are described in terms of rectangular sets over subsets of the set $\mathscr{X}$ of continuous variables. The post operator of a PRA is characterised in the following way: for each location, (1) the rate of change of each variable belongs to an interval, and (2) the variable values that can be obtained as time passes are constrained to be within a rectangular set. Formally a PRA $\mathscr{R} = (L, \mathscr{X}, \mathit{Events}, \mathit{post}, \mathit{prob})$ is defined as a PHA with the following restrictions. For each probabilistic edge $(l,g,e,p) \in \mathit{prob}$, we have that g is rectangular. Furthermore, writing $\mathsf{support}(p) = \{(\vartheta_1,l_1),\ldots,(\vartheta_n,l_n)\}$, for each $1 \leq i \leq n$ there exists variable set $\mathrm{Reset}(\vartheta_i) \subseteq \mathscr{X}$ and a rectangular set $\prod_{x \in \mathrm{Reset}(\vartheta_i)} [\underline{\mathrm{u}}_x^{\vartheta_i}, \overline{\mathrm{u}}_x^{\vartheta_i}]$ over $\mathrm{Reset}(\vartheta_i)$ such that, for each $v \in \mathscr{V}, v' \in \vartheta_i(v)$, we have $v'(x) \in [\underline{\mathrm{u}}_x^{\vartheta_i}, \overline{\mathrm{u}}_x^{\vartheta_i}]$ for each $x \in \mathrm{Reset}(\vartheta_i)$ and $v'(x) = v(x)$ for each $x \in \mathscr{X} \setminus \mathrm{Reset}(\vartheta_i)$. The post operator *post* takes the following form: for each location $l \in L$, there exists a rectangular set $\mathit{inv}(l) = \prod_{x \in \mathscr{X}} I_x^l$, and there exists an interval $[\underline{\mathrm{f}}_x^l, \overline{\mathrm{f}}_x^l]$ with $\underline{\mathrm{f}}_x^l, \overline{\mathrm{f}}_x^l \in \mathbb{Z}$ for each $x \in \mathscr{X}$ such that, for each valuation $v \in \mathscr{V}$ and delay $\delta \in \mathbb{R}_{\geq 0}$, we have $\mathit{post}(l,v,\delta) = \{v' \mid (\forall x \in \mathscr{X}. v'(x) \in [v(x) + \delta * \underline{\mathrm{f}}_x^l, v(x) + \delta * \overline{\mathrm{f}}_x^l]) \wedge v' \in \mathit{inv}(l)\}$.

As PRA is a subclass of PHA, we can define the dense-time semantics $[\![\mathscr{R}]\!]$ of a PRA as in Section 2. The *discrete-time semantics of the PRA* $\mathscr{R} = (L, \mathscr{X}, \mathit{Events}, \mathit{post}, \mathit{prob})$ *with respect to* $k \in \mathbb{N} \setminus \{0\}$ is the PA $\langle\!\langle \mathscr{R} \rangle\!\rangle^k = (S, \mathit{Act}, \Rightarrow)$ defined as for the dense-time semantics except for $\Rightarrow$, which is defined as $\overset{\frac{1}{k}}{\Rightarrow} \cup (\bigcup_{e \in \mathit{Events}} \overset{e}{\Rightarrow})$. We say that the variable $x \in \mathscr{X}$ is *nondecreasing* if both $I_x^l \subseteq \mathbb{R}_{\geq 0}$ and $[\underline{\mathrm{f}}_x^l, \overline{\mathrm{f}}_x^l] \subseteq \mathbb{R}_{\geq 0}$ for all locations $l \in L$. The variable $x \in \mathscr{X}$ is *bounded* if I_x^l is a bounded set, for all locations $l \in L$. The PRA $\mathscr{R}$ has *nondecreasing or bounded* variables if all variables in $\mathscr{X}$ are either nondecreasing or bounded. Given a PRA $\mathscr{R}$ with nondecreasing or bounded variables, the discrete-time semantics $\langle\langle \mathscr{R} \rangle\rangle^k$ of $\mathscr{R}$ with respect to any $k \in \mathbb{N}$ has a finite number of classes for probabilistic bisimulation with respect to $\equiv_{\mathsf{loc}}$ [Spr11]. This result permits the construction of a finite-state PA that is equivalent to the PRA, and which can then be analysed using tools for finite-state PA, such as PRISM [KNP11].

3.3 Probabilistic Slope-Update Hybrid Automata

We now consider PSUHA, a class of PHA that generalises PRA, and in which the rate of change of some variables can be described as the value of other (constant) variables. A PSUHA $\mathscr{U} = (L, \mathscr{X}, Events, post, prob)$ is a PHA where the components L, $\mathscr{X}$, $Events$ and $prob$ are defined as in the case of PRA, and where the post operator $post$ is defined in the following way. First, we identify a subset $\mathsf{C} \subseteq \mathscr{X}$ of variables that remain constant as time passes, in each location (variables in C may be reset when probabilistic edges are taken). For each location $l \in L$, there exists a subset $\mathsf{Dep}(l) \subseteq \mathscr{X}$ of variables that have a rate of change that is equal to the value of a variable in C (note that $\mathsf{Dep}(l) \cap \mathsf{C} = \emptyset$). For variable $x \in \mathsf{Dep}(l)$, we write $\mathsf{DepOn}(l,x) \in \mathsf{C}$ to denote the variable on which the rate of change of x depends in l. Let $\mathsf{Rectangular}(l) = \mathscr{X} \setminus (\mathsf{C} \cup \mathsf{Dep}(l))$. We can now proceed to define the post operator $post$: for each location $l \in L$, there exists a rectangular set $inv(l) = \prod_{x \in \mathscr{X}} I_x$, and an interval $[\underline{\mathrm{f}}_x^l, \overline{\mathrm{f}}_x^l]$ with $\underline{\mathrm{f}}_x^l, \overline{\mathrm{f}}_x^l \in \mathbb{Z}$ for each $x \in \mathsf{Rectangular}(l)$ such that, for each valuation $v \in \mathscr{V}$ and delay $\delta \in \mathbb{R}_{\geq 0}$, we have:

$$
\begin{aligned}
post(l,v,\delta) = \{v' \mid \quad & (\forall x \in \mathsf{Rectangular}(l).v'(x) \in [v(x) + \delta * \underline{\mathrm{f}}_x^l, v(x) + \delta * \overline{\mathrm{f}}_x^l]) \\
& \wedge (\forall x \in \mathsf{Dep}(l).v'(x) = v(x) + \delta * v(\mathsf{DepOn}(l,x)) \\
& \wedge (\forall x \in \mathsf{C}.v'(x) = v(x)) \\
& \wedge v' \in inv(l)\} \,.
\end{aligned}
$$

The set $prob$ of probabilistic edges of a PSUHA is subject to the following assumptions:

1. *(Variables in* C *are reset on entry to each location.)* Each $(l,g,e,p) \in prob$ is such that, for each $(\vartheta, l') \in \mathsf{support}(p)$ and for each $x \in \mathsf{C}$, we have that $x \in \mathsf{Reset}(\vartheta)$.

2. *(On entry to a given location by multiple probabilistic edges, the same interval is used to define the value of a variable in* C*.)* For each $l \in L$ and each $x \in \mathsf{C}$, there exist $\underline{\mathrm{u}}_x^l, \overline{\mathrm{u}}_x^l \in \mathbb{Z}$ such that, for each $(\vartheta, l) \in \bigcup_{(l',g,e,p) \in prob} \mathsf{support}(p)$, we have $\underline{\mathrm{u}}_x^\vartheta = \underline{\mathrm{u}}_x^l$ and $\overline{\mathrm{u}}_x^\vartheta = \overline{\mathrm{u}}_x^l$.

3. *(No probabilistic edge features a probabilistic choice between updates associated with the same location.)* For each $(l,g,e,p) \in prob$, for each pair $(\vartheta', l'), (\vartheta'', l'') \in \mathsf{support}(p)$, we have $l' \neq l''$.

4. *(The guards of probabilistic edges do not constrain values of variables in* C*.)* For each $(l,g,e,p) \in prob$ where we write $g = \prod_{x \in \mathscr{X}} I_x$, and each $x \in \mathsf{C}$, we have that $I_x = (-\infty, \infty)$.

We now show that, for any PSUHA, we can construct a PRA with the same state set such that the maximum reachability probabilities are identical in the PSUHA and PRA, and similarly for the minimum reachability probabilities. We present the result in the context of the discrete-time semantics, although we note that an analogous result holds for the dense-time semantics. Let $k \in \mathbb{N} \setminus \{0\}$, and let $\mathscr{U}$ and $\mathscr{R}$ be PSUHA and PRA, respectively, such that $\langle\langle \mathscr{U} \rangle\rangle^k$ and $\langle\langle \mathscr{R} \rangle\rangle^k$ have the same state set. Then we introduce the following equivalence relation on states: for state (l,v) of $\langle\langle \mathscr{U} \rangle\rangle^k$ and state (m,w) of $\langle\langle \mathscr{R} \rangle\rangle^k$, we write $(l,v) \equiv_{\overline{\mathsf{C}}} (m,w)$ if and only if (1) $l = m$ and (2) $v(x) = w(x)$ for all $x \in \mathscr{X} \setminus \mathsf{C}$.

Proposition 2 *Let $\mathscr{U}$ be a PSUHA, $k \in \mathbb{N} \setminus \{0\}$ and $F \subseteq L$ be a set of locations. Then there exists a PRA $\mathscr{R}$ such that $\langle\langle \mathscr{U} \rangle\rangle^k$ and $\langle\langle \mathscr{R} \rangle\rangle^k$ have the same state set and, for each pair $(l,v),(m,w)$ of states such that $(l,v) \equiv_{\overline{\mathsf{C}}} (m,w)$, we have:*

$$MaxReach_{(l,v)}^{\langle\langle \mathscr{U} \rangle\rangle^k}(S_F) = MaxReach_{(m,w)}^{\langle\langle \mathscr{R} \rangle\rangle^k}(S_F)\ ,\ MinReach_{(l,v)}^{\langle\langle \mathscr{U} \rangle\rangle^k}(S_F) = MinReach_{(m,w)}^{\langle\langle \mathscr{R} \rangle\rangle^k}(S_F)\ .$$

where $S_F = \{(l,v) \mid (l,v) \in S \wedge l \in F\}$.

Proof sketch. For details of the proof, see [Spr14]. Let $\mathscr{U} = (L, \mathscr{X}, Events, post, prob)$ be a PSUHA. We construct the PRA $\mathscr{R} = (L, \mathscr{X}, Events, post', prob)$ to be identical to $\mathscr{U}$ except for the post operator. For each state $(l,v) \in S$ and each duration $\delta \in \mathbb{R}_{\geq 0}$, we define:

$$\begin{aligned} post'(l,v,\delta) = \{v' \mid\ & (\forall x \in \mathsf{Rectangular}(l).v'(x) \in [v(x) + \delta * \underline{\mathrm{f}}^l_x, v(x) + \delta * \overline{\mathrm{f}}^l_x]) \\ & \wedge (\forall x \in \mathsf{Dep}(l).v'(x) \in [v(x) + \delta * \underline{\mathrm{u}}^l_{\mathsf{DepOn}(l,x)}, v(x) + \delta * \overline{\mathrm{u}}^l_{\mathsf{DepOn}(l,x)}]) \\ & \wedge (\forall x \in \mathsf{C}.v'(x) = v(x)) \\ & \wedge v' \in inv(l)\}\ . \end{aligned}$$

Note that, for each variable x in $\mathsf{Dep}(l)$, the slope of x in l is chosen the interval $[\underline{\mathrm{u}}^l_{\mathsf{DepOn}(l,x)}, \overline{\mathrm{u}}^l_{\mathsf{DepOn}(l,x)}]$, i.e., the interval used to determine the value of the variable $\mathsf{DepOn}(l,x)$ on which x depends on entry to l. It can be seen that $post'$ conforms to the definition of possible post operators of PRA, hence $\mathscr{R}$ is a PRA.

The proof then involves showing the following condition: a jump transition of $\langle\langle \mathscr{U} \rangle\rangle^k$ followed by a sequence of time transitions can be replicated by a jump transition of $\langle\langle \mathscr{R} \rangle\rangle^k$ followed by a sequence of time transitions, and vice versa, where the states before and after the transitions are related by $\equiv_{\overline{\mathsf{C}}}$. Furthermore, the jump transitions are made using the same probabilistic edges, which means that corresponding transitions can be shown to have the same probability. Note that we consider such a *multi-step equivalence* relation (that is, we compare $\langle\langle \mathscr{U} \rangle\rangle^k$ and $\langle\langle \mathscr{R} \rangle\rangle^k$ by considering a jump transition and sequences of time-elapse transitions), rather than a 1-step relation such as probabilistic bisimulation, for the following reason: when considering the case in which $\langle\langle \mathscr{U} \rangle\rangle^k$ must emulate a jump transition of $\langle\langle \mathscr{R} \rangle\rangle^k$, the rates of change chosen by $\langle\langle \mathscr{R} \rangle\rangle^k$ in subsequent transitions must be known by $\langle\langle \mathscr{U} \rangle\rangle^k$, so that $\langle\langle \mathscr{U} \rangle\rangle^k$ knows what value to set the constants on which the rates of change of variables will depend.

Using this multi-step equivalence property, given that time-elapse transitions correspond to probability 1, it follows that, for state (l,v) of $\langle\langle \mathscr{U} \rangle\rangle^k$ and state (m,w) of $\langle\langle \mathscr{R} \rangle\rangle^k$ such that $(l,v) \equiv_{\overline{\mathsf{C}}} (m,w)$, we have $MaxReach_{(l,v)}^{\langle\langle \mathscr{U} \rangle\rangle^k}(S_F) = MaxReach_{(m,w)}^{\langle\langle \mathscr{R} \rangle\rangle^k}(S_F)$ and $MinReach_{(l,v)}^{\langle\langle \mathscr{U} \rangle\rangle^k}(S_F) = MinReach_{(m,w)}^{\langle\langle \mathscr{R} \rangle\rangle^k}(S_F)$. □

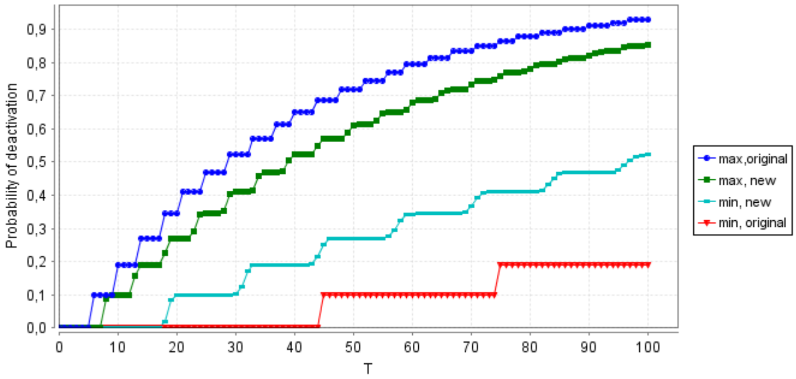

Figure 2: Maximum and minimum probabilities of visiting l_{deact} within time bound T

Proposition 2 suggests the following approximate analysis method for the class of discrete-time SHA that, when abstracted using the method of [FHH$^+$11, Hah13], result in a discrete-time PSUHA with nondecreasing or bounded variables: from the PSUHA, then obtain an equivalent PRA according to Proposition 2; subsequently, the obtained PRA is transformed into a finite-state PA using the results of [Spr11].

We also mention that we can encode in PSUHA (and hence in the associated PRA) the periodic *resampling* of variables in C, which can be done every $\frac{1}{k}$ time units, for some $k \in \mathbb{N} \setminus \{0\}$. The intuition underlying the encoding is that at most $\frac{1}{k}$ time units can elapse in each location before taking a probabilistic edge. This can be enforced by adding a clock variable x and by defining the post operator so that the value of x cannot exceed $\frac{1}{k}$. Each location has a “self-loop” probabilistic edge that is enabled when x is equal to $\frac{1}{k}$, resets x to 0, resets variables in C and does not change any other variable.[1]

3.4 Example: Faulty Thermostat

We now illustrate the approximate analysis method proposed in the previous section with an application to the PRA model of a faulty thermostat presented in Figure 1. As described in Section 1, when the heater is on, we consider that the rate of increase of the temperature is chosen from the normal distribution with mean 3.5 and standard deviation 1, truncated to the interval $[1,6]$. Similarly, when the heater is off, instead of a nondeterministically chosen decrease of within $[-4,-1]$, we consider the normal distribution with mean -2.5 and standard deviation 0.5, truncated to the interval $[-4,-1]$. This system can be modelled as a SHA in the sense of Section 3.1. Now we consider how the approximate abstraction approach of [FHH$^+$11, Hah13] can be used to obtain a PSUHA. For the case in which the heater is on, we consider three subintervals, $[1,3]$, $[3,4]$ and $[4,6]$, which correspond approximately to the probabilities 0.312, 0.376 and 0.312, respectively, of the aforementioned normal distribution. For the case in which the

[1] Note that this construction involves probabilistic edges that choose between alternatives involving the same location: to satisfy the assumption (3) in the definition of the set of probabilistic edges of a PSUHA, we can consider multiple copies of the location, each corresponding to a different alternative associated with each probabilistic edge, w.l.o.g.

heater is off, we consider the three subintervals $[-4,-3]$, $[-3,-2]$ and $[-2,-1]$, corresponding to the probabilities 0.159, 0.682 and 0.159, respectively. We then constructed the PSUHA according to the discrete-time resampling construction, then transformed the resulting model to a PRA and, in turn, to a finite-state PA, which was analysed with PRISM.

In Figure 2, we present the maximum and minimum probabilities of visiting location l_{deact} within time bound T, both in the "original" PRA model shown in Figure 1, and in the "new" PRA model obtained by the method described above. In both cases, we use the time granularity $k = 10$, resulting in a PA with 136112 states. It is clear that the results obtained for the original PRA model bound those obtained by the new PRA model, often by a substantial amount.

4 Conclusions

We have presented general results on obtaining finite bisimulation quotients for the exact verification of PHA, and a method for the approximate verification of a restricted subclass of SHA, based on a combination of the approximation technique of [FHH+11, Hah13] and the discrete-time verification method of [Spr11]. We mention some limitations of the latter approach. First, the approach of choosing probabilistically a slope of a variable on entry to a location, which then remains constant in that location, may be unrealistic for some applications, in which the slope of a variable may be subject to stochastic fluctuation as time passes. This criticism applies principally to the case of the dense-time semantics: in a discrete-time context, only the target state after $\frac{1}{k}$ time units is important, rather than the trajectory used to reach it, because nondeterministic choice is disabled as time passes during the $\frac{1}{k}$ time units. Second, our restriction of bounded slopes in the context of PSUHA leads to the necessity of truncation of some continuous distributions. We envisage that this restriction can be lifted. In future work we plan to apply the results of Section 3 to more realistic case studies.

Acknowledgements: Thanks to Ahmed Bouajjani for comments regarding the results of Section 2.

Bibliography

[ACH+95] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine. The Algorithmic Analysis of Hybrid Systems. *TCS* 138(1):3–34, 1995.

[APLS08] A. Abate, M. Prandini, J. Lygeros, S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica* 44(11):2724–2734, 2008.

[BK08] C. Baier, J.-P. Katoen. *Principles of model checking*. MIT Press, 2008.

[Buj04] M. L. Bujorianu. Extended Stochastic Hybrid Systems and Their Reachability Problem. In *Proc. HSCC'04*. LNCS 2993, pp. 234–249. Springer, 2004.

[FHH+11] M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, L. Zhang. Measurability and Safety Verification for Stochastic Hybrid Systems. In *Proc. HSCC'11*. Pp. 43–52. ACM, 2011.

[Fre08] G. Frehse. PHAVer: Algorithmic Verification of Hybrid systems Past HyTech. *STTT* 10(3), 2008.

[Hah13] E. M. Hahn. *Model checking stochastic hybrid systems*. Dissertation, Universität des Saarlandes, 2013.

[HHW97] T. A. Henzinger, P.-H. Ho, H. Wong-Toi. HyTech: A Model Checker for Hybrid Systems. *STTT* 1(1-2):110–122, 1997.

[HLS00] J. Hu, J. Lygeros, S. Sastry. Towards a Theory of Stochastic Hybrid Systems. In *Proc. HSCC'00*. LNCS 1790, pp. 160–173. Springer, 2000.

[KNP11] M. Kwiatkowska, G. Norman, D. Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *Proc. CAV'11*. LNCS 6806, pp. 585–591. Springer, 2011.

[KSK76] J. G. Kemeny, J. L. Snell, A. W. Knapp. *Denumerable Markov Chains*. Graduate Texts in Mathematics. Springer, 2nd edition, 1976.

[LS91] K. G. Larsen, A. Skou. Bisimulation through Probabilistic Testing. *I & C* 94(1):1–28, 1991.

[Put94] M. L. Puterman. *Markov Decision Processes*. J. Wiley & Sons, 1994.

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

[Spr00] J. Sproston. Decidable Model Checking of Probabilistic Hybrid Automata. In *Proc. FTRTFT'00*. LNCS 1926, pp. 31–45. Springer, 2000.

[Spr01] J. Sproston. *Model Checking for Probabilistic Timed and Hybrid Systems*. PhD thesis, School of Computer Science, University of Birmingham, 2001.

[Spr11] J. Sproston. Discrete-Time Verification and Control for Probabilistic Rectangular Hybrid Automata. In *Proc. QEST'11*. Pp. 79–88. IEEE, 2011.

[Spr14] J. Sproston. Exact and Approximate Abstraction for Classes of Stochastic Hybrid Systems. 2014. Available from http://www.di.unito.it/∼sproston/Research/avocs14-extended.pdf.

[VPVD08] V. Vladimerou, P. Prabhakar, M. Viswanathan, G. E. Dullerud. STORMED Hybrid Systems. In *Proc. ICALP'08*. LNCS 5126, pp. 136–147. Springer, 2008.

[ZSR+12] L. Zhang, Z. She, S. Ratschan, H. Hermanns, E. M. Hahn. Safety Verification for Probabilistic Hybrid Systems. *European Journal of Control* 18(6):572–587, 2012.