

# State Explosion in Almost-Sure Probabilistic Reachability

François Laroussinie

*Lab. Spécification & Vérification, ENS de Cachan & CNRS UMR 8643,  
61, av. Pdt. Wilson, 94235 Cachan Cedex France*

Jeremy Sproston

*Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy*

---

## Abstract

We show that the problem of reaching a state set with probability 1 in probabilistic-nondeterministic systems operating in parallel is EXPTIME-complete. We then show that this probabilistic reachability problem is EXPTIME-complete also for probabilistic timed automata.

*Key words:* probabilistic systems, model checking, computational complexity, formal methods, timed automata

---

## 1 Introduction

*Model checking* is an automatic method for guaranteeing that a mathematical model of a system satisfies a formula representing a desired property [4]. Many real-life systems, such as multimedia equipment, communication protocols, networks and fault-tolerant systems, exhibit *probabilistic* behavior, leading to the study of *probabilistic model checking* of probabilistic and stochastic models (for an overview, see [12]). We often incorporate nondeterministic choice in probabilistic models, resulting in formalisms akin to *Markov decision processes* [15]. Furthermore, formalisms such as *probabilistic timed automata* [11,9] (an

---

*Email addresses:* `f1@lsv.ens-cachan.fr` (François Laroussinie),  
`sproston@di.unito.it` (Jeremy Sproston).

<sup>1</sup> Supported in part by Miur project Fibr-Perf and EEC project Crutial.

extension of Markov decision processes with clock variables, as in timed automata [2]) can represent models in which nondeterminism, probability and timing information coexist.

The description of a probabilistic system is usually given in terms of interacting sub-systems composed in parallel, or by models referring to variables; an example is the system description language of the probabilistic model-checking tool PRISM [8]. However, the number of system states is exponential in the size of such a description: this is known as the *state-explosion problem*, and is the main practical limitation of model checking. In this paper, we show that the problem “*does there exist a way of resolving the nondeterministic choice of the system such that a set of states is reached with probability 1?*” is EXPTIME-complete both for a set of probabilistic systems operating in parallel and for probabilistic timed automata. A positive answer to this *almost-sure (or qualitative) probabilistic reachability problem* establishes that the probabilistic system can guarantee an event (such as the completion of a task) with probability 1. The reachability problem is a fundamental sub-problem of model checking, and, analogously, the almost-sure probabilistic reachability problem is a fundamental sub-problem of probabilistic model checking. Hence, the EXPTIME lower bounds shown in this paper apply to *all* probabilistic model-checking problems for the systems we consider.

A similar result has been shown by Littman [14] in the context of probabilistic propositional planning, which involves the solution of a probabilistic reachability problem on a concisely-described Markov decision process. Littman’s result relies on the reduction of the two-player game  $G_4$  to reachability on a Markov decision process described in the sequential-effect trees notation. Our approach is instead to reduce the acceptance problem on linearly-bounded alternating Turing machines to the almost-sure probabilistic reachability problem, both on probabilistic systems operating in parallel and on probabilistic timed automata, in a similar manner to the reductions in [13,1].

*Preliminaries.* An *Alternating Turing Machine* (ATM) [3] is a tuple  $\mathcal{A} = (Q, Q_\vee, Q_\wedge, \Gamma, \delta, q_0, q_{acc})$ , with a set  $Q = Q_\vee \cup Q_\wedge$  of states partitioned into disjunctive states  $Q_\vee$  and conjunctive states  $Q_\wedge$ , an initial state  $q_0 \in Q$ , an accepting state  $q_{acc} \in Q_\vee$ , a tape alphabet  $\Gamma = \{a, b\}$ , and a transition relation  $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{-1, 1\}$ . A *configuration* of  $\mathcal{A}$  is a triple  $\alpha = (q, i, w)$  where  $q \in Q$  is the current state,  $w \in \Gamma^*$  is a word describing the tape content, and  $0 < i \leq |w|$  is the position of the head on the tape. The symbol written in the  $i$ th cell of the tape is denoted by  $w(i)$ . An ATM moves like a usual nondeterministic Turing machine: for example, if  $\alpha = (q, i, w)$ ,  $w(i) = a$  and  $(q, a, q', b, \varepsilon) \in \delta$ , then  $\mathcal{A}$  may move from  $\alpha$  to  $\alpha' = (q', i', w')$ , where  $w'$  is  $w$  updated by writing  $b$  in position  $i$ , and  $i'$  is  $i + \varepsilon$  (with  $i + \varepsilon > 0$ ). We say that  $\alpha'$  is a *successor* of  $\alpha$ . We also assume that  $\mathcal{A}$  has only one reachable configuration  $(q, i, w)$  for which  $q = q_{acc}$ , and that  $i = 1$  and  $w = a^n$ .

A *run* of  $\mathcal{A}$  from some configuration  $\alpha_0$  is a tree, the root of which corresponds to  $\alpha_0$ , and where every node corresponding to  $\alpha$  has a child node for each successor  $\alpha'$  of  $\alpha$ . For  $k \in \mathbb{N}$ , a run rooted at some disjunctive configuration  $\alpha$  is *accepting in  $k$  steps* if and only if its state is  $q_{acc}$  or  $k \geq 1$  and at least *one* of its children is accepting in  $k - 1$  steps. A run rooted at some conjunctive configuration  $\alpha$  is accepting in  $k$  steps if and only if  $k \geq 1$  and *all* of its children is accepting in  $k - 1$  steps (and there is at least one child). A word  $v$  is accepted by  $\mathcal{A}$  if and only if there exists some  $k$  such that the run from  $(q_0, 1, v)$  is accepting in  $k$  steps. We say that  $\mathcal{A}$  is *linearly-bounded* (LB-ATM) on  $v$  if all configurations  $(q, i, w)$  in the run of  $\mathcal{A}$  have  $|w| \leq |v|$ . The problem of acceptance of a LB-ATM, which we denote by LB-ATM-ACCEPT, is written as:

**Input** An ATM  $\mathcal{A}$  and a word  $v \in \Gamma^*$  such that  $\mathcal{A}$  is linearly-bounded on  $v$ .

**Output** YES if and only if  $\mathcal{A}$  accepts  $v$ , NO otherwise.

A classical result says that the problem LB-ATM-ACCEPT is EXPTIME-complete [3]. In the following, we assume, as in [5], that *along a single branch of a run of an LB-ATM, no configuration is repeated*; thus every branch is finite. This assumption does not change the complexity issues: one can easily reduce an instance  $(\mathcal{A}, v)$  of LB-ATM-ACCEPT to some instance  $(\mathcal{A}', v')$  where  $\mathcal{A}'$  avoids repetitions by inserting on the tape a counter (encoded in binary) whose value is bounded by  $2^{|v|} \cdot |Q| \cdot |v|$  (the maximum number of distinct configurations along the run). Then  $\mathcal{A}'$  simulates the moves of  $\mathcal{A}$  and increases the counter by 1 for every simulated move of  $\mathcal{A}$ .

## 2 Concurrent Markov Decision Processes

A (discrete) probability *distribution* over a countable set  $Q$  is a function  $\mu : Q \rightarrow [0, 1]$  such that  $\sum_{q \in Q} \mu(q) = 1$ . For a possibly uncountable set  $Q'$ , let  $\text{Dist}(Q')$  be the set of distributions over countable subsets of  $Q'$ . A distribution  $\mu$  will occasionally be denoted by  $\{q \mapsto \mu(q) \mid q \in Q \text{ and } \mu(q) > 0\}$ . Given the distributions  $\mu_1, \dots, \mu_k$  over the sets  $Q_1, \dots, Q_k$ , respectively, the independent product  $\mu_1 \otimes \dots \otimes \mu_k$  is defined as  $\{(q_1, \dots, q_k) \mapsto \mu_1(q_1) \cdot \dots \cdot \mu_k(q_k) \mid (q_1, \dots, q_k) \in Q_1 \times \dots \times Q_k\}$ . A *Markov decision process* (MDP)  $\mathbf{M} = (\Sigma, S, D)$  comprises a set  $\Sigma$  of *actions*, a set  $S$  of *states*, and the *transition relation*  $D \subseteq S \times \Sigma \times \text{Dist}(S)$ . The transitions from state to state of an MDP are performed in two steps: given that the current state is  $s$ , the first step concerns a nondeterministic selection of an triple  $(s, a, \mu) \in D$  associated with  $s$ ; the second step comprises a probabilistic choice, made according to the distribution  $\mu$  of the chosen triple, as to which state to make the transition (that is, we move to a state  $s' \in S$  with probability  $\mu(s')$ ). We often write  $s \xrightarrow{a, \mu}$  instead of  $(s, a, \mu) \in D$ ; when  $\mu = \{s' \mapsto 1\}$ , we write  $s \xrightarrow{a} s'$ . An MDP is finite if  $\Sigma$ ,  $S$  and  $D$  are finite sets.

Unless stated otherwise, we henceforth assume that MDPs are finite.

A *finite path* is a finite sequence  $s_0 \xrightarrow{a_0}_{\mu_0} s_1 \xrightarrow{a_1}_{\mu_1} \cdots \xrightarrow{a_{n-1}}_{\mu_{n-1}} s_n$  of consecutive transitions followed by a state, such that  $\mu_i(s_{i+1}) > 0$  for all  $i < n$ . An *infinite path* is an infinite sequence  $s_0 \xrightarrow{a_0}_{\mu_0} s_1 \xrightarrow{a_1}_{\mu_1} \cdots$  of consecutive transitions, such that  $\mu_i(s_{i+1}) > 0$  for all  $i \in \mathbb{N}$ . A state  $s$  is *reached* along the path if there exists  $i \in \mathbb{N}$  such that  $s = s_i$ . An *adversary* of an MDP is a partial function mapping finite paths to triples  $(s, a, \mu) \in D$ , such that  $s$  is the state at the end of the path [7,17]. In the standard way, we define the probability measure  $Prob_s^A$  over measurable sets in the set of paths generated by adversary  $A$  from state  $s$  [10]. Given  $F \subseteq S$ , let  $Reach_s^A(F)$  be the set of paths generated by  $A$  from  $s$  along which a state in  $F$  is reached. For an MDP  $M = (\Sigma, S, D)$ , an *initial state*  $\bar{s} \in S$ , and a set  $F \subseteq S$  of *final states*, the *almost-sure reachability problem for MDPs* (MDP-ASR) consists in checking the existence of an adversary of  $M$  that assigns probability 1 to reaching  $F$  from  $\bar{s}$ , and can be solved in polynomial time in the size of  $M$ , independently of the transition probabilities (see, for example, [6]). Formally, MDP-ASR is written as:

**Input** An MDP  $M$ , an initial state  $\bar{s}$ , and a set of final states  $F$ .

**Output** YES if and only if there exists an adversary  $A$  of  $M$  such that  $Prob_{\bar{s}}^A\{Reach_{\bar{s}}^A(F)\} = 1$ , NO otherwise.

A *concurrent Markov decision process* (CMDP)  $\mathcal{M} = (M_1, \dots, M_k)$  is a  $k$ -tuple of Markov decision processes. The *flattening* of the concurrent Markov decision process  $\mathcal{M}$  is a Markov decision process  $(\Sigma, S, D)$ , where  $\Sigma = \Sigma_1 \cup \dots \cup \Sigma_k$ ,  $S = S_1 \times \dots \times S_k$ , and  $D$  is the set of all triples  $((s_1, \dots, s_k), a, \mu)$  from  $S \times \Sigma \times \text{Dist}(S)$  such that  $\mu = \mu_1 \otimes \dots \otimes \mu_k$ , where, for each  $1 \leq i \leq k$ , either  $(s_i, a, \mu_i) \in D_i$  or  $(a \notin \Sigma_i \text{ and } \mu_i = \{s_i \mapsto 1\})$  [16]. For a concurrent Markov decision process  $\mathcal{M} = (M_1, \dots, M_k)$  with the flattening  $M = (\Sigma, S, D)$ , an initial state  $(\bar{s}_1, \dots, \bar{s}_k) \in S$  and a set of final states  $F \subseteq S$  of  $\mathcal{M}$ , the *almost-sure reachability problem for CMDPs* (CMDP-ASR) is similar to MDP-ASR, but checks for the existence of an adversary on the flattening of the CMDP:

**Input** A CMDP  $\mathcal{M}$ , an initial state  $\bar{s}$ , and a set of final states  $F$ .

**Output** YES if and only if there exists an adversary  $A$  of the flattening  $M$  of  $\mathcal{M}$  such that  $Prob_{\bar{s}}^A\{Reach_{\bar{s}}^A(F)\} = 1$ , NO otherwise.

**Theorem 2.1** *The problem CMDP-ASR is EXPTIME-complete.*

*Proof.* An EXPTIME algorithm is obtained by applying standard polynomial time algorithms for MDP-ASR [6] over the (exponential) flattening of the CMDP in question. It remains to show the EXPTIME-hardness of CMDP-ASR. Let  $v \in \Gamma^n$  and  $\mathcal{A} = (Q, Q_\vee, Q_\wedge, \Gamma, \delta, q_0, q_{acc})$  be an LB-ATM. We define a CMDP  $\mathcal{M}_{\mathcal{A},v} = (M^{\text{ctrl}}, M_1, \dots, M_n)$  which models the run of  $\mathcal{A}$  over  $v$ :

- For each  $1 \leq i \leq n$ , the MDP  $M_i$  models the  $i$ th tape cell. The state set

is  $S_i = \{s_a^i, s_b^i\}$ , and the initial state is  $s_a^i$  if  $v(i) = a$ , and  $s_b^i$  otherwise. The alphabet is  $\Sigma_i = (\delta \times \{i\}) \cup \{(a, i), (b, i)\}$ . For each transition  $t = (q, e, q', e', \varepsilon) \in \delta$  such that  $i + \varepsilon \in \{1, \dots, n\}$ , there is a transition  $s_e^i \xrightarrow{t, i} s_{e'}^i$  in  $D_i$  to simulate the behavior of  $t$ . Furthermore, for each  $e \in \{a, b\}$ , there is a transition  $s_e^i \xrightarrow{e, i} s_e^i$  in  $D_i$  to indicate the current value of the cell.

- The MDP  $\mathbf{M}^{\text{ctrl}} = (\Sigma^{\text{ctrl}}, S^{\text{ctrl}}, D^{\text{ctrl}})$  models the control part of  $\mathcal{A}$ . The alphabet is  $\Sigma^{\text{ctrl}} = (\delta \times \{1, \dots, n\}) \cup (\{a, b\} \times \{1, \dots, n\})$ , and the state set is  $S^{\text{ctrl}} = (Q \times \{1, \dots, n\}) \cup (Q_\wedge \times \{1, \dots, n\} \times \delta)$ . The initial state is  $(q_0, 1)$ . The transition relation  $D^{\text{ctrl}}$  is defined as follows:
  - For each  $q \in Q_\vee$ , each  $1 \leq i \leq n$ , and each  $t \in \delta$ , a transition  $(q, i) \xrightarrow{t, i} (q', i + \varepsilon)$  is included in  $D^{\text{ctrl}}$  if  $t = (q, e, q', e', \varepsilon)$  and  $i + \varepsilon \in \{1, \dots, n\}$ .
  - For each  $q \in Q_\wedge$ , each  $1 \leq i \leq n$  and each  $e \in \{a, b\}$  such that the set  $T_{q, i, e} = \{(q, e, q', e', \varepsilon) \in \delta \mid i + \varepsilon \in \{1, \dots, n\}\}$  is non empty, we have a transition  $(q, i) \xrightarrow{e, i} \mu_{(q, i, e)}$  in  $D^{\text{ctrl}}$ , where  $\mu_{(q, i, e)}$  is the distribution (with equal probabilities) over the states  $(q, i, t)$  for all  $t \in T_{q, i, e}$ . Then we add transitions  $(q, i, t) \xrightarrow{t, i} (q', i + \varepsilon)$  to  $D^{\text{ctrl}}$  according to the definition of  $t$ .

The size of  $\mathcal{M}_{\mathcal{A}, v}$  is  $O(n \times |Q| \times |\delta|)$ , including the probabilities represented as the ratio of two integers encoded in binary, and the reduction can be done in logarithmic space. Now we show that  $\mathcal{A}$  accepts  $v$  if and only if CMDP-ASR returns YES for  $\mathcal{M}_{\mathcal{A}, v}$  with the initial state  $((q_0, 1), s_{v(1)}^1, \dots, s_{v(n)}^n)$ , and the set containing the single final state  $((q_{\text{acc}}, 1), s_a^1, \dots, s_a^n)$ . As the problem LB-ATM-ACCEPT is EXPTIME-hard, this will suffice to show the EXPTIME-hardness of CMDP-ASR.

In the following, for a given word  $w \in \Gamma^n$ , we write  $\mathbf{s}_w$  instead of  $s_{w(1)}^1, \dots, s_{w(n)}^n$ . Let  $\mathbf{M}_{\mathcal{A}, v} = (\Sigma, S, D)$  be the flattening of the CMDP  $\mathcal{M}_{\mathcal{A}, v}$ . Our first task is to construct a modified, action-less version of  $\mathbf{M}_{\mathcal{A}, v}$ , denoted by  $\bar{\mathbf{M}} = (\bar{S}, \bar{D})$ , so that we are better able to relate the transitions of  $\mathcal{M}_{\mathcal{A}, v}$  with those of  $\mathcal{A}$ . Intuitively, we obtain  $\bar{\mathbf{M}}$  by removing intermediate states of the form  $((q, i, t), \mathbf{s}_w)$  from  $\mathbf{M}_{\mathcal{A}, v}$ . Let  $\bar{S}_N \subseteq S$  be the set of states of  $\mathcal{M}_{\mathcal{A}, v}$  which have the component  $\mathbf{M}^{\text{ctrl}}$  in a state in the set  $Q_\vee \times \{1, \dots, n\}$ , and similarly let  $\bar{S}_P \subseteq S$  be the set of states for which  $\mathbf{M}^{\text{ctrl}}$  is in a state in  $Q_\wedge \times \{1, \dots, n\}$ . Then let  $\bar{S} = \bar{S}_N \cup \bar{S}_P$ . The transition relation  $\bar{D} \subseteq \bar{S} \times \text{Dist}(\bar{S})$  is defined as follows. For states  $((q, i), \mathbf{s}_w) \in \bar{S}_N$ , for each transition  $((q, i), \mathbf{s}_w) \xrightarrow{t, i} ((q', i'), \mathbf{s}_{w'})$  of  $D$  we have  $((q, i), \mathbf{s}_w) \rightarrow ((q', i'), \mathbf{s}_{w'})$  in  $\bar{D}$ . For states  $((q, i), \mathbf{s}_w) \in \bar{S}_P$ , observe that in  $\mathbf{M}_{\mathcal{A}, v}$  we have transitions  $((q, i), \mathbf{s}_w) \xrightarrow{w(i), i} \mu((q, i, t), \mathbf{s}_w)$ , and, from  $((q, i, t), \mathbf{s}_w)$ , there is a unique transition  $((q, i, t), \mathbf{s}_w) \xrightarrow{t, i} ((q', i'), \mathbf{s}_{w'})$ , where  $q'$ ,  $i'$  and  $w'$  depend on  $t$ . In  $\bar{\mathbf{M}}$  we skip the intermediate state  $((q, i, t), \mathbf{s}_w)$  and consider a transition  $((q, i), \mathbf{s}_w) \rightarrow_{\bar{\mu}}$  such that  $\bar{\mu}((q', i'), \mathbf{s}_{w'})$  equals  $\mu((q, i, t), \mathbf{s}_w)$  if there is a transition  $t$  such that  $((q, i, t), \mathbf{s}_w) \rightarrow ((q', i'), \mathbf{s}_{w'})$ , and 0 otherwise. We can verify that, for all states  $s \in \bar{S}$  and any  $F \subseteq \bar{S}$ , CMDP-ASR returns YES for  $\mathcal{M}_{\mathcal{A}, v}$ ,  $s$  and  $F$  if and only if MDP-ASR returns YES for  $\bar{\mathbf{M}}$ ,  $s$  and  $F$ .

Note that we can obtain an isomorphism between the configurations of  $\mathcal{A}$  and the states of  $\overline{\mathbf{M}}$ , which relates a configuration  $(q, i, w)$  of  $\mathcal{A}$  to a state  $((q, i), \mathbf{s}_w)$  of  $\overline{\mathbf{M}}$ . For configurations  $(q, i, w)$  and  $(q', i', w')$ , we have that  $(q', i', w')$  is a successor of  $(q, i, w)$  if and only if  $((q, i), \mathbf{s}_w) \rightarrow_{\mu} ((q', i'), \mathbf{s}_{w'})$ . Because no configuration is repeated along a branch of a run of  $\mathcal{A}$  (see page 3), the MDP  $\overline{\mathbf{M}}$  is acyclic (*i.e.* there does not exist a finite path  $s_0 \xrightarrow{a_0}_{\mu_0} s_1 \xrightarrow{a_1}_{\mu_1} \cdots \xrightarrow{a_{n-1}}_{\mu_{n-1}} s_n$  of  $\mathbf{M}$  such that  $s_0 = s_n$ ). Hence  $\overline{\mathbf{M}}$  has no infinite path.

Next, we introduce the *alternating reachability* problem (*ALT-REACH*) on  $\overline{\mathbf{M}}$ . First we consider the variant in  $k$  steps (*ALT-REACH- $k$* ):

**Input** An MDP  $\mathbf{M}$ , a partition of the states of  $\mathbf{M}$  into *disjunctive states*  $S_{\vee}$  and *conjunctive states*  $S_{\wedge}$ , an initial state  $s$ , and a set of final states  $F$ .

**Output** YES if and only if:

- $s \in S_{\vee}$  and either  $s \in F$  or  $k \geq 1$  and there exists a transition  $s \rightarrow_{\mu} s'$  such that *ALT-REACH- $(k-1)$*  returns YES on  $\mathbf{M}$ ,  $S_{\vee}$ ,  $S_{\wedge}$ ,  $s'$ , and  $F$ ;
  - $s \in S_{\wedge}$ ,  $k \geq 1$ , and, for all states  $s' \in S$ , we have that  $s \rightarrow_{\mu} s'$  implies that *ALT-REACH- $(k-1)$*  returns YES on  $\mathbf{M}$ ,  $S_{\vee}$ ,  $S_{\wedge}$ ,  $s'$ , and  $F$ ;
- and NO otherwise.

Then the answer to *ALT-REACH* is YES if and only if there exists some  $k$  such that the corresponding instance of *ALT-REACH- $k$*  is positive. We apply the problem *ALT-REACH* by letting the set of disjunctive and conjunctive states considered be equal to  $\overline{S}_N$  and  $\overline{S}_P$ , respectively. From the acyclic property of  $\overline{\mathbf{M}}$ , we have that the problem *MDP-ASR* outputs YES on  $\overline{\mathbf{M}}$ ,  $s$  and  $F$  if and only if *ALT-REACH* outputs YES on  $\overline{\mathbf{M}}$ ,  $\overline{S}_N$ ,  $\overline{S}_P$ ,  $s$  and  $F$ .<sup>2</sup> We claim that the following statements are equivalent:

- (1) *CMDP-ASR* returns YES on input  $\mathcal{M}_{\mathcal{A},v}$ ,  $((q_0, 1), \mathbf{s}_v)$ , and  $((q_{acc}, 1), \mathbf{s}_{a\dots a})$ ;
- (2) *MDP-ASR* returns YES on input  $\overline{\mathbf{M}}$ ,  $((q_0, 1), \mathbf{s}_v)$ , and  $((q_{acc}, 1), \mathbf{s}_{a\dots a})$ ;
- (3) *ALT-REACH* returns YES on input  $\overline{\mathbf{M}}$ ,  $((q_0, 1), \mathbf{s}_v)$ , and  $((q_{acc}, 1), \mathbf{s}_{a\dots a})$ ;
- (4) *LB-ATM-ACCEPT* returns YES on input  $\mathcal{A}$  and  $v$ .

The equivalence of statements (1) and (2), (statements (2) and (3), respectively) follows from the arguments relating *CMDP-ASR* and *MDP-ASR* (*MDP-ASR* and *ALT-REACH*, respectively) given above. The equivalence of statements (3) and (4) follows from the aforementioned isomorphism between configurations of  $\mathcal{A}$  and states of  $\overline{\mathbf{M}}$ . Hence, the *CMDP-ASR* problem and the acceptance problem for *LB-ATM* are equivalent, and thus the *CMDP-ASR* problem is EXPTIME-hard.  $\square$

<sup>2</sup> The proof can be done directly on the number of steps of the accepting runs, which is the same in both problems.

### 3 Probabilistic Timed Automata

In this section, we study the complexity of the almost-sure probabilistic reachability problem for probabilistic timed automata. We use standard notation from (probabilistic) timed automata, such as *clock valuations*  $val : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$  which are mappings from the set of clocks  $\mathcal{X}$  to the set of non-negative real numbers  $\mathbb{R}_{\geq 0}$ , and *clock constraints*  $\Psi_{\mathcal{X}}$  over  $\mathcal{X}$ . A *probabilistic timed automaton* (PTA)  $\mathbf{P} = (L, \mathcal{X}, prob)$  [11,9] is a tuple consisting of a finite set  $L$  of *locations*, a finite set  $\mathcal{X}$  of clocks, and a finite set  $prob \subseteq L \times \Psi_{\mathcal{X}} \times \text{Dist}(2^{\mathcal{X}} \times L)$  of *probabilistic edges*. A probabilistic edge  $(l, g, p) \in prob$  is a triple containing (1) a source  $l$  location, (2) a guard  $g$ , and (3) a probability distribution  $p$  which assigns probability to pairs of the form  $(X, l')$  for some clock reset  $X$  and target location  $l'$ . The semantics of  $\mathbf{P}$  is the action-less, infinite-state Markov decision process  $\mathbf{M}[\mathbf{P}] = (S, D)$ . The state set  $S = L \times \mathbb{R}_{\geq 0}^{\mathcal{X}}$  comprises location-valuation pairs. The transition relation  $D$  is defined as the smallest set such that  $((l, val), \delta, \mu) \in D$  if there exist  $\delta \in \mathbb{R}_{\geq 0}$  and a probabilistic edge  $(l, g, p) \in prob$  such that (1)  $val + \delta \models g$  and (2) for each  $(l', val') \in S$ , we have  $\mu(l', val') = \sum_{X \subseteq \mathcal{X} \ \& \ val' = (val + \delta)[X := 0]} p(X, l')$ .

Let  $\mathbf{0} \in \mathbb{R}_{\geq 0}^{\mathcal{X}}$  be the clock valuation which assigns 0 to all clocks in  $\mathcal{X}$ . For a probabilistic timed automaton  $\mathbf{P} = (L, \mathcal{X}, prob)$ , an initial location  $\bar{l} \in L$ , and a set  $L_F \subseteq L$  of *final locations*, the *almost-sure reachability problem for PTAs* consists in checking the existence of an adversary that assigns probability 1 to reaching  $L_F$  from  $(\bar{l}, \mathbf{0})$ . Formally, PTA-ASR is the problem written as:

**Input** A PTA  $\mathbf{P}$ , an initial location  $\bar{l}$ , and a set of final locations  $L_F$ .

**Output** YES if and only if there exists an adversary  $A$  of  $\mathbf{M}[\mathbf{P}]$  such that  $Prob_{(\bar{l}, \mathbf{0})}^A \{ \text{Reach}_{(\bar{l}, \mathbf{0})}^A(L_F \times \mathbb{R}_{\geq 0}^{\mathcal{X}}) \} = 1$ , NO otherwise.

Kwiatkowska *et al.* [11] show that the problem PTA-ASR can be solved in exponential time in the size of  $\mathbf{P}$  using a variant of the *region graph* technique for timed automata [2]. We now show that this bound is optimal.

**Theorem 3.1** *The problem PTA-ASR is EXPTIME-complete.*

*Proof.* Given that an EXPTIME algorithm has been presented previously, it remains to show the EXPTIME-hardness of PTA-ASR. Let  $\mathcal{A} = (Q, Q_{\vee}, Q_{\wedge}, \Gamma, \delta, q_0, q_{acc})$  be an LB-ATM and  $v$  be a word of length  $n$ . We define a PTA  $\mathbf{P}_{\mathcal{A}, v} = (L, \mathcal{X}, prob)$  which models the run of  $\mathcal{A}$  over  $v$ . Then we let  $L = (Q \times \{1, \dots, n\}) \cup \{\bar{l}, l_F\}$ , and  $\mathcal{X} = \{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n, y\}$ . The contents of the tape of  $\mathcal{A}$  are encoded by the relative values of the clocks  $x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n$ : cell  $i$  contains  $a$  if  $x_i = \bar{x}_i$ , and  $b$  if  $x_i < \bar{x}_i$ . Clock  $y$  is used to ensure the elapse of time.

The probabilistic edge relation  $prob$  of  $\mathbf{P}_{\mathcal{A}, v}$  is obtained in a similar way to

the transition relation of the CMDP of the proof of Theorem 2.1, as we now explain. The idea is that probabilistic edges emulate the transitions of  $\mathcal{A}$ : in particular, the guards of probabilistic edges from a given location  $(q, i)$  can test whether the current tape symbol is  $a$  or  $b$  by checking whether  $x_i = \bar{x}_i$  or  $x_i < \bar{x}_i$ , respectively. Furthermore, the writing of a symbol in a tape cell can be replicated by clock resets: for example, to represent the writing of  $a$  in cell  $i$ , we reset clocks  $x_i$  and  $\bar{x}_i$  to 0 (so that  $x_i = \bar{x}_i$ ), whereas to write  $b$  we reset only  $x_i$  (so that  $x_i < \bar{x}_i$ ). The target locations of the probabilistic edges are derived from the target states of the transition of  $\mathcal{A}$  involved in the definition of the probabilistic edge, and by the associated movement of the tape head.

In a location  $(q, i)$  derived from a disjunctive ATM state (that is  $q \in Q_\vee$ ), there will be a nondeterministic choice between probabilistic edges, each of which is derived from a transition of  $\mathcal{A}$  from  $q$ , and each of which will assign probability 1 to a single outcome. In contrast, in a location  $(q, i)$  derived from a conjunctive ATM state (that is  $q \in Q_\wedge$ ), there are at most two probabilistic edges, one of which has a guard testing whether the current tape symbol is  $a$  (using  $x_i = \bar{x}_i$ , as above), the other testing for  $b$  (using  $x_i < \bar{x}_i$ ). The probabilistic branching is done (with equal probability) over the various outcomes derived from the outgoing transitions of  $q$  labeled with  $a$  or  $b$ , respectively. To the guard of each probabilistic edge, we add the constraint  $y > 0$  to force some time to elapse, in order to ensure that a clock reset of  $\{x_j\}$  encodes the writing of  $b$  in cell  $j$ . Finally, we add the probabilistic edge  $(\bar{l}, y > 0, \{X_v, (q_0, 1) \mapsto 1\})$ , where  $X_v = \{x_i \mid v(i) = b\} \cup \{y\}$ , to encode the initialization of the input word  $v$  on the tape, and also the probabilistic edge  $((q_{acc}, 1), \bigwedge_{i=1}^n (x_i = \bar{x}_i), \{\emptyset, l_F \mapsto 1\})$ .

The size of the PTA  $\mathsf{P}_{\mathcal{A},v}$  is linear in  $|\mathcal{A}| \cdot |v|$ : we have  $|L| = |Q| \cdot |v| + 2$ ,  $|\mathcal{X}| = 2 \cdot |v| + 1$ , and the size of the probabilistic edge set *prob* – including the probabilities encoded in binary – is bounded by  $2 \cdot |v| \cdot |\delta|$ . The reduction can be done in logarithmic space. Then  $\mathcal{A}$  accepts  $v$  if and only if PTA-ASR returns YES on the input PTA  $\mathsf{P}_{\mathcal{A},v}$ , the initial location  $\bar{l}$ , and the set  $\{l_F\}$  comprising the single final location  $l_F$ . Hence PTA-ASR is EXPTIME-hard.  $\square$

## References

- [1] L. Aceto and F. Laroussinie. Is your model checker on time? On the complexity of model checking for timed modal logics. *Journal of Logic and Algebraic Programming*, 52-53:7–51, 2002.
- [2] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
- [3] A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.



- [4] E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT Press, 1999.
- [5] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.
- [6] L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *Proc. of the 10th Int. Conf. on Concurrency Theory (CONCUR'99)*, volume 1664 of *LNCS*, pages 66–81. Springer, 1999.
- [7] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent program. *ACM Trans. Program. Lang. Syst.*, 5(3):356–380, 1983.
- [8] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *Proc. of the 12th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.
- [9] H. E. Jensen. Model checking probabilistic real time systems. In *Proc. of the 7th Nordic Workshop on Programming Theory*, volume 86, pages 247–261. Chalmers Institute of Technology, 1996.
- [10] J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov Chains*. Graduate Texts in Mathematics. Springer, 2nd edition, 1976.
- [11] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 286:101–150, 2002.
- [12] M. Z. Kwiatkowska. Model checking for probability and time: from theory to practice. In *Proc. of the 18th IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 351–360. IEEE Computer Society, 2003.
- [13] F. Laroussinie and Ph. Schnoebelen. The state-explosion problem from trace to bisimulation equivalence. In *Proc. of the 3rd Inter. Conf. on Foundations of Software Science and Computation Structures (FoSSaCS 2000)*, volume 1784 of *LNCS*, pages 192–207. Springer, 2000.
- [14] M. L. Littman. Probabilistic propositional planning: Representations and complexity. In *Proc. of the 14th National Conf. on Artificial Intelligence and the 9th Innovative Applications of Artificial Intelligence Conf. (AAAI/IAAI'97)*, pages 748–754. AAAI Press/MIT Press, 1997.
- [15] M. L. Puterman. *Markov Decision Processes*. J. Wiley & Sons, 1994.
- [16] R. Segala and N. A. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.
- [17] M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of the 16th An. Symp. on Foundations of Computer Science (FOCS'85)*, pages 327–338. IEEE Computer Society Press, 1985.