Almost-Sure Model Checking of Infinite Paths in One-Clock Timed Automata

Christel Baier[1] Nathalie Bertrand[2] Patricia Bouyer[3,*] Thomas Brihaye[4] Marcus Größer[1]

[1] *Technische Universität Dresden, Germany*
{baier,groesser}@tcs.inf.tu-dresden.de

[2] *IRISA, INRIA Rennes, France*
nathalie.bertrand@irisa.fr

[3] *LSV, CNRS & ENS Cachan, France*
bouyer@lsv.ens-cachan.fr

[4] *Université de Mons-Hainaut, Belgium*
thomas.brihaye@umh.ac.be

Abstract

In this paper, we define two relaxed semantics (one based on probabilities and the other one based on the topological notion of largeness) for LTL *over infinite runs of timed automata which rule out unlikely sequences of events. We prove that these two semantics match in the framework of single-clock timed automata (and only in that framework), and prove that the corresponding relaxed model-checking problems are* PSPACE*-Complete. Moreover, we prove that the probabilistic non-Zenoness can be decided for single-clock timed automata in* NLOGSPACE.

1 Introduction

Nowadays *timed automata* [1] are a well-established formalism for the modelling and analysis of timed systems. Roughly speaking timed automata are finite-state automata enriched with clocks and clock constraints. This model has been extensively studied, and several verification tools have been developed. However, like most models used in model checking, timed automata are an idealized mathematical model. In particular it has infinite precision, instantaneous events, *etc*. Recently, more and more research has been devoted to propose alternative semantics for timed automata that provide more realistic operational models for real-time systems. Let us first mention the *Almost ASAP semantics* introduced in [14] and further studied in [13, 2, 8]. This AASAP semantics somewhat relaxes the constraints and precision of clocks. However, it induces a very strong notion of *robustness*, suitable for really critical systems, but maybe too strong for less critical systems. Another *"robust semantics"*, based on the notion of *tube acceptance*, has been proposed in [16, 17]. In this framework, a metric is put on the set of traces of the timed automaton, and roughly, a trace is robustly accepted if and only if a tube around that trace is classically accepted. This language-focused notion of acceptance is not completely satisfactory for implementability issues, because it does not take into account the structure of the automaton, and hence is not related to the most-likely behaviours of the automaton.

Varacca and Völzer recently proposed in [25] a *probabilistic framework for finite-state (time-abstract) systems* to overcome side-effects of modelling. They use probabilities to define the notion of being fairly correct as having probability zero to fail, when every non-deterministic choice has been transformed into a "reasonable" probabilistic choice. Moreover, in their framework, a system is fairly correct with respect to some property if and only if the set of traces satisfying that property in the system is topologically large, which somehow attests the relevance of this notion of fair correctness.

In the recent paper [4], we used similar concepts as in [25] and proposed two alternative semantics for reasoning about the *finite runs* of timed automata: *(i)* a *probabilistic semantics* which assigns probabilities both on delays and on discrete choices, and *(ii)* a *topological semantics*, following ideas of [16, 17] but rather based on the structure of the automaton than on its accepted language. For both semantics, we naturally addressed a model-checking problem for LTL interpreted over finite paths. We proved, by means of Banach-Mazur games, that both semantics coincide and that both model-checking problems for LTL specifications on finite words are PSPACE-Complete.

The purpose of this paper is to develop techniques for analyzing the *infinite behaviours* of timed automata by means of a probabilistic *almost-sure* interpretation of LTL over infinite runs (which requires that the given LTL formula φ holds with probability 1) and a topological interpretation (which requires topological largeness of the set of infinite runs where φ holds). The formal definitions of the almost-

[*] This work was carried out while this author was visiting Oxford University Computing Laboratory, and supported by a Marie Curie fellowship.

sure and topological semantics of LTL interpreted over the infinite runs in a timed automata are rather straightforward adaptions of the corresponding definitions in the case of finite runs [4]. However, to establish a link between the two semantics and to show that the topological semantics of LTL is reasonable in the sense that it matches the standard meaning of negation, the proof techniques used in [4] are no longer appropriate. Instead, methods are required that are specific for infinite runs. To confirm that the topological semantics yields a reasonable interpretation for LTL, we prove that the underlying topology constitutes a *Baire space*. For the case of one-clock timed automata, we will show that some kind of strong *fairness* is inherent in the almost-sure semantics. This observation will be used to prove that the almost-sure and topological semantics for infinite paths in one-clock timed automata agree. As the topological semantics only relies on the graph-structure of the given automaton (but not on any quantitative assumption on the resolution of the nondeterministic choices as it is the case for the probabilistic setting), this result yields the key to establish a polynomially space-bounded model checking algorithm for LTL over infinite words with respect to our non-standard semantics. In addition, we introduce a notion of probabilistic non-Zenoness, which requires that the set of Zeno runs have measure 0, and show that it has a simple topological characterization which can serve as a basis for a nondeterministic logarithmic space-bounded algorithm to checking probabilistic non-Zenoness. We also show that analogous results cannot be established for timed automata with two or more clocks, as then the probabilistic and topological semantics for LTL over infinite words do not agree.

Organisation of the paper. Section 2 summarizes our notations for timed automata, LTL and the relevant topological concepts. The probabilistic space and the topological space associated with a timed automaton together with the almost-sure and topological LTL semantics are defined in Section 3. The relation between the two semantics and the induced model checking problems are studied in Section 4. Probabilistic Zenoness is considered in Section 5. Most proofs are omitted in the paper, but they can be found in the corresponding research report [5].

2 Preliminaries

2.1 The timed automaton model

We assume the classical notions of clocks, clock valuations, and guards are familiar to the reader [1]. We denote by $\mathcal{G}(X)$ the set of guards over the finite set of clocks X, and AP a finite set of atomic propositions.

A *timed automaton* is a tuple $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{L})$ such that: (i) L is a finite set of locations, (ii) X is a finite set of clocks, (iii) $E \subseteq L \times \mathcal{G}(X) \times 2^X \times L$ is a finite set of edges, (iv) $\mathcal{I} : L \to \mathcal{G}(X)$ assigns an invariant to each location, and (v) $\mathcal{L} : L \to 2^{\mathsf{AP}}$ is a labelling function.

The semantics of a timed automaton $\mathcal{A}$ is a timed transition system whose states are pairs $(\ell, \nu) \in L \times \mathbb{R}_+^{|X|}$ with $\nu \models \mathcal{I}(\ell)$, and whose transitions are of the form $(\ell, \nu) \xrightarrow{\tau, e} (\ell', \nu')$ if there exists an edge $e = (\ell, g, Y, \ell')$ such that for every $0 \leq \tau' \leq \tau$, $\nu + \tau' \models \mathcal{I}(\ell)$, $\nu + \tau \models g$, $\nu' = [Y \leftarrow 0](\nu + t)$, and $\nu' \models \mathcal{I}(\ell')$. A finite (resp. infinite) *run* ϱ of $\mathcal{A}$ is a finite (resp. infinite) sequence of transitions, *i.e.*, $\varrho = s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \ldots$ We write $\mathsf{Runs}_f(\mathcal{A}, s_0)$ (resp. $\mathsf{Runs}(\mathcal{A}, s_0)$) for the set of finite runs (resp. infinite runs) of $\mathcal{A}$ from state s_0. If s is a state of $\mathcal{A}$ and $(e_i)_{1 \leq i \leq n}$ is a finite sequence of edges of $\mathcal{A}$, if $\mathcal{C}$ is a constraint over n variables $(t_i)_{1 \leq i \leq n}$, the *(symbolic) path* starting from s, determined by $(e_i)_{1 \leq i \leq n}$, and constrained by $\mathcal{C}$, is the following set of runs:

$$\pi_{\mathcal{C}}(s, e_1 \ldots e_n) = \{\varrho = s \xrightarrow{\tau_1, e_1} s_1 \ldots \xrightarrow{\tau_n, e_n} s_n \mid \varrho \in \mathsf{Runs}_f(\mathcal{A}, s) \text{ and } (\tau_i)_{1 \leq i \leq n} \models \mathcal{C}\}\,.$$

If $\mathcal{C}$ is equivalent to 'true', we simply write $\pi(s, e_1 \ldots e_n)$. Let $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1 \ldots e_n)$ be a finite symbolic path, we define the *cylinder* generated by $\pi_{\mathcal{C}}$ as

$$\mathsf{Cyl}(\pi_{\mathcal{C}}) = \{\varrho \in \mathsf{Runs}(\mathcal{A}, s) \mid \exists \varrho' \in \mathsf{Runs}_f(\mathcal{A}, s), \text{ finite prefix of } \varrho, \text{ s.t. } \varrho' \in \pi_{\mathcal{C}}\}\,.$$

In the following, we will also use infinite symbolic paths defined, given s a state of $\mathcal{A}$ and $(e_i)_{i \geq 1}$ an infinite sequence of edges, as:

$$\pi(s, e_1 \ldots) = \{\varrho = s \xrightarrow{\tau_1, e_1} s_1 \ldots \mid \varrho \in \mathsf{Runs}(\mathcal{A}, s)\}\,.$$

If $\varrho \in \mathsf{Runs}(\mathcal{A}, s)$, we write π_ϱ for the unique symbolic path containing ϱ. Given s a state of $\mathcal{A}$ and e an edge, we define $I(s, e) = \{\tau \in \mathbb{R}_+ \mid s \xrightarrow{\tau, e} s'\}$ and $I(s) = \bigcup_e I(s, e)$. The automaton $\mathcal{A}$ is *non-blocking* if, for every state s, $I(s) \neq \emptyset$.

2.2 The region automaton abstraction

The well-known region automaton construction is a finite abstraction of timed automata which can be used for verifying many properties like ω-regular untimed properties [1]. For lack of space, we do not redefine the region equivalence relation, and we write $R_{\mathcal{A}}$ for the set of (clock) regions of automaton $\mathcal{A}$. Here we use a slight modification of the original construction, which is still a timed automaton, and just serves to simplify the further technical developments.

If $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{L})$ be a timed automaton then the *region automaton* of $\mathcal{A}$ is the timed automaton $\mathsf{R}(\mathcal{A}) = (Q, X, T, \kappa, \lambda)$ such that $Q = L \times R_{\mathcal{A}}$ and:

- $\kappa((\ell,r)) = \mathcal{I}(\ell)$, and $\lambda((\ell,r)) = \mathcal{L}(\ell)$ for all $(\ell,r) \in L \times R_{\mathcal{A}}$;
- $T \subseteq (Q \times \mathsf{cell}(R_{\mathcal{A}}) \times E \times 2^X \times Q)$, and $(\ell,r) \xrightarrow{\mathsf{cell}(r''),e,Y} (\ell',r')$ is in T iff there exists $e = \ell \xrightarrow{g,Y} \ell'$ in E s.t. there exist $\nu \in r$, $\tau \in \mathbb{R}_+$ with $(\ell,\nu) \xrightarrow{\tau,e} (\ell',\nu')$, $\nu + \tau \in r''$ and $\nu' \in r'$ ($\mathsf{cell}(r'')$ is the smallest guard containing r'').

We recover the usual region automaton of [1] by labelling the transitions with 'e' instead of '$\mathsf{cell}(r''), e, Y$', and by interpreting $\mathsf{R}(\mathcal{A})$ as a finite automaton. The above timed interpretation satisfies strong timed bisimulation properties that we do not detail here. To every finite path $\pi((\ell,\nu), e_1 \dots e_n)$ in $\mathcal{A}$ corresponds a finite set of paths $\pi(((\ell,[\nu]),\nu), f_1 \dots f_n)$ in $\mathsf{R}(\mathcal{A})$, each one corresponding to a choice in the regions that are crossed. If ϱ is a run in $\mathcal{A}$, we denote $\iota(\varrho)$ its unique image in $\mathsf{R}(\mathcal{A})$. Note that if $\mathcal{A}$ is non-blocking, then so is $\mathsf{R}(\mathcal{A})$.

In the rest of the paper we assume that timed automata are non-blocking, even though general timed automata could also be handled (but at a technical extra cost).

2.3 The logic LTL

We consider the linear-time temporal logic LTL [21] defined inductively as:

$$\mathsf{LTL} \ni \varphi ::= p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \mathbf{X}\,\varphi$$

where $p \in \mathsf{AP}$ is an atomic proposition. We use classical shorthands like $\mathtt{tt} \overset{\text{def}}{=} p \vee \neg p$, $\mathtt{ff} \overset{\text{def}}{=} p \wedge \neg p$, $\mathbf{F}\,\varphi \overset{\text{def}}{=} \mathtt{tt}\,\mathbf{U}\,\varphi$, and $\mathbf{G}\,\varphi \overset{\text{def}}{=} \neg\mathbf{F}\,(\neg\varphi)$. We assume the reader is familiar with the semantics of LTL, that we interpret here on infinite runs of a timed automaton.

2.4 Largeness, meagerness and the Banach-Mazur game

We assume the reader is familiar with basic notions of topology (see *e.g.* [19]). However, we recall the more elaborate notions of *meagerness* and *largeness*. If $(A, \mathcal{T})$ is a topological space, a set $B \subseteq A$ is *nowhere dense* if the interior of the closure of B is empty. A set is *meager* if it is a countable union of nowhere dense sets, and a set is *large* if its complement is meager. For example, when considering $\mathbb{R}$ with the classical topology, any single point is a nowhere dense set, hence $\mathbb{Q}$ is meager and $\mathbb{R} \setminus \mathbb{Q}$ is large. These notions of meagerness and largeness have very nice characterizations in terms of Banach-Mazur games. A *Banach-Mazur game* is based on a topological space $(A, \mathcal{T})$ equipped with a family $\mathcal{B}$ of subsets of A such that: (1) $\forall B \in \mathcal{B}$, $\mathring{B} \neq \emptyset$ and (2) $\forall O \in \mathcal{T}$ s.t. $O \neq \emptyset$, $\exists B \in \mathcal{B}$, $B \subseteq O$. Given $C \subseteq A$, players alternate their moves choosing decreasing elements in $\mathcal{B}$, and build an infinite sequence $B_1 \supseteq B_2 \supseteq B_3 \cdots$. Player 1 wins the play if $\bigcap_{i=1}^{\infty} B_i \cap C \neq \emptyset$, otherwise Player 2 wins. The relation between Banach-Mazur games and meagerness is given in the following theorem.

Theorem 1 (Banach-Mazur [20]). *Player* 2 *has a winning strategy in the Banach-Mazur game with target set C if and only if C is meager.*

3 Probabilistic and Topological Semantics for Timed Automata

In [4], we defined two relaxed semantics for LTL over finite runs of timed automata: the *almost-sure* semantics, based on probabilities, and the *large* semantics, based on the topological notion of largeness. In this section, we extend, in a natural way these semantics to infinite runs of timed automata.

3.1 A probabilistic semantics for LTL

Let $\mathcal{A}$ be a timed automaton. As in [4], we assume probability distributions are given from every state s of $\mathcal{A}$ both over delays and over enabled moves. For every state s of $\mathcal{A}$, the probability measure μ_s over delays in $\mathbb{R}_+$ (equipped with the standard Borel σ-algebra) must satisfy several requirements. A first series, is denoted $(\star)$ in the sequel:

- $\mu_s(I(s)) = \mu_s(\mathbb{R}_+) = 1$,[1]
- Denoting λ the Lebesgue measure, if $\lambda(I(s)) > 0$, μ_s is equivalent[2] to λ on $I(s)$; Otherwise, μ_s is equivalent on $I(s)$ to the uniform distribution over points of $I(s)$.

This last condition denotes some kind of fairness w.r.t. enabled transitions, in that we cannot disallow one transition by putting a probability 0 to delays enabling that transition.

For technical reasons, we also ask for additional requirements (denoted $(\dagger)$ in the sequel):

- $(s,a,b) \mapsto \mu_s\big(\{d \mid s + d \in [a,b]\}\big)$ is continuous on $\{(s,a,b) \mid \exists e \text{ s.t. } [a,b] \subseteq I(s,e)\}$;
- If $s' = s + t$ for some $t \geq 0$, and $0 \notin I(s+t', e)$ for every $0 \leq t' \leq t$, then $\mu_s(I(s,e)) \leq \mu_{s'}(I(s',e))$;
- There is $0 < \lambda_0 < 1$ s.t. for every state s with $I(s)$ unbounded, $\mu_s([0,1/2]) \leq \lambda_0$.

[1] Note that this is possible, as we assume $\mathcal{A}$ is non-blocking, hence $I(s) \neq \emptyset$ for every state s of $\mathcal{A}$.

[2] Two measures ν and ν' are *equivalent* whenever for each measurable set A, $\nu(A) = 0 \Leftrightarrow \nu'(A) = 0$.

Remark 2. *The three last requirements are technical and needed to deal with infinite behaviours, but they are natural and easily satisfiable. For instance, a timed automaton equipped with uniform (resp. exponential) distributions on bounded (resp. unbounded) intervals satisfy these conditions. If we assume exponential distributions on unbounded intervals, the very last requirement corresponds to the bounded transition rate condition in [15], required to have reasonable and realistic behaviours.*

For every state s of $\mathcal{A}$, we also assume a probability distribution p_s over edges, such that for every edge e, $p_s(e) > 0$ iff e is enabled in s. As it is classically done for resolving non-determinism [24], we assume that p_s is given by weights on transitions: we associate with each edge e a weight $w(e) > 0$, and for every state s, for every edge e, $p_s(e) = 0$ if e is not enabled in s, and $p_s(e) = w(e)/(\sum_{e' \text{ enabled in } s} w(e'))$ otherwise. As a consequence, if s and s' are region equivalent, then for every edge e, $p_s(e) = p_{s'}(e)$. We then define a measure over finite symbolic paths from state s as

$$\mathbb{P}_{\mathcal{A}}(\pi(s, e_1 \dots e_n)) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2 \dots e_n))\, \mathrm{d}\mu_s(t)$$

where $s \xrightarrow{t} (s+t) \xrightarrow{e_1} s_t$, and we initialize with $\mathbb{P}_{\mathcal{A}}(\pi(s)) = 1$.[3] The formula for $\mathbb{P}_{\mathcal{A}}$ relies on the fact that the probability of taking transition e_1 at time t coincides with the probability of waiting t time units and then choosing e_1 among the enabled transitions, *i.e.*, $p_{s+t}(e_1)\mathrm{d}\mu_s(t)$. Note that, time passage and actions are independent events.

The value $\mathbb{P}_{\mathcal{A}}(\pi(s, e_1 \dots e_n))$ is the result of n successive one-dimensional integrals, but it can also be viewed as the result of an n-dimensional integral. Hence, we can easily extend the above definition to finite constrained paths $\pi_{\mathcal{C}}(s, e_1 \dots e_n)$ when $\mathcal{C}$ is Borel-measurable. This extension to constrained paths will allow to express (and thus later measure) various and rather complex sets of paths, for instance Zeno behaviours (see Section 5). The measure $\mathbb{P}_{\mathcal{A}}$ can then be defined on cylinders, letting $\mathbb{P}_{\mathcal{A}}(\mathsf{Cyl}(\pi)) = \mathbb{P}_{\mathcal{A}}(\pi)$ if π is a finite (constrained) symbolic path. Finally we extend $\mathbb{P}_{\mathcal{A}}$ in a standard and unique way to the σ-algebra generated by these cylinders, that we note $\Omega^s_{\mathcal{A}}$.

Proposition 3. *Let $\mathcal{A}$ be a timed automaton. For every state s, $\mathbb{P}_{\mathcal{A}}$ is a probability measure over $(\mathit{Runs}(\mathcal{A}, s), \Omega^s_{\mathcal{A}})$.*

Example 4. *Consider the timed automaton $\mathcal{A}$ depicted on Fig. 1, and assume for all states both uniform distributions over delays and discrete moves. If $s_0 = (\ell_0, 0)$ is the initial state, then $\mathbb{P}_{\mathcal{A}}(\mathit{Cyl}(\pi(s_0, e_1e_1))) = \mathbb{P}_{\mathcal{A}}(\pi(s_0, e_1e_1)) = 1/4$ and $\mathbb{P}_{\mathcal{A}}(\pi(s_0, {e_1}^{\omega})) = 0$.*

We have seen in [4] how to transfer probabilities from $\mathcal{A}$ to $\mathsf{R}(\mathcal{A})$, and proved the correctness of the transformation. Under the same hypotheses (for every state s in $\mathcal{A}$, $\mu^{\mathcal{A}}_s = \mu^{\mathsf{R}(\mathcal{A})}_{\iota(s)}$, and for every $t \in \mathbb{R}_+$ $p^{\mathcal{A}}_{s+t} = p^{\mathsf{R}(\mathcal{A})}_{\iota(s)+t}$) this correctness still holds in our case by definition of the probability measure (first on finite paths, then on cylinders, and finally on any measurable set of infinite runs).

Lemma 5. *Assume measures in $\mathcal{A}$ and in $\mathsf{R}(\mathcal{A})$ are related as above. Then, for every set S of runs in $\mathcal{A}$ we have: $S \in \Omega^s_{\mathcal{A}}$ iff $\iota(S) \in \Omega^{\iota(s)}_{\mathsf{R}(\mathcal{A})}$, and in this case $\mathbb{P}_{\mathcal{A}}(S) = \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\iota(S))$.*

We can therefore lift results proved on $\mathsf{R}(\mathcal{A})$ to $\mathcal{A}$. In the sequel, we write $\mathcal{A} = \mathsf{R}(\mathcal{A})$ when we consider a region automaton rather than a general timed automaton.

Given an infinite symbolic path π and an LTL formula φ, either all concretizations of π (*i.e.*, concrete runs $\varrho \in \pi$) satisfy φ, or they all do not satisfy φ. Hence, the set $\{\varrho \in \mathsf{Runs}(\mathcal{A}, s_0) \mid \varrho \models \varphi\}$ is measurable (in $\Omega^{s_0}_{\mathcal{A}}$), as it is an ω-regular property [26]. In the sequel, we write $\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi)$ for $\mathbb{P}_{\mathcal{A}}\{\varrho \in \mathsf{Runs}(\mathcal{A}, s_0) \mid \varrho \models \varphi\}$.

Definition 6. *Let φ be an LTL formula and $\mathcal{A}$ a timed automaton. We say that $\mathcal{A}$* almost-surely satisfies *φ from s_0, and we then write $\mathcal{A}, s_0 \mathrel{|\!\!\approx_{\mathbb{P}}} \varphi$, whenever $\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi) = 1$. The* almost-sure model-checking problem *asks, given $\mathcal{A}$, φ and s_0, whether $\mathcal{A}, s_0 \mathrel{|\!\!\approx_{\mathbb{P}}} \varphi$.*

Example 7. *Consider the timed automaton $\mathcal{A}$ of Fig. 1 again with both uniform distributions over delays and discrete moves in all states and initial state $s_0 = (\ell_0, 0)$. Then, $\mathcal{A}, s_0 \mathrel{|\!\!\approx_{\mathbb{P}}} \mathbf{F}\,(p_1 \wedge \mathbf{G}\,(p_1 \Rightarrow \mathbf{F}\,p_2))$. Indeed, in state (ℓ_0, ν) with $0 \le \nu \le 1$, the probability of firing e_2 (after some delay) is always $1/2$ (guards of e_1 and e_2 are the same, there is thus a uniform distribution over both edges), thus the location ℓ_1 is reached with probability 1. In ℓ_1, the transition e_3 will unlikely happen, because its guard $x = 1$ is much too "small" compared to the guard $x \ge 3$ of the transition e_4. The same phenomenon arises in location ℓ_2 between the transitions e_5 and e_6. In conclusion, the runs of the timed automaton $\mathcal{A}$ (from s_0) are almost surely following sequences of transitions of the form ${e_1}^* e_2 (e_4 e_5)^{\omega}$. Hence, with probability 1, the formula $\mathbf{F}\,(p_1 \wedge \mathbf{G}\,(p_1 \Rightarrow \mathbf{F}\,p_2))$ is satisfied. Note that the previous formula is not satisfied with the classical LTL semantics. Indeed several counter-examples to the satisfaction of the formula can be found:* 'staying in ℓ_0 forever'*,* 'reaching ℓ_3'*, etc... All these counter-examples are unlikely and vanish thanks to our probabilistic semantics.*

Although the values $\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi)$ depend on the chosen weights $p_s(e)$ and measures μ_s, we will see that for one-clock timed automata the almost-sure satisfaction relation is

[3] In [4] the definition was slightly different since we wanted the measure of all finite paths to be 1. We therefore used a normalisation factor $1/2$ so that the measure of all paths of length i were $1/2^{i+1}$.

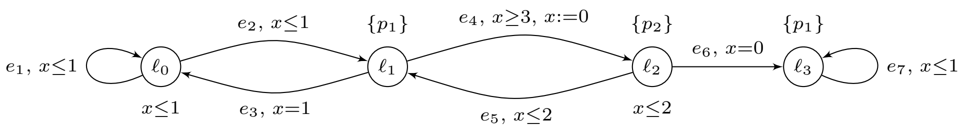


Fig. 1. A running example

not affected by the choice of the weights and distributions. This will be crucial for the decidability of the almost-sure model checking problem. The way to establish this result is to prove the equivalence of the almost-sure semantics with a topological semantics, which is defined on the basis of the so-called dimension of symbolic paths.

3.2 A topological semantics for LTL

In [4], we introduced a notion of dimension for finite constrained symbolic paths. Intuitively, a path is of *defined dimension* if it corresponds to a polyhedron of maximal dimension (in the space induced by the automaton). Formally, let $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1 \dots e_n)$ be a constrained path of a timed automaton $\mathcal{A}$. We define its associated polyhedron as follows:

$$\mathsf{Pol}(\pi_{\mathcal{C}}) = \{(\tau_i)_{1 \le i \le n} \in (\mathbb{R}_+)^n \mid \\ s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \in \pi_{\mathcal{C}}(s, e_1 \dots e_n)\}\,.$$

For each $0 \le i \le n$, we write $\mathcal{C}_i$ for the constraint induced by the projection of $\mathsf{Pol}(\pi_{\mathcal{C}})$ over the i first coordinates, with the convention that $\mathcal{C}_0$ is true. We say that the dimension of $\pi_{\mathcal{C}}$ is *undefined*, denoted $\dim_{\mathcal{A}}(\pi_{\mathcal{C}}) = \bot$, whenever there exists some index $1 \le i \le n$ with

$$\dim\Big(\mathsf{Pol}\big(\pi_{\mathcal{C}_i}(s, e_1 \dots e_i)\big)\Big) < \\ \dim\Big(\cup_e \, \mathsf{Pol}\big(\pi_{\mathcal{C}_{i-1}}(s, e_1 \dots e_{i-1} e)\big)\Big)\,.$$

Otherwise we say that the dimension of $\pi_{\mathcal{C}}$ is *defined*, denoted $\dim_{\mathcal{A}}(\pi_{\mathcal{C}}) = \top$.

The notion of dimension naturally extends to infinite symbolic paths: If $\pi = \pi(s, e_1 e_2 \dots)$ is an infinite symbolic path, its *dimension* is

$$\dim_{\mathcal{A}}(\pi) \;=\; \lim_{n \to \infty} \dim_{\mathcal{A}}(\pi(s, e_1 \dots e_n)).$$

Example 8. *On the automaton $\mathcal{A}$ of Fig. 1 with $s_0 = (\ell_0, 0)$, $\dim_{\mathcal{A}}(\pi(s_0, {e_1}^{\omega})) = \top$ and $\dim_{\mathcal{A}}(\pi(s_0, e_1(e_2e_3)^{\omega})) = \bot$.*

In the context of finite paths, a symbolic path has probability 0 iff it has an undefined dimension. In the context of infinite paths, this is no more true as infinite paths with defined dimension can have probability 0, like $\pi(s_0, e_1^{\omega})$ in the automaton of Fig. 1. However, writing $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{dim\_undef})$ for $\mathbb{P}_{\mathcal{A}}\{\varrho \in \mathsf{Runs}(\mathcal{A}, s) \mid \dim_{\mathcal{A}}(\varrho) = \bot\}$, the following holds:

Lemma 9. *If $\mathcal{A}$ is a timed automaton, for every state s in $\mathcal{A}$, $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{dim\_undef}) = 0$.*

Let $\mathcal{A}$ be a timed automaton, and s be a state of $\mathcal{A}$. Let $\mathcal{T}_{\mathcal{A}}^{s}$ be the topology over the set of runs of $\mathcal{A}$ starting in s defined with the following basic opens sets: either the set $\mathsf{Runs}(\mathcal{A}, s)$, or the cylinders $\mathsf{Cyl}(\pi_{\mathcal{C}})$ where $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1 e_2 \dots e_n)$ is a finite constrained symbolic path of $\mathcal{A}$ such that: (i) $\dim(\pi_{\mathcal{C}}) = \top$, (ii) $\mathcal{C}$ is convex (and Borel-measurable), and (iii) $\mathsf{Pol}(\pi_{\mathcal{C}})$ is open in $\mathsf{Pol}(\pi)$ for the classical topology on $\mathbb{R}^n$.

We first prove that our topological space is a *Baire space*:[4] indeed, in non Baire spaces, the notions of largeness and meagerness do not always make sense. For instance, in $\mathbb{Q}$ with the classical topology, which is not a Baire space, every set is both meager and large. Hence negation would have little meaning in our topological satisfaction. In Baire spaces, however, if a set is large its complement is not.

Proposition 10. *Let $\mathcal{A}$ be a timed automaton. For every state s of $\mathcal{A}$, the topological space $(\mathsf{Runs}(\mathcal{A}, s), \mathcal{T}_{\mathcal{A}}^{s})$ is a Baire space.*

Let us just mention that the proof of Proposition 10 (see the research report) heavily relies on the Banach-Mazur game but is not a consequence of the same result for finite runs [4].

Definition 11. *Let φ be an LTL formula and $\mathcal{A}$ a timed automaton. We say that $\mathcal{A}$* largely satisfies *φ from s_0, and we write $\mathcal{A}, s_0 \mathrel{|\!\!\approx_{\mathcal{T}}} \varphi$, if $\{\varrho \in \mathsf{Runs}(\mathcal{A}, s_0) \mid \varrho \models \varphi\}$ is topologically large. The* large model-checking problem *asks, given $\mathcal{A}$, φ and s_0, whether $\mathcal{A}, s_0 \mathrel{|\!\!\approx_{\mathcal{T}}} \varphi$.*

Example 12. *On the timed automaton $\mathcal{A}$ of Fig. 1 with initial state $s_0 = (\ell_0, 0)$, $\mathcal{A}, s_0 \mathrel{|\!\!\approx_{\mathcal{T}}} \mathbf{F}\,(p_1 \wedge \mathbf{G}\,(p_1 \Rightarrow \mathbf{F}\,p_2))$.*

Although the topological spaces given by $\mathcal{A}$ and $\mathsf{R}(\mathcal{A})$ are not homeomorphic, the topologies in $\mathcal{A}$ and in $\mathsf{R}(\mathcal{A})$ somehow match, as stated by the next proposition. This allows to lift result from $\mathsf{R}(\mathcal{A})$ to $\mathcal{A}$.

[4] Recall that a topological space $(A, \mathcal{T})$ is a *Baire space* if every non-empty open set in $\mathcal{T}$ is not meager (see [19, p.295]).

Proposition 13. *Let $\mathcal{A}$ be a timed automaton, and s a state of $\mathcal{A}$. Let $S \subseteq \mathsf{Runs}(\mathcal{A}, s)$. Then, S is large in $(\mathsf{Runs}(\mathcal{A}, s), \mathcal{T}^s_{\mathcal{A}})$ iff $\iota(S)$ is large in $(\mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s)), \mathcal{T}^{\iota(s)}_{\mathsf{R}(\mathcal{A})})$.*

4 The Two Semantics Match

We now prove that our two relaxed semantics match in the case of one-clock timed automata, and provide a decidability algorithm for the almost-sure (or equivalently large) LTL model-checking problem. It is however not a straightforward consequence of the same result for finite runs [4]. It is indeed rather involved and requires the development of techniques mixing classical probabilistic techniques and strong properties of one-clock timed automata. Note that these techniques only apply in the one-clock framework!

We first recall a construction made in [4] to decide the almost-sure model checking of LTL interpreted over finite paths. Any edge e in $\mathsf{R}(\mathcal{A})$ is colored in red if $\mu_s(I(s, e)) = 0$, and in blue otherwise. Then, a finite path in $\mathsf{R}(\mathcal{A})$ has an undefined dimension iff it crosses a red edge. Hence, having a defined (or undefined) dimension for a path can be specified locally in $\mathsf{R}(\mathcal{A})$. We say that a blue (resp. red) edge has a defined (resp. undefined) dimension. We call $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$ the restriction of $\mathsf{R}(\mathcal{A})$ to edges with defined dimension.

4.1 A notion of fairness

In the case of finite paths, if $\mathcal{A}$ satisfies an LTL property φ almost-surely, only paths of undefined dimension may not satisfy φ. Unfortunately, this is in general wrong for infinite paths. Indeed, on the timed automaton $\mathcal{A}$ of Fig. 1, when starting from $s = (\ell_0, 0)$, location ℓ_1 is clearly reached with probability 1. However the infinite path $\pi(s, {e_0}^\omega)$ has defined dimension although it never reaches ℓ_1. This kind of behaviours forces us to restrict our study to *fair* infinite paths, which is rather natural since probabilities and strong fairness are closely related in finite-state systems [22, 23, 7].

Let $\mathcal{A} = \mathsf{R}(\mathcal{A})$ be a timed automaton. An infinite region path $q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} q_2 \ldots$ in $\mathcal{A}$ is *fair* iff for every edge e with defined dimension, if e is enabled in infinitely many q_i with $i \in \mathbb{N}$, then $e_i = e$ for infinitely many $i \in \mathbb{N}$. Note that region paths and symbolic paths are closely related, as we assume $\mathcal{A} = \mathsf{R}(\mathcal{A})$: to any non-empty symbolic path $\pi(s, e_1e_2\ldots)$, we associate a unique region path $q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} q_2 \ldots$ with $s \in q_0$. Hence, we say that a symbolic path $\pi(s, e_1e_2\ldots)$ is fair whenever its corresponding region path is fair. Finally, we say that an infinite run ϱ is fair whenever π_ϱ is fair. Obviously, the set of fair infinite runs from s is $\Omega^s_{\mathcal{A}}$-measurable, as fairness is an ω-regular property over infinite paths. Writing $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{fair})$ for $\mathbb{P}_{\mathcal{A}}\{\varrho \in \mathsf{Runs}(\mathcal{A}, s) \mid \varrho \text{ is fair}\}$, we get the following property:

Lemma 14. *If $\mathcal{A}$ is a one-clock timed automaton, for every state s in $\mathcal{A}$, $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{fair}) = 1$.*

The proof of this lemma is involved, we just briefly sketch the main steps of the proof.

(i) We first prove that any edge with defined dimension is almost-surely taken infinitely often within a compact set (for the value of the unique clock), provided it is enabled infinitely often within that compact set.

(ii) Then, restricting to runs with infinitely many resets, those paths will pass infinitely often in a given configuration (because we only have one clock, hence resetting the clock and going to location q means entering the configuration $(q, 0)$). We can then apply the previous result, and get that any sequence of edges with defined dimension will be taken infinitely often with probability 1.

(iii) Concerning the runs ending up in the unbounded region (with no more resets of the clock), we prove that the distributions over edges correspond ultimately to a finite Markov chain, and hence that these runs are fair with probability 1.

(iv) Finally, restricting to runs ending up in a bounded region (with no more resets of the clock), only edges labelled with that precise region as a constraint can be enabled, and it will ultimately behave like a finite Markov chain, hence leading to the fairness property with probability 1.

We shortly argue why this lemma requires the restriction to single-clock timed automata. As pointed out in [10], timed automata admit various *time converging* behaviours, and some of these behaviours, not occurring in one-clock timed automata, can lead to "big" sets of *unfair* executions. Inspired by an example of [10], we design a two-clock timed automaton $\mathcal{A}$ (see Fig. 2) which does not satisfy Lemma 14. Let us describe the evolution of the clock y along an infinite path ϱ of $\mathcal{A}$. We denote by ν_n the valuation of the clock y when ϱ enters location ℓ_0 for the n-th time. One can easily check that (i) $\nu_n < 1$ and (ii) $\nu_n < \nu_{n+1}$, for $n \in \mathbb{N}$. Due to (i) all fair infinite paths have to visit both the top loop and the bottom loop infinitely often; while (ii) implies that the probability of taking the top loop decreases. More precisely, when $\mathcal{A}$ is equipped with uniform distributions, one can show that the probability to run forever through the cycle $\ell_0\,\ell_3\,\ell_4\,\ell_0$ is positive and therefore $\mathbb{P}_{\mathcal{A}}((\ell_0, 0, 0) \models \mathsf{fair}) < 1$.

4.2 Relating probabilities and fair symbolic paths

We now come to one of the main results of this paper:

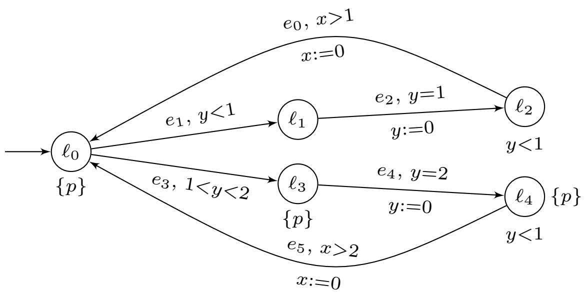


Fig. 2. A two-clock example with non negligible set of unfair runs

Theorem 15. (Relating probabilities and fair symbolic paths) *Let $\mathcal{A}$ be a one-clock (non-blocking) timed automaton such that $\mathcal{A} = \mathsf{R}(\mathcal{A})$, and φ be an* LTL *formula. If s is a state of $\mathcal{A}$, then $\mathbb{P}_{\mathcal{A}}(s \models \varphi) > 0$ iff there exists a fair infinite symbolic path $\pi = \pi(s, e_1 e_2 \ldots)$ such that $\dim_{\mathcal{A}}(\pi) = \top$, and $\pi \models \varphi$.*

Sketch of the proof. The left-to-right implication is an immediate consequence of Lemmas 9 and 14, as $\mathbb{P}_{\mathcal{A}}(s \models \varphi) = \mathbb{P}_{\mathcal{A}}(s \models \varphi \wedge \mathsf{fair} \wedge \neg \mathsf{dim\_undef})$. We quickly explain the right-to-left implication in the case of a prefix-indepdent location-based ω-regular property. The details of the complete proof and its extension to LTL properties can be found in the research report [5]. The fair infinite symbolic path π (with defined dimension) mentioned in the theorem ultimately ends up in a bottom strongly connected component (BSCC in short) of $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$. Any prefix π_{pref} of π ending in this BSCC has defined dimension, and it is easy to get that the probability of all fair infinite runs ϱ in $\mathsf{Cyl}(\pi_{\mathsf{pref}})$ with $\dim(\pi_\varrho) = \top$ is equal to the probability of $\mathsf{Cyl}(\pi_{\mathsf{pref}})$, hence is positive. All such fair infinite runs moreover satisfy the ω-regular property, hence the result. □

4.3 Relating probabilities and large sets of runs

We can now state the second main result of this paper, relating the almost-sure and the large semantics for LTL. In particular, this result shows that the almost-sure semantics does not depend on the concrete choice of the weights $p_s(e)$ and the measures μ_s.

Theorem 16. (Equivalence of the almost-sure and large semantics) *Let $\mathcal{A}$ be a one-clock (non-blocking) timed automaton, and φ an* LTL *formula. Let s be a state of $\mathcal{A}$. Then, $\mathcal{A}, s \approx_{\mathbb{P}} \varphi \Leftrightarrow \mathcal{A}, s \approx_{\mathcal{T}} \varphi$.*

Sketch of the proof. Thanks to Lemma 5, Corollary 13 and Theorem 15, it is sufficient to prove that, in $\mathsf{R}(\mathcal{A})$, the two following properties are equivalent: (1) the set $[\![\varphi]\!]_{\mathit{fair}}$ of fair infinite runs satisfying φ is large and (2) every fair infinite symbolic path π such that $\dim_{\mathcal{A}}(\pi) = \top$ satisfies φ. Indeed, using Banach-Mazur games, one can show that the set of fair runs is large, hence (1) is equivalent to "the set of paths satisfying φ is large".

We first prove that (2) implies (1). We prove that $[\![\varphi]\!]^c_{\mathit{fair}}$ (the complement of $[\![\varphi]\!]_{\mathit{fair}}$) is meager using Banach-Mazur games. We define $\mathcal{B}$, the family we play with, as the set of all basic open sets. A winning strategy for Player 2 (to avoid $[\![\varphi]\!]^c_{\mathit{fair}}$) is the following. After Player 1's first move $\mathsf{Cyl}(\alpha_0)$, Player 2 chooses $\mathsf{Cyl}(\alpha_1)$ such that α_0 is a strict prefix of α_1, and α_1 ends up in a BSCC B of $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$. Then, whatever Player 1 chooses, Player 2 can ensure that all possible edges with defined dimension of the BSCC B are visited infinitely often. Under that strategy, the outcome is either empty (in case constraints defining $(\alpha_i)_{i \geq 0}$ tend to the empty set), or included in an infinite symbolic path, which is fair, has defined dimension (because all chosen cylinders have defined dimension), and hence satisfies φ by hypothesis. Hence, its intersection with $[\![\varphi]\!]^c_{\mathit{fair}}$ is empty, which yields the expected result.

We now prove that (1) implies (2) (or more precisely its contrapositive). We assume that there exists a fair infinite path π such that $\dim(\pi) = \top$ and $\pi \not\models \varphi$, and show that the set $[\![\varphi]\!]_{\mathit{fair}}$ is not large. This fair infinite path π ends up in a BSCC of $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$. Let π' be the shortest prefix of π which ends in this BSCC. Then, as φ is location-based, every fair infinite path with prefix π' (*i.e.*, in $\mathsf{Cyl}(\pi')$) will not satisfy φ. Hence $[\![\varphi]\!]^c_{\mathit{fair}}$ is non-meager (because $(\mathsf{Runs}(\mathcal{A}, s), \mathcal{T}^s_{\mathcal{A}})$ is a Baire space) and $[\![\varphi]\!]_{\mathit{fair}}$ is not large. □

Remark 17. *Theorems 15 and 16 do not hold for general timed automata. Indeed for the two-clock example $\mathcal{A}$ of Fig. 2, with $s_0 = (\ell_0, 0, 0)$: (1) $\mathbb{P}(s_0 \models \mathbf{G}\, p) > 0$ but there is no fair path satisfying $\mathbf{G}\, p$, and (2) $\mathcal{A}, s_0 \approx_{\mathcal{T}} \mathbf{F}\, \neg p$ but $\mathcal{A}, s_0 \not\approx_{\mathbb{P}} \mathbf{F}\, \neg p$.*

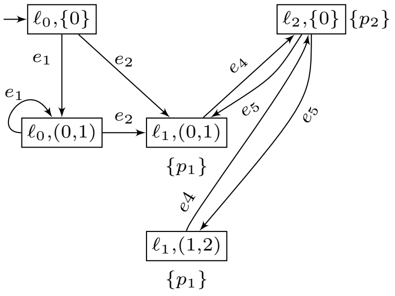
4.4 Decidability of the model-checking problems

Gathering the results of this section, and using an "optimized version" of one-dimensional regions [18] as well as the tricky automata-based approach of [11] for the LTL probabilistic verification problem, we get the following results for the two model-checking problems:

Corollary 18. (Decidability of the model-checking problems) *The almost-sure and large model-checking problems for one-clock timed automata are (i)* NLOGSPACE*-Complete for prefix-independent location-based ω-regular properties, and (ii)* PSPACE*-Complete for* LTL *properties.*

Example 19. *If we come back to the running example (see Fig. 1), the algorithm to decide the almost-sure model-*

checking constructs a subgraph of $\mathsf{R}(\mathcal{A})$ *(which is depicted below) in which all transitions with 'small' guards have been removed (it corresponds to* $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$*). The correctness of our algorithm then says that the original timed automaton satisfies almost-surely a property iff this automaton, interpreted as a finite Markov chain (with any distribution over edges) satisfies the property. Hence, in this example, it is easy to see that the Markov chain below satisfies the property* $\mathbf{F}\,(p_1 \wedge \mathbf{G}\,(p_1 \Rightarrow \mathbf{F}\,p_2))$ *with probability* 1*, hence the original timed automaton satisfies the above property almost-surely.*



5 A Note on Zeno Behaviours

In timed automata, and more generally in continuous-time models, some runs are *Zeno*.[5] These behaviours are problematic since they most of the time have no physical interpretation. As argued in [15], some fairness constraints are often put on executions, enforcing non-Zeno behaviours, but in probabilistic systems, probabilities are supposed to replace fairness assumptions, and it is actually the case in continuous-time Markov chains in which Zeno runs have probability 0 [6].

In our framework, it is hopeless to get a similar result because some timed automata are *inherently* Zeno. For instance, all runs are Zeno in the automaton consisting of a single location with a non-resetting loop guarded by $x \leq 1$. However, we show that we can decide whether the probability of the set of Zeno runs in a (one-clock) timed automaton is 0. We also give a nice characterization of the one-clock timed automata for which Zeno behaviours are negligible. This class is natural, since it corresponds to those automata which have no *'inherently Zeno components'* (reachable with a positive probability). Finally, we will see that the so-defined class encompasses classical definitions of *non-Zeno* timed automata.

We write $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{Zeno})$ for the probability of the set of Zeno runs in $\mathcal{A}$ from s. This set is measurable (in $\Omega^s_{\mathcal{A}}$), as it can be written as $\bigcup_{M\in\mathbb{N}}\bigcap_{n\in\mathbb{N}}\bigcup_{e_1,\cdots,e_n}\mathsf{Cyl}(\pi_{\mathcal{C}_{n,M}}(s, e_1 \ldots e_n))$ where $\mathcal{C}_{n,M}$ is the constraint $\sum_{1\leq i\leq n}\tau_i \leq M$.

Theorem 20. (Checking probabilistic non-Zenoness) *Given a single-clock (non-blocking) timed automaton* $\mathcal{A}$ *and a state* s *of* $\mathcal{A}$*, one can decide in* NLOGSPACE *whether* $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{Zeno}) = 0$*.*

Sketch of the proof. We first prove that the probability of the set of Zeno runs agrees with the probability of the set of runs with finitely many resets and ending in a bounded region. To decide whether such runs have positive measure, we show that it is sufficient to check whether there exists in $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$ a reachable 'Zeno BSCC' (*i.e.* a bounded BSCC with no reset edges). Reachability in a graph being in NLOGSPACE, the complexity follows. □

In Section 4, we gave a topological characterization of the probability of sets of runs defined by an LTL formula. Although Zeno runs cannot be defined in LTL, we obtain a similar result.

Theorem 21. (Topological characterization of probabilistic non-Zenoness) *Let* $\mathcal{A}$ *be a one-clock (non-blocking) timed automaton and* s *a state of* $\mathcal{A}$*. Then,* $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{Zeno}) = 0$ *iff the set of Zeno runs starting in* s *is meager.*

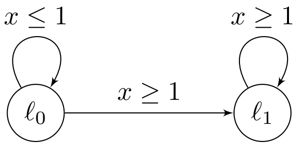
Sketch of proof. The proof is based on Banach-Mazur games after remembering the equivalence between almost-surely non Zeno and no Zeno BSCC in $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$ that was established in the proof of Theorem 20. □

Relation with classical non-Zenoness assumptions. The proof of Theorem 20 gives a characterization of automata for which the probability of Zeno runs is 0: they are those timed automata $\mathcal{A}$ in which there are no Zeno BSCCs in $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$. In the literature, several assumptions can be found, to handle Zeno runs. We pick two such assumptions, and show that our framework gives probability zero to Zeno runs under those restrictions.

In [3], the authors consider *strongly non-Zeno* automata: any cycle in the transition graph contains at the same time (1) a reset transition $x := 0$ and (2) a transition enabled only if $x \geq 1$, for some clock x. This condition removes all Zeno runs. As a consequence, for any strongly non-Zeno timed automaton (with n clocks) $\mathcal{A}$, $\mathbb{P}_{\mathcal{A}}(s \models \mathsf{Zeno}) = 0$. In [1], Alur and Dill want to decide the existence of non-Zeno accepted behaviours. They prove it is equivalent to having, in the region automaton, a reachable SCC (strongly connected component) satisfying the following *progress condition*: the SCC is either not bounded or with a reset of a clock. This condition is weaker than the strong non-Zenoness of [3] but is stronger than our condition on Zeno BSCC in $\mathcal{G}_{\mathsf{b}}(\mathcal{A})$. Below, we give a simple example to illustrate these claims, other examples can be found in the research report [5]. The automaton below, denote $\mathcal{A}_1$, is not *strongly non-Zeno* and does not satisfy the *progress condition*, however it satisfies $\mathbb{P}_{\mathcal{A}_1}(s \models \mathsf{Zeno}) = 0$. Let us notice that $\mathcal{A}_1$ does not satisfy

[5] A run $\varrho = s_0 \xrightarrow{\tau_1\cdot e_1} s_1 \xrightarrow{\tau_2\cdot e_2} \cdots$ of a timed automaton is *Zeno* if $\sum_{i=1}^{\infty}\tau_i < \infty$.

the *progress condition* since its region automaton contains a bounded SCC without resetting edges (which is not a bottom SCC).

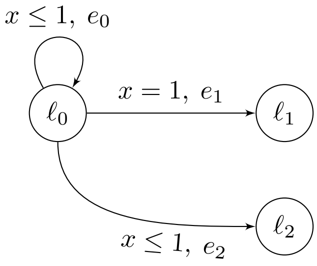


6 Conclusion

The goal of this paper was to present non-standard semantics for LTL interpreted over timed automaton that rule out "unlikely" events, but do not affect the decidability and complexity of the model checking problem. Our proofs do not use any specific feature of LTL, and also apply to the full class of ω-regular properties. We introduced a probabilistic almost-sure semantics that relies on some mild stochastic assumptions about the delays and the resolution of the non-deterministic choices, and a topological semantics based on the notion of largeness. For one-clock timed automata we proved the equivalence of the two semantics. The topological characterization of the almost-sure semantics has several important consequences: first, it shows that the precise choice of the measures used in the definition of the almost-sure semantics are irrelevant and second, as the topology is defined by the local conditions (using the notion of dimension), it yields a graph-based model-checking algorithm.

Although the formal definitions of the probabilistic and topological semantics reuse concepts of [4], where similar questions have been studied when interpreting LTL over finite words, the results for LTL over infinite words presented in this paper cannot be viewed as consequences of [4]. This becomes clear from the observation that the almost-sure and topological semantics for LTL over infinite words do not agree for timed automata with two or more clocks, while the approach of [4] does not impose any restrictions on the number of clocks. In fact, our proof for the topological dimension-based characterization of the almost-sure semantics LTL over infinite words in one-clock timed automata relies on a combination of techniques for the analysis of probabilistic systems with properties that are specific for timed automata with a single clock. Moreover, for one-clock timed automata, we obtain a nice characterization of timed automata having non-Zeno behaviours with probability one, and show that it can be decided in NLOGSPACE if an automaton has this property.

In some cases, the interpretation we give to transitions with singular guards might not correspond to what we want to model: for instance, in the automaton below, the transition e_1 might correspond to a deadline, and it could be unrealistic to consider it as unlikely to happen (somehow, if transition e_2 has not been taken within the first time unit, then transition e_1 will be taken at the end of the invariant). We could easily adapt our semantics to put a non-zero probability to that transition, and technics developed in this paper could easily be extended to that case.



As future works, we obviously plan to study the general case of n-clock timed automata. We will also look at timed games and see how probabilities can help simplify the techniques (used for instance in [12, 9]) for handling Zeno behaviours.

Acknowledgment: We are grateful to Nicolas Markey for insightful discussions about complexity classes.

References

[1] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[2] R. Alur, S. La Torre, and P. Madhusudan. Perturbed timed automata. In *Proc. 8th International Workshop on Hybrid Systems: Computation and Control (HSCC'05)*, vol. 3414 of *Lecture Notes in Computer Science*, pp. 70–85. Springer, 2005.

[3] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symposium on System Structure and Control*, pp. 469–474. Elsevier Science, 1998.

[4] C. Baier, N. Bertrand, P. Bouyer, T. Brihaye, and M. Größer. Probabilistic and topological semantics for timed automata. In *Proc. 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)*, vol. 4855 of *Lecture Notes in Computer Science*, pp. 179–191. Springer, 2007.

[5] C. Baier, N. Bertrand, P. Bouyer, T. Brihaye, and M. Größer. Almost-sure model checking of infinite paths in one-clock timed automata. Research Report LSV-08-13, Laboratoire Spécification et Vérification, ENS Cachan, France, 2008.

[6] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):524–541, 2003.

[7] C. Baier and M. Z. Kwiatkowska. On the verification of qualitative properties of probabilistic processes under fairness constraints. *Information Processing Letters*, 66(2):71–79, 1998.

[8] P. Bouyer, N. Markey, and P.-A. Reynier. Robust model-checking of timed automata. In *Proc. 7th Latin American Symposium on Theoretical Informatics (LATIN'06)*, vol. 3887 of *Lecture Notes in Computer Science*, pp. 238–249. Springer, 2006.

[9] T. Brihaye, T. A. Henzinger, V. Prabhu, and J.-F. Raskin. Minimum-time reachability in timed games. In *Proc. 34th International Colloquium on Automata, Languages and Programming (ICALP'07)*, vol. 4596 of *Lecture Notes in Computer Science*, pp. 825–837. Springer, 2007.

[10] F. Cassez, T. A. Henzinger, and J.-F. Raskin. A comparison of control problems for timed and hybrid systems. In *Proc. 5th International Workshop on Hybrid Systems: Computation and Control (HSCC'02)*, vol. 2289 of *Lecture Notes in Computer Science*, pp. 134–148. Springer, 2002.

[11] J.-M. Couvreur, N. Saheb, and G. Sutre. An optimal automata approach to LTL model checking of probabilistic systems. In *Proc. 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'03)*, vol. 2850 of *Lecture Notes in Computer Science*, pp. 361–375. Springer, 2003.

[12] L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *Proc. 14th International Conference on Concurrency Theory (CONCUR'03)*, vol. 2761 of *Lecture Notes in Computer Science*, pp. 142–156. Springer, 2003.

[13] M. De Wulf, L. Doyen, N. Markey, and J.-F. Raskin. Robustness and implementability of timed automata. In *Proc. Joint Conference on Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant System (FORMATS+FTRTFT'04)*, vol. 3253 of *Lecture Notes in Computer Science*, pp. 118–133. Springer, 2004.

[14] M. De Wulf, L. Doyen, and J.-F. Raskin. Almost ASAP semantics: From timed models to timed implementations. In *Proc. 7th International Workshop on Hybrid Systems: Computation and Control (HSCC'04)*, vol. 2993 of *Lecture Notes in Computer Science*, pp. 296–310. Springer, 2004.

[15] J. Desharnais and P. Panangaden. Continuous stochastic logic characterizes bisimulation of continuous-time Markov processes. *Journal of Logic and Algebraic Programming*, 56:99–115, 2003.

[16] V. Gupta, T. A. Henzinger, and R. Jagadeesan. Robust timed automata. In *Proc. International Workshop on Hybrid and Real-Time Systems (HART'97)*, vol. 1201 of *Lecture Notes in Computer Science*, pp. 331–345. Springer, 1997.

[17] T. A. Henzinger and J.-F. Raskin. Robust undecidability of timed and hybrid systems. In *Proc. 3rd International Workshop on Hybrid Systems: Computation and Control (HSCC'00)*, vol. 1790 of *Lecture Notes in Computer Science*, pp. 145–159. Springer, 2000.

[18] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking timed automata with one or two clocks. In *Proc. 15th International Conference on Concurrency Theory (CONCUR'04)*, vol. 3170 of *Lecture Notes in Computer Science*, pp. 387–401. Springer, 2004.

[19] J. R. Munkres. *Topology*. Prentice Hall, 2nd edition, 2000.

[20] J. C. Oxtoby. The Banach-Mazur game and Banach category theorem. *Annals of Mathematical Studies*, 39:159–163, 1957. Contributions to the Theory of Games, volume 3.

[21] A. Pnueli. The temporal logic of programs. In *Proc. 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*, pp. 46–57. IEEE Computer Society Press, 1977.

[22] A. Pnueli. On the extremely fair treatment of probabilistic algorithms. In *Proc. 15th Annual Symposium on Theory of Computing (STOC'83)*, pp. 278–290. ACM Press, 1983.

[23] A. Pnueli and L. D. Zuck. Probabilistic verification. *Information and Computation*, 103(1):1–29, 1993.

[24] C. Tofts. Processes with probabilities, priority and time. *Formal Aspects of Computing*, 6(5):536–564, 1994.

[25] D. Varacca and H. Völzer. Temporal logics and model checking for fairly correct systems. In *Proc. 21st Annual Symposium on Logic in Computer Science (LICS'06)*, pp. 389–398. IEEE Computer Society Press, 2006.

[26] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. 26th Symposium on Foundations of Computer Science (FOCS'85)*, pp. 327–338. IEEE Computer Society Press, 1985.