Probabilistic and Topological Semantics for Timed Automata

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

[1] Technische Universität Dresden, Germany
[2] LSV - CNRS & ENS Cachan, France
[3] Oxford University, England

Abstract. Like most models used in model-checking, timed automata are an idealized mathematical model used for representing systems with strong timing requirements. In such mathematical models, properties can be violated, due to unlikely (sequences of) events. We propose two new semantics for the satisfaction of LTL formulas, one based on probabilities, and the other one based on topology, to rule out these sequences. We prove that the two semantics are equivalent and lead to a PSPACE-Complete model-checking problem for LTL over finite executions.

1 Introduction

Timed automata, a model for verification. In the 90's, Alur and Dill proposed timed automata [3] as a model for verification purposes, which takes into account real-time constraints. With this model, one can express constraints on (possibly relative) dates of events. One of the fundamental properties of this model is that, though there are infinitely many possible configurations in the system, many verification problems can be solved (*e.g.* reachability and safety properties, branching-time timed temporal properties). Since then, this model has been intensively studied, and several verification tools have been developed.

Idealization of mathematical models. Timed automata are an idealized mathematical model, in which several assumptions are implicitely made: it has infinite precision, instantaneous events, *etc.* Several ideas have been explored to overcome the fact that these hypotheses are in practice unrealistic. The model of implementable controllers has been proposed, where constraints and precision of clocks are somewhat relaxed [8]. In this framework, if the model satisfies a safety property, then, on a simple model of processor, its implementation will also satisfy this property. This implementation model has been considered in [15, 7, 4, 6]. However, it induces a very strong notion of robustness, suitable for really critical systems (like rockets or X-by-wire systems in cars), but maybe too strong for less critical systems (like mobile phones or network applications).

⋆ Partly supported by a Lavoisier fellowship.
⋆⋆ Partly supported by a Marie Curie fellowship.

Another robustness model has been proposed at the end of the 90's in [9] with the notion of tube acceptance: a metric is put on the set of traces of the timed automaton, and a trace is robustly accepted if and only if a tube around that trace is classically accepted. This acceptance has been further studied for language-based properties, for instance the universality problem [11]. However, 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.

Using probabilities to alleviate the disadvantages of mathematical models. In their recent paper [17], Varacca and Völzer propose a probabilistic framework for finite-state 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.

Contribution. We address both motivations, ruling out unlikely sequences of transitions (as in the approach of [17]) and ruling out unlikely events (from a time point of view, as in the implementability paradigm discussed above). In order to do so, we propose two alternative semantics for timed automata: (*i*) a *probabilistic semantics* which assigns probabilities both on delays and on discrete choices, and (*ii*) a *topological semantics*, following ideas of [9, 11] but rather based on the structure of the automaton than on its accepted language. For both semantics, we can naturally address a model-checking problem for LTL: almost-sure model-checking for the probabilistic case and large model-checking for the topological case. Our results in these new frameworks are twofold. First we prove, by means of Banach-Mazur games, that the two semantics coincide: an LTL formula is almost-surely satisfied if and only if it is largely satisfied. Second we show that the almost-sure model-checking problem (and hence the large model-checking problem) for LTL specifications is PSPACE-Complete, *i.e.*, no more expensive than the classical LTL model-checking problem.

About probabilistic timed systems. Probabilities are not new in the model-checking community, and neither are timed systems. Several pieces of work even combine both. We refer to [16] for a survey on probabilistic timed systems. However, all of them were designed for modelling and analysing stochastic hybrid systems under quantitative aspects, whereas we aim at a probabilistic interpretation of non-probabilistic systems, which rule out unlikely events and yield a non-standard but still purely qualitative satisfaction relation for linear-time properties. To the best of our knowledge, we present here the first attempt to provide a probabilistic interpretation for non probabilistic timed systems in order to establish linear-time properties assuming 'fairness' on actions and delays.

Detailed proofs and complements can be found in the research report [5].

2 Timed Automata and Region Automata

In this section, we recall the classical notions of *timed automaton* and its well-known abstraction, the *region automaton* [3].

Timed automata. Let X be a finite set of *clocks*. A *clock valuation* over X is a mapping $\nu : X \to \mathbb{R}_+$, where $\mathbb{R}_+$ is the set of nonnegative reals. We write $\mathbb{R}_+^X$ for the set of clock valuations over X. If $\nu \in \mathbb{R}_+^X$ and $\tau \in \mathbb{R}_+$, $\nu + \tau$ is the clock valuation defined by $(\nu + \tau)(x) = \nu(x) + \tau$ if $x \in X$. If $Y \subseteq X$, the valuation $[Y \leftarrow 0]\nu$ is the valuation assigning 0 to $x \in Y$ and $\nu(x)$ to $x \notin Y$. A *guard* over X is a finite conjunction of expressions of the form $x \sim c$ where $x \in X$, $c \in \mathbb{N}$, and $\sim \in \{<, \leq, =, \geq, >\}$. We denote by $\mathcal{G}(X)$ the set of guards over X. The satisfaction relation for guards over clock valuations is defined in a natural way, and we write $\nu \models g$, if ν satisfies g. We denote AP a finite set of atomic propositions.

Definition 1. *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 the labelling function.*

The semantics of a timed automaton $\mathcal{A}$ is given by a labelled transition system $T_{\mathcal{A}} = (S, E \cup \mathbb{R}_+, \to)$ where the set S of states is $\{s = (\ell, \nu) \in L \times \mathbb{R}_+^X \mid \nu \models \mathcal{I}(\ell)\}$, and the transition relation $\to$ $(\subseteq S \times (E \cup \mathbb{R}_+) \times S)$ is composed of:

- *(delay transition)* $(\ell, \nu) \xrightarrow{\tau} (\ell, \nu + \tau)$ if $\tau \in \mathbb{R}_+$ and for all $0 \leq \tau' \leq \tau$, $\nu + \tau' \models \mathcal{I}(\ell)$,
- *(discrete transition)* $(\ell, \nu) \xrightarrow{e} (\ell', \nu')$ if $e = (\ell, g, Y, \ell') \in E$ is such that $\nu \models \mathcal{I}(\ell) \wedge g$, $\nu' = [Y \leftarrow 0]\nu$, and $\nu' \models \mathcal{I}(\ell')$.

A *finite run* ϱ of $\mathcal{A}$ is a finite sequence of states obtained by alternating delay and discrete transitions, *i.e.*, $\varrho = s_0 \xrightarrow{\tau_1} s'_1 \xrightarrow{e_1} s_1 \xrightarrow{\tau_2} s'_2 \xrightarrow{e_2} s_2 \cdots s_{n-1} \xrightarrow{\tau_n} s'_n \xrightarrow{e_n} s_n$ or more compactly $s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \cdots s_{n-1} \xrightarrow{\tau_n, e_n} s_n$. We write $\mathsf{Runs}(\mathcal{A}, s_0)$ for the set of finite runs of $\mathcal{A}$ from state s_0.

Given $s \in S$ and e an edge, we denote by $I(s, e) = \{\tau \in \mathbb{R}_+ \mid s \xrightarrow{\tau, e} s'\}$ and $I(s) = \bigcup_e I(s, e)$. The timed automaton $\mathcal{A}$ is said *non-blocking* whenever for every state $s \in S$, $I(s) \neq \emptyset$.

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 convex constraint over n real-valued 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:

$$\pi_{\mathcal{C}}(s, e_1 \dots e_n) = \{\varrho = s \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \cdots \mid \varrho \in \mathsf{Runs}(\mathcal{A}, s) \text{ and } (\tau_i)_{1 \leq i \leq n} \models \mathcal{C}\,[4]\}\,.$$

If $\mathcal{C}$ is equivalent to 'true', we write $\pi(s, e_1 \dots e_n)$, and say it is *unconstrained*. Occasionally, we refer to symbolic path for unconstrained symbolic path.

[4] We write $(\tau_i)_{1 \leq i \leq n} \models \mathcal{C}$ whenever the system $\mathcal{C}[t_i/\tau_i]$, obtained by replacing each variable t_i in $\mathcal{C}$ by the value τ_i, is true.

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, for instance regular untimed properties [3]. Roughly, the region automaton of $\mathcal{A}$ is the quotient of $T_{\mathcal{A}}$ by an equivalence relation over clock valuations. For lack of space, we do not redefine the region equivalence relation, and we write $R_{\mathcal{A}}$ for the set of regions of automaton $\mathcal{A}$. In this paper, we will use a slight modification of the original construction, which is still a timed automaton, but which satisfies very strong properties.

Definition 2. *Let $\mathcal{A} = (L, X, E, \mathcal{I}, \mathcal{L})$ be a timed automaton. The* region automaton *of $\mathcal{A}$ is the timed automaton $\mathsf{R}(\mathcal{A}) = (Q, X, T, \kappa, \lambda)$ such that:*

- *$Q = L \times R_{\mathcal{A}}$; $-$ $\kappa((\ell, r)) = \mathcal{I}(\ell)$, 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 [3] by labelling the transitions 'e' instead of '$\mathsf{cell}(r''), e, Y$', and by interpreting $\mathsf{R}(\mathcal{A})$ as a finite automaton. However, 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 write $\iota(\varrho)$ for its (unique) image in $\mathsf{R}(\mathcal{A})$. Finally, note that if $\mathcal{A}$ is non-blocking, then so is $\mathsf{R}(\mathcal{A})$.

In the rest of the paper we assume timed automata are non-blocking, even though general timed automata could also be handled (but at a technical extra cost). In all examples, if a state has no outgoing transition, we implicitely add a self-loop on that state with no constraints, so that the automaton is non-blocking.

3 A Probabilistic Semantics for Timed Automata

In the literature, several models gather probabilities and timed constraints (see [16] for a survey). Here, we take the model of timed automata, and give a probabilistic interpretation to delays, so that unlikely events will happen with probability 0.

For the rest of this section, we fix a timed automaton $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I}, \mathcal{L})$, which we assume is non-blocking. For every state s of $\mathcal{A}$, we assume a probability measure μ_s over $\mathbb{R}_+$ with the following requirements: (i) $\mu_s(I(s)) = \mu_s(\mathbb{R}_+) = 1$;[5] (ii) Writing λ for the Lebesgue measure, if $\lambda(I(s)) > 0$, μ_s is equivalent[6] to λ on $I(s)$; Otherwise, μ_s is equivalent on $I(s)$ to the uniform distribution over points of $I(s)$. 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 enabled in s (*i.e.*, $s \xrightarrow{e} s'$ for some s').

[5] Note that this is possible, as we assume s is non-blocking, hence $I(s) \neq \emptyset$.

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

Remark 3. The above constraints on probability measures are rather loose and are for instance satisfied by: (*i*) the uniform discrete distribution over $I(s)$ if $I(s)$ is a finite set of points, (*ii*) the Lebesgue measure over $I(s)$, normalized to have a probability measure, if $I(s)$ is a finite set of bounded intervals, and (*iii*) an exponential distribution if $I(s)$ contains an unbounded interval.

3.1 Definition of a Probability Measure over Finite Paths

Definition 4. *Let $\mathcal{A}$ be a timed automaton. We define inductively the probability for an unconstrained symbolic path $\pi(s, e_1 \dots e_n)$ to be fired (or equivalently for the sequence $e_1, \dots, e_n$ of transitions in $\mathcal{A}$ to be fired from s) as follows:*

$$\mathbb{P}_{\mathcal{A}}(\pi(s, e_1 \dots e_n)) = \frac{1}{2} \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2 \dots e_n))\, \mathrm{d}\mu_s(t)$$

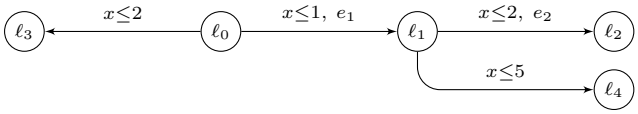
where $s \xrightarrow{t} (s+t) \xrightarrow{e_1} s_t$. We initialize with $\mathbb{P}_{\mathcal{A}}(\pi(s)) = \frac{1}{2}$.

Using Fubini's theorem, by induction on the length of symbolic paths, we can prove that $\mathbb{P}_{\mathcal{A}}$ is well-defined. When clear from the context, we omit subscript $\mathcal{A}$.

The formula for $\mathbb{P}_{\mathcal{A}}$ can be read as follows: the probability of taking transition e_1 at time t coincides with the probability of waiting t time units and then choose e_1 among the enabled transitions, *i.e.*, $p_{s+t}(e_1)\mathrm{d}\mu_s(t)$. We need to sum up over all t's in $I(s, e_1)$ the probability of runs starting by such a move. Normalisation factor $\frac{1}{2}$ ensures that the probability of all finite runs be one.[7]

Let us illustrate the previous definition on an example.

Example 5. Consider the following timed automaton:



We assume a uniform distribution over delays and enabled edges in every state. Then we can compute that $\mathbb{P}(\pi((\ell_0, 0), e_1 e_2)) = \frac{1}{64}\big(1 - 3\log\big(\frac{5}{4}\big)\big)$ as $\mu_{(\ell_0,0)} = \frac{\lambda}{2}$ (resp. $\mu_{(\ell_1,t)} = \frac{\lambda}{5-t}$) is the uniform distribution over $[0, 2]$ (resp. $[t, 5]$).

Lemma 6. *For every state s, $\mathbb{P}_{\mathcal{A}}$ is a probability measure over the set $\mathsf{Runs}(\mathcal{A}, s)$.*

We establish that probabilities in $\mathcal{A}$ and in $\mathsf{R}(\mathcal{A})$ are closely related, provided the measures we initially assign to $\mathcal{A}$ and $\mathsf{R}(\mathcal{A})$ are similar. Hence, if $\mu^{\mathcal{A}}$ (resp. $\mu^{\mathsf{R}(\mathcal{A})}$) is the measure in $\mathcal{A}$ (resp. $\mathsf{R}(\mathcal{A})$), we assume that for every state s in $\mathcal{A}$, $\mu^{\mathcal{A}}_s = \mu^{\mathsf{R}(\mathcal{A})}_{\iota(s)}$.[8] This is possible as one can easily be convinced that $I(s) = I(\iota(s))$. Similarly, if $p^{\mathcal{A}}$ (resp. $p^{\mathsf{R}(\mathcal{A})}$) is the distribution over edges in $\mathcal{A}$ (resp. $\mathsf{R}(\mathcal{A})$), we assume that for every state s in $\mathcal{A}$, for every $t \in \mathbb{R}_+$ $p^{\mathcal{A}}_{s+t} = p^{\mathsf{R}(\mathcal{A})}_{\iota(s)+t}$. Under those assumptions, we have the following result.

[7] Without this factor, for all n, the measure of runs of length n is one. This factor is not completely satisfactory as it has no 'physical' interpretation, but it is not a problem as we are only interested in qualitative properties.

[8] Note that we abuse notations and use $\iota(s)$ for $\iota(\pi(s))$.

Lemma 7. *Let $\mathcal{A}$ be a non-blocking timed automaton. Assume measures in $\mathcal{A}$ and in $\mathsf{R}(\mathcal{A})$ are related as described above. Let π be a symbolic path in $\mathcal{A}$. Then, $\iota(\pi)$[9] is a $\mathbb{P}_{\mathsf{R}(\mathcal{A})}$-measurable set of runs in $\mathsf{R}(\mathcal{A})$, and $\mathbb{P}_{\mathcal{A}}(\pi) = \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\iota(\pi))$.*

3.2 Probabilistic Semantics

We consider the logic LTL [14], defined inductively as:

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

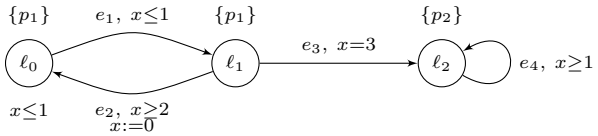
where $p \in \mathsf{AP}$ is an atomic proposition. We use classical shorthands like $\mathtt{tt} \stackrel{\text{def}}{=} p \vee \neg p$, $\mathtt{ff} \stackrel{\text{def}}{=} p \wedge \neg p$, $\varphi_1 \Rightarrow \varphi_2 \stackrel{\text{def}}{=} \neg\varphi_1 \vee \varphi_2$, $\mathbf{F}\,\varphi \stackrel{\text{def}}{=} \mathtt{tt}\,\mathbf{U}\,\varphi$, and $\mathbf{G}\,\varphi \stackrel{\text{def}}{=} \neg\mathbf{F}\,(\neg\varphi)$.

We interpret LTL formulas over finite runs of a timed automaton. Given a 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, it is correct to speak of the probability $\mathbb{P}_{\mathcal{A}}\{\varrho \in \mathsf{Runs}(\mathcal{A}, s_0) \mid \varrho \models \varphi\}$, which we simply write $\mathbb{P}_{\mathcal{A}}(s_0, \varphi)$.

Let φ be an LTL formula. We say that $\mathcal{A}$ *almost-surely satisfies* φ from s_0 w.r.t. $\mathbb{P}_A$, and we then write $\mathcal{A}, s_0 \models_{\mathbb{P}} \varphi$, if $\mathbb{P}_{\mathcal{A}}(s_0, \varphi) = 1$.

Remark 8. Our model of timed automata has no accepting locations. This is restrictive as some formulas will be trivially wrong (for instance, eventualities). However, we can deal with accepting locations as well. Let acc be a new atomic proposition and ψ be an LTL formula characterising the accepting runs, *i.e.*, $\psi \stackrel{\text{def}}{=} \mathbf{F}\,\mathbf{G}\,\mathsf{acc}$. Instead of considering $\mathbb{P}_{\mathcal{A}}(s_0, \varphi)$ we would rather evaluate the conditional probability $\mathbb{P}_{\mathcal{A}}(s_0, \varphi \mid \psi)$. Clearly enough, verifying that $\mathbb{P}_{\mathcal{A}}(s_0, \varphi \mid \psi) = 1$ in the automaton without accepting locations corresponds to checking $\mathbb{P}_{\mathcal{A}}(s_0, \varphi) = 1$ in the automaton where accepting locations are those labelled with acc. Note that this only makes sense if $\mathbb{P}_{\mathcal{A}}(s_0, \psi) \neq 0$, however timed automata such that $\mathbb{P}_{\mathcal{A}}(s_0, \psi) = 0$ can be considered as degenerated.

Example 9. Consider the timed automaton $\mathcal{A}$ depicted below:



If $s_0 = (\ell_0, 0)$ is the initial state, then $\mathcal{A}, s_0 \not\models \mathbf{G}\,p_1$ but $\mathcal{A}, s_0 \models_{\mathbb{P}} \mathbf{G}\,p_1$. Indeed, in this example, the transition e_3 will unlikely happen, because its guard $x = 3$ is much too 'small' compared with the guard $x \geq 2$ of the transition e_2.

Lemma 7 directly implies the following:

Corollary 10. *Let $\mathcal{A}$ be a non-blocking timed automaton, s a state of $\mathcal{A}$, and φ an LTL formula. Then,*

$$\mathcal{A}, s \models_{\mathbb{P}} \varphi \ \Leftrightarrow \ \mathsf{R}(\mathcal{A}), \iota(s) \models_{\mathbb{P}} \varphi\,.$$

[9] Recall that, if ϱ is a run in $\mathcal{A}$, then $\iota(\varrho)$ is the image of ϱ in $\mathsf{R}(\mathcal{A})$ (see page 4).

4 A Topological Semantics for Timed Automata

In this section, we propose a *large* semantics for LTL over timed automata. This large semantics, based on a natural topology on timed automata, asserts that an LTL formula is *largely satisfied* if 'most of the runs' satisfy it. We use classical topological tools (including the dimension) to characterise what we mean by 'most of the runs'.

4.1 Some Topological Notions

We do not recall classical definitions in topology but refer to [12]. However, some notions are less common, we thus recall them here. The density notion is not appropriate to express a 'most of the runs' notion, because rather small sets are dense, *e.g.* the set $\mathbb{Q}$ in $\mathbb{R}$. As already pointed out in [17] the notion of *largeness*, and its complement the *meagerness* are more appropriate. Let $(A, \mathcal{T})$ be a topological space. If $B \subseteq A$, we denote by $\mathring{B}$ (resp. $\overline{B}$) the interior (resp. closure) of B. A set $B \subseteq A$ is nowhere dense if $\mathring{\overline{B}} = \emptyset$. A set is *meager* if it is a countable union of nowhere dense sets. Finally, a set is *large* if its complement is meager.

Although the notion of largeness is quite abstract, it admits a very nice characterisation in terms of a two-player game, known as *Banach-Mazur game*. 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 a subset of 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 $\bigcap_{i=1}^{\infty} B_i \cap C \neq \emptyset$, else Player 2 wins.

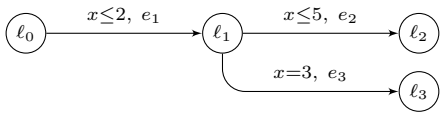
Banach-Mazur games are not always determined, even for simple topological spaces (see [13, Remark 1]). Still a natural question is to know when the players have winning strategies. The following result gives a partial answer:

Theorem 11 (Banach-Mazur [13]). *Player 2 has a winning strategy in the Banach-Mazur game with target set C if and only if C is meager.*

4.2 The Dimension of a Symbolic Path

In $\mathbb{R}^n$, open sets are among those sets of maximal dimension. Here, we are not exactly in $\mathbb{R}^n$, but each symbolic constrained path can be embedded in some $\mathbb{R}^m$. A notion of *dimension of a symbolic path* then naturally arises. Before going to the details, let us explain through an example the intuition behind this notion.

Example 12. Let $\mathcal{A}$ be the timed automaton depicted below, let s_0 be the state $(\ell_0, 0)$ and π be the (unconstrained) symbolic path $\pi(s_0, e_1e_2)$.



One can naturally associate a polyhedron of $(\mathbb{R}_+)^2$ with π:

$$\begin{aligned}\mathsf{Pol}(\pi) &= \{(\tau_1,\tau_2) \in (\mathbb{R}_+)^2 \mid \varrho = s_0 \xrightarrow{\tau_1,e_1} s_1 \xrightarrow{\tau_2,e_2} s_2 \in \mathsf{Runs}(\mathcal{A},s_0)\}\\ &= \{(\tau_1,\tau_2) \in (\mathbb{R}_+)^2 \mid (0 \le \tau_1 \le 2) \wedge (0 \le \tau_1+\tau_2 \le 5)\}\end{aligned}$$

$\mathsf{Pol}(\pi)$ has dimension 2 in $\mathbb{R}^2$. Since it is of maximal dimension, we say the dimension of the symbolic path π is *defined*. Consider now the symbolic path $\pi' = \pi(s_0, e_1e_3)$. The polyhedron $\mathsf{Pol}(\pi')$ associated with π' has dimension 1, and is embedded in a two-dimensional space. In that case, we say that its dimension is *undefined*.

In general, we need to be careful with singular transitions, *i.e.*, transitions which do not increase the dimension but play an important role (in the previous example, it would be the case if the edge e_1 was labelled with the guard $x = 2$; though this guard is very small, the role of edge e_1 is essential in the behaviour of the automaton).

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:

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

Definition 13. *Let $\mathcal{A}$ be a timed automaton, and $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1 \dots e_n)$ a constrained path. For each $0 \le i \le n$, we write $\mathcal{C}_i$ for the projection of $\mathsf{Pol}(\pi_{\mathcal{C}})$ over the variables of the i first coordinates, with the convention that $\mathcal{C}_0$ is true. We say that the dimension of $\pi_{\mathcal{C}}$ is* undefined*, and we then write $\dim_{\mathcal{A}}(\pi_{\mathcal{C}}) = \bot$, whenever there exists some index $1 \le i \le n$ such that*

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

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

4.3 Definition of a Topology over Finite Paths

For $\mathcal{A}$ a timed automaton, and s a state of $\mathcal{A}$, we define a *basic open set* as a constrained symbolic path $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1 \dots e_n)$ such that $\dim_{\mathcal{A}}(\pi_{\mathcal{C}})$ is defined, and $\mathsf{Pol}(\pi_{\mathcal{C}})$ is open in $\mathsf{Pol}(\pi)$ for the topology of $\mathbb{R}^n$ induced on $\mathsf{Pol}(\pi)$, where π stands for the (unconstrained) path $\pi(s, e_1 \dots e_n)$.

We write $\mathcal{T}_{\mathcal{A}}$ for the topology over $\mathsf{Runs}(\mathcal{A}, s)$ induced by these basic open sets and $\mathsf{Runs}(\mathcal{A}, s)$. Note that the basic open sets $\pi_{\mathcal{C}}$ together with $\mathsf{Runs}(\mathcal{A}, s)$ form a base for $\mathcal{T}_{\mathcal{A}}$.

Example 14. Let $\mathcal{A}$ be the timed automaton of Example 9 and $s_0 = (\ell_0, 0)$ be its initial state. The basic (unconstrained) open sets of $\mathsf{Runs}(\mathcal{A}, s_0)$ are sets of the form $\pi(s_0, (e_1e_2)^*)$ or of the form $\pi(s_0, e_1(e_2e_1)^*)$. A (constrained) basic open set is then for instance $\pi_{\mathcal{C}}(s_0, e_1e_2)$ with $\mathcal{C} = \{\frac{1}{3} < t_1 < \frac{1}{2}; t_1 + t_2 > 5\}$. One can be convinced that the set of paths of the form $\pi(s_0, (e_1e_2)^*e_3e_4^*)$ is meager.

Proposition 15. *Let $\mathcal{A}$ be a timed automaton, and s a state of $\mathcal{A}$. The topological space $(\mathsf{Runs}(\mathcal{A}, s), \mathcal{T}_{\mathcal{A}})$ is a Baire space.*[10]

Proof (Sketch). Let $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1 \dots e_n)$ be a non-empty basic open set. We then use Banach-Mazur games and Theorem 11 to prove that $\pi_{\mathcal{C}}$ is not meager: we prove that Player 2 has no winning strategy for the game playing with basic open sets and with $\pi_{\mathcal{C}}$ as an objective, by exhibiting a counter-strategy for Player 1.

Player 1 proceeds as follows: for the first round, she picks $\pi_1 = \pi_{\mathcal{C}}$. For the second round, Player 2 picks some $\pi_2 \subseteq \pi_1$. For the third round, Player 1 must be careful and cannot take an arbitrary open path included in π_2, because Player 1 could manage to choose the constraints so that the limit of the intersections be empty (by analogy in $\mathbb{R}$, the limit of $(0, \frac{1}{2^i})$ is the empty set). To avoid this, Player 1 can first consider a 'big' compact set F_2 within π_2 ('big' here means with a non-empty interior) — note that this is possible as the topology we consider, restricted to $\pi(s, e_1 \dots e_n)$, can be embedded in some $\mathbb{R}^m$ through the application $\mathsf{Pol}(\cdot)$. Then, she can play with a basic open set π_3 included in F_2. The game continues like this, and Player 1 only needs to use the above-described trick at each of her rounds. The intersection of all paths that have been played then corresponds to the intersection of a chain of compact sets, hence it is non-empty, by Heine-Borel theorem. □

We can now define a topological semantics for LTL based on the notion of largeness. Let φ be an LTL formula. We say that $\mathcal{A}$ *largely satisfies* φ from s, and we write $\mathcal{A}, s \models\!\!\approx_{\mathcal{T}} \varphi$, if the set $\{\varrho \in \mathsf{Runs}(\mathcal{A}, s) \mid \varrho \models \varphi\}$ is topologically large. The topologies in $\mathcal{A}$ and in $\mathsf{R}(\mathcal{A})$ are equivalent in the following sense.

Lemma 16. *Let $\iota : \mathsf{Runs}(\mathcal{A}, s) \to \mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s))$ be the projection of finite runs ϱ in $\mathcal{A}$ onto the region automaton (see page 4). Then ι is continuous, and for every non-empty open set $\mathcal{O} \in \mathcal{T}_{\mathcal{A}}$, $\iota(\mathcal{O}) \neq \emptyset$.*

Corollary 17. *Let $\mathcal{A}$ be a timed automaton, s a state of $\mathcal{A}$, and φ an* LTL *formula. Then,*

$$\mathcal{A}, s \models\!\!\approx_{\mathcal{T}} \varphi \;\Leftrightarrow\; \mathsf{R}(\mathcal{A}), \iota(s) \models\!\!\approx_{\mathcal{T}} \varphi\,.$$

5 Correspondence of the Two Semantics

In this section we prove our main theorem: probabilistic and topological semantics coincide! We first relate dimension and probabilities in the region automaton.

Proposition 18. *Let $\mathcal{A}$ be a non-blocking timed automaton, and π be an unconstrained symbolic path in $\mathsf{R}(\mathcal{A})$. Then, $\mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi) > 0$ iff $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \top$.*[11]

[10] In modern definitions, a topological space is a Baire space if each countable union of closed sets with an empty interior has an empty interior. However, originally, a topological space is a Baire space whenever every non-empty open set is not meager. The two definitions coincide, see [12, p.295].

[11] This is in particular independent of the choice of the probability distributions over delays.

The main result of this paper is the following theorem.

Theorem 19. *Let $\mathcal{A}$ be a non-blocking timed automaton, s a state of $\mathcal{A}$, and φ an* LTL *formula. Then,*

$$\mathcal{A}, s \models_{\mathbb{P}} \varphi \;\Leftrightarrow\; \mathcal{A}, s \models_{\mathcal{T}} \varphi\,.$$

Proof (Sketch). Thanks to Corollaries 10 and 17, it is equivalent to prove that $\mathsf{R}(\mathcal{A}), \iota(s) \models_{\mathcal{T}} \varphi$ iff $\mathsf{R}(\mathcal{A}), \iota(s) \models_{\mathbb{P}} \varphi$. Moreover, $\mathsf{R}(\mathcal{A}), \iota(s) \models_{\mathbb{P}} \varphi$ iff $\mathbb{P}_{\mathcal{A}}(\iota(s), \neg\varphi) = 0$, thus applying Proposition 18, $\mathsf{R}(\mathcal{A}), \iota(s) \models_{\mathbb{P}} \varphi$ iff every symbolic path π in $\mathsf{R}(\mathcal{A})$ starting in $\iota(s)$ and satisfying $\neg\varphi$ has an undefined dimension. We finally prove that this last property is equivalent to $\mathsf{R}(\mathcal{A}), \iota(s) \models_{\mathcal{T}} \varphi$, *i.e.*, to the fact that $[\![\neg\varphi]\!] = \{\varrho \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s)) \mid \varrho \not\models \varphi\}$ is topologically meager.

For the first implication, we use Banach-Mazur games and Theorem 11 to prove that Player 2 has a winning strategy for the objective $[\![\neg\varphi]\!]$ (still playing with the basic open sets of $\mathcal{T}_{\mathsf{R}(\mathcal{A})}$). Let π_1 be the path chosen by Player 1 at the first round. This path has necesseraly defined dimension and thus, by hypothesis and Proposition 18, it satisfies φ. Whatever is played afterwards, the intersection with the objective will be empty. Hence Player 2 wins and $[\![\neg\varphi]\!]$ is meager.

For the second implication, assume that $[\![\neg\varphi]\!]$ is meager. As the topological space $(\mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s)), \mathcal{T}_{\mathsf{R}(\mathcal{A})})$ is a Baire space (see Proposition 15), $[\![\neg\varphi]\!]^{\circ} = \emptyset$. Hence, there is no path in $\mathsf{R}(\mathcal{A})$ from $\iota(s)$ with defined dimension which does not satisfy φ. □

Remark 20. To handle accepting states in the previous theorem, it would be sufficient to quantify only over paths in $\mathsf{R}(\mathcal{A})$ which are accepting.

6 Decidability Issues

Theorem 21. *Over finite timed words, the almost-sure and the large* LTL *model-checking problems over non-blocking timed automata are* PSPACE*-Complete.*

Proof (Sketch). The two problems are equivalent, due to Theorem 19. The PSPACE-Hardness follows from the PSPACE-Hardness of LTL model checking over finite automata. To describe a PSPACE algorithm, we first color each edge of $\mathsf{R}(\mathcal{A})$ as follows: if e is an edge in $\mathsf{R}(\mathcal{A})$, we color it in red whenever $\mu_s(I(s,e)) = 0$ for some $s \in q$ (note that this property is independent of the choice of $s \in q$, and that it is equivalent to $\dim(I(s,e)) < \dim(I(s))$ thanks to the property of the measure μ_s, see page 4), and we color it in blue otherwise.

Lemma 22. *Let $\mathcal{A}$ be a timed automaton and $\pi = \pi(s, e_1 \dots e_n)$ a symbolic path in $\mathsf{R}(\mathcal{A})$. Then, $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \bot$ iff at least one of the edges of π is red.*

Now, applying Proposition 18, to decide whether $\mathcal{A}, s \not\models_{\mathbb{P}} \varphi$, it is sufficient to guess a path in $\mathsf{R}(\mathcal{A})$ which has defined dimension (*i.e.*, does not contain any red edge), and does not satisfy φ. There is such a path with length at most exponential, it can thus be done in NPSPACE =PSPACE. □

7 Related Work

In this section we briefly compare our two semantics with existing works. A deeper related work section can be found in our research report [5].

The model of *real-time probabilistic processes* introduced in [1, 2] seems similar to timed automata interpreted with our probabilistic semantics, but it is indeed not the case. First, such a system is composed of a number of independent processes with a single clock, which implies in particular that clocks are completely independent. Then, and this is even more important, the choice of the transition to be taken is made before choosing probabilistically a delay. As a consequence, even transitions with small firing intervals can have a high probability to be taken, even though events with much larger firing intervals are possible. This is why this model satisfies stronger properties than ours.

We now compare our topology with the one introduced in [9] and further studied in [11]. First notice that their topology is defined on finite timed words and we define our topology on the set of finite runs. In particular, as already mentioned in the introduction, their topology only depends on the language and not on the automaton, while ours does. This implies that the topologies are 'incomparable', more precisely we can find sets that are open for our topology and not for their topology, and *vice-versa*.

8 Conclusion

In this paper, we have proposed two satisfaction relations for LTL formulas over timed automata which rule out unlikely (sequences of) events. The first one is based on a probabilistic semantics of timed automata, and to the best of our knowledge, is the first attempt to provide a probabilistic interpretation for non probabilistic timed systems in order to establish linear-time properties assuming 'fairness' on actions and delays. It naturally raises (qualitative) model-checking questions, for instance whether the probability that an LTL property holds is 1 (almost-sure model-checking problem). The second one is based on the topological concept of largeness, and yields a natural large semantics for LTL. We prove that these two interpretations for LTL coincide. Moreover, we establish that LTL model checking under those non-standard semantics is not harder than ordinary LTL model-checking (PSPACE-Complete).

The method we have developed here could straightforwardly extend in various directions. All untimed properties over finite runs, whose truth is invariant by regions, can be treated that way (for instance properties expressed in the logic CTL* or in the μ-Calculus). It could also be applied to various classes of hybrid systems with a finite bisimulation quotient [10].

We are currently extending this work to the framework of infinite timed words which raises even more complex problems, and we plan to extend it further in several directions, like for properties expressed in a timed logic, or to the quantitative analysis of this model (for instance, computing the exact, or approximate, probability of satisfying a given property, *etc*), or to control problems, *etc.*

References

1. R. Alur, C. Courcoubetis, and D. Dill. Model-checking for probabilistic real-time systems. In *Proc. 18th Int. Coll. Automata, Languages and Programming (ICALP'91)*, volume 510 of *LNCS*, pages 115–126. Springer, 1991.
2. R. Alur, C. Courcoubetis, and D. Dill. Verifying automata specifications of probabilistic real-time systems. In *Proc. Work. Real-Time: Theory in Practice, 1991*, volume 600 of *LNCS*, pages 28–44. Springer, 1992.
3. R. Alur and D. Dill. A theory of timed automata. *Theoretical Comp. Sci.*, 126(2):183–235, 1994.
4. R. Alur, S. La Torre, and P. Madhusudan. Perturbed timed automata. In *Proc. 8th Int. Work. Hybrid Systems: Computation and Control (HSCC'05)*, volume 3414 of *LNCS*, pages 70–85. Springer, 2005.
5. C. Baier, N. Bertrand, P. Bouyer, Th. Brihaye, and M. Größer. Probabilistic and topological semantics for timed automata. Research Report LSV–07–26, LSV, ENS de Cachan, France, 2007.
6. P. Bouyer, N. Markey, and P.-A. Reynier. Robust model-checking of timed automata. In *Proc. 7th Latin American Symp. Theoretical Informatics (LATIN'06)*, volume 3887 of *LNCS*, pages 238–249. Springer, 2006.
7. M. De Wulf, L. Doyen, N. Markey, and J.-F. Raskin. Robustness and implementability of timed automata. In *Proc. Joint Conf. Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant System (FORMATS+FTRTFT'04)*, volume 3253 of *LNCS*, pages 118–133. Springer, 2004.
8. M. De Wulf, L. Doyen, and J.-F. Raskin. Almost ASAP semantics: From timed models to timed implementations. In *Proc. 7th Int. Work. Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *LNCS*, pages 296–310. Springer, 2004.
9. V. Gupta, Th. A. Henzinger, and R. Jagadeesan. Robust timed automata. In *Proc. Int. Work. Hybrid and Real-Time Systems (HART'97)*, volume 1201 of *LNCS*, pages 331–345. Springer, 1997.
10. Th. A. Henzinger, R. Majumdar, and J.-F. Raskin. A classification of symbolic transition systems. *ACM Transactions on Computational Logic*, 6(1):1–32, 2005.
11. Th. A. Henzinger and J.-F. Raskin. Robust undecidability of timed and hybrid systems. In *Proc. 3rd Int. Work. Hybrid Systems: Computation and Control (HSCC'00)*, volume 1790 of *LNCS*, pages 145–159. Springer, 2000.
12. J. R. Munkres. *Topology.* Prentice Hall, 2nd edition, 2000.
13. J. C. Oxtoby. The Banach-Mazur game and Banach category theorem. *Annals of Mathematical Studies*, 39:159–163, 1957.
14. A. Pnueli. The temporal logic of programs. In *Proc. 18th Ann. Symp. Foundations of Computer Science (FOCS'77)*, pages 46–57. IEEE Comp. Soc. Press, 1977.
15. A. Puri. Dynamical properties of timed automata. In *Proc. 5th Int. Symp. Formal techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98)*, volume 1486 of *LNCS*, pages 210–227. Springer, 1998.
16. J. Sproston. Model checking for probabilistic timed systems. In *Validation of Stochastic Systems – A Guide to Current Research*, volume 2925 of *LNCS*, pages 189–229. Springer, 2004.
17. D. Varacca and H. Völzer. Temporal logics and model checking for fairly correct systems. In *Proc. 21st Ann. Symp. Logic in Computer Science (LICS'06)*, pages 389–398. IEEE Comp. Soc. Press, 2006.