Christel Baier, Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, Marcus Größer

A Probabilistic Semantics for Timed Automata

Research Report LSV-07-13

Laboratoire Spécification et Vérification

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

A Probabilistic 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 ideal mathematical model to represent systems with strong timing requirements. In such mathematical models, properties can be violated, due to unlikely events. Following ideas of Varacca and Völzer [VV06], we aim at defining a notion of "fair" correctness for timed systems. For this purpose, we introduce a probabilistic semantics for timed automata, which ignores unlikely events, and naturally raises a notion of almost-sure satisfaction of properties in timed systems. We prove that the almost-sure satisfaction has a corresponding topological interpretation in terms of largeness of the set of paths satisfying the property. Moreover, we show PSPACE-Completeness for the LTL model-checking problem over finite computations.

1 Introduction

Timed automata, a model for verification. In the 90's, Alur and Dill proposed timed automata [AD94] as a model for verification purposes, which takes into account real-time constraints. Timed automata are finite-state automata enriched with clock variables that increase synchronously with time, can be compared to fixed integral values and reset to zero. With such a model, one can very easily 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 (like Uppaal [LPY97] or Kronos [DOTY96]).

Idealization of mathematical models. Like most models used in model checking, timed automata are an ideal mathematical model, in which several assumptions are implicitely made. For instance, such a model is supposed to have infinite precision, instantaneous events, *etc.* Of course this can only be an idealization of a real behaviour of a system, as these hypotheses are in practice unrealistic. In the literature, several ideas for attacking this problem have been explored. For instance the model of implementable controllers in [DDR04] was proposed, where constraints and precision of clocks are somewhat relaxed. In this framework, if the model (with the relaxed, or robust, semantics) satisfies a safety property, then, on a simple model of processor,

⋆ Partly supported by a Lavoisier fellowship (program of the French Ministry of Foreign Affairs).
⋆⋆ Partly supported by a Marie Curie fellowship (program of the European Commission).

its implementation will also satisfy this property. This implementation model with relaxed constraints has been then considered in [Pur98,DDMR04,ALM05,BMR06]. 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).

Another robustness model has been proposed at the end of the 90's in [GHJ97] with the notion of tube acceptance: a metric is put on the set of traces of the timed automaton, and a trace of the automaton is robustly accepted by the automaton if and only if a tube around that trace is classically accepted. That way, "lone" traces are somehow removed from the system behaviours. This acceptance has been further studied for language-based properties, for instance the universality problem [HR00]. 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 [VV06], Varacca and Völzer propose a probabilistic framework for finite-state systems to overcome a side-effect of modelling. For instance (following [VV06]), in the Dijkstra's dining philosopher algorithm [Dij71], a philosopher may starve just because his two neighbours conspire and use their fork in alternation so that this precise philosopher never has the two forks available. However, this has very few chance to happen, as it requires special conditions. Still, in a formal way, the algorithm does not satisfy the starvation-free property (stating that all philosophers eventually eat), but it will satisfy this property, except along very few traces. The use of probability helps quantifying this notion of "very few", and defining 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 [VV06]) and ruling out unlikely events (from a time point of view, as in the implementability paradigm discussed above), and use probability as a tool to ignore "almost impossible" behaviours. We hence propose a *probabilistic semantics* for timed automata which assigns probabilities both on delays and on discrete choices. This then provides a purely probabilistic dense-time operational interpretation, on which there is a natural *almost-sure* model-checking definition: a property is almost-surely satisfied if the probability that this property holds is equal to one, or equivalently if there are only "very few" behaviours not satisfying that property ("very few" being obviously interpreted as "with probability zero"). We then reinforce this definition by providing a topological characterization using the notion of largeness already argued in [VV06]. Finally, we prove the decidability

of the almost-sure model-checking problem for specifications expressed in LTL, and prove that it 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 briefly review the different probability+time models that have been considered in the literature (we refer to [Spr04] for more details). However, they 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. The model of probabilistic timed automata [KNSS02] implemented in the tool PRISM [KNP04] has been used for several industrial applications, but probabilities are discrete and time-passage is treated non-deterministically. Continuous-time Markov chains [BHHK03] put random delays on the edges, but have no structural restrictions (like clock constraints), and have only one implicit clock which is reset in each step. Real-time probabilistic systems [ACD91,ACD92] (later applied for quantifying test cases [JPQ05]) are sets of one-clock independent timed automata that schedule tasks probabilistically and can be seen as finite Markov chains. Finally, our measure defined for timed automata yields a stochastic model that is close to labelled Markov processes [DGJP03,DGJP04] defined on a continuum.

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* [AD94].

Timed automata. We denote by $X = \{x_1, \ldots, x_k\}$ a finite set of *clocks*. A *clock valuation* over X is a mapping $\nu : X \to \mathbb{R}_+$, where $\mathbb{R}_+$ denotes the set of nonnegative reals. We write $\mathbb{R}_+^X$ for the set of clock valuations over X. Given a clock valuation ν and $\tau \in \mathbb{R}_+$, $\nu + \tau$ is the clock valuation defined by $(\nu + \tau)(x) = \nu(x) + \tau$ for every $x \in X$. If $Y \subseteq X$, the valuation $[Y \leftarrow 0]\nu$ is the valuation ν' such that $\nu'(x) = 0$ if $x \in Y$, and $\nu'(x) = \nu(x)$ otherwise. A *guard* over X is a finite conjunction of expressions of the form $x \sim c$ where $x \in X$ is a clock, $c \in \mathbb{N}$ is an integer, and $\sim$ is one of the symbols $\{<, \leq, =, \geq, >\}$. 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 the clock valuation ν satisfies the guard g. We denote AP a finite set of atomic propositions.

Definition 1. *A* timed automaton *is a tuple* $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I}, \mathcal{L})$ *such that:* (i) *L is a finite set of locations,* (ii) *X is a finite set of clocks,* (iii) *Σ is a finite set of actions,* (iv) *$E \subseteq L \times \Sigma \times \mathcal{G}(X) \times 2^X \times L$ is a finite set of edges,* (v) *$\mathcal{I} : L \to \mathcal{G}(X)$ assigns an invariant to each location, and* (vi) *$\mathcal{L} : L \to 2^{\mathsf{AP}}$ is the labelling function.*

The semantics of a timed automaton $\mathcal{A}$ is given by a labelled transition system $T_{\mathcal{A}} = (S_{\mathcal{A}}, E \cup \mathbb{R}_+, \rightarrow_{\mathcal{A}})$[4] where the set $S_{\mathcal{A}}$ of states is $\{s = (\ell, \nu) \in L \times \mathbb{R}_+^X \mid \nu \models \mathcal{I}(\ell)\}$, and the transition relation $\rightarrow_{\mathcal{A}}$ ($\subseteq S_{\mathcal{A}} \times (E \cup \mathbb{R}_+) \times S_{\mathcal{A}}$) is composed of delay and discrete transitions:

- *(delay transition)* $(\ell, \nu) \xrightarrow{\tau}_{\mathcal{A}} (\ell, \nu + \tau)$ if $\tau \in \mathbb{R}_+$ and if for all $0 \leq \tau' \leq \tau$, $\nu + \tau' \models \mathcal{I}(\ell)$,
- *(discrete transition)* $(\ell, \nu) \xrightarrow{e}_{\mathcal{A}} (\ell', \nu')$ if $e = (\ell, a, g, Y, \ell') \in E$ 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.*, $\varrho = s_0 \xrightarrow{\tau_1} s'_1 \xrightarrow{e_1} s_1 \xrightarrow{\tau_2} s'_2 \xrightarrow{e_2} s_2 \dots s_{n-1} \xrightarrow{\tau_n} s'_n \xrightarrow{e_n} s_n$ or more compactly $s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \dots s_{n-1} \xrightarrow{\tau_n, e_n} s_n$. We write $\mathsf{Runs}(\mathcal{A}, s_0)$ for the set of finite runs of $\mathcal{A}$ from state s_0.

In the sequel, given $s \in S_{\mathcal{A}}$ and e an edge, we denote by $I(s, e) = \{\tau \in \mathbb{R}_+ \mid s \xrightarrow{\tau, e} s'\}$ and $I(s) = \bigcup_e I(s, e)$. Note that $I(s, e)$ is an interval, whereas $I(s)$ is a finite union of intervals. We also define $S(s, e) = \{s' \mid \exists \tau \in I(s, e) \text{ s.t. } s \xrightarrow{\tau, e} s'\}$. The timed automaton $\mathcal{A}$ is said *non-blocking* whenever for every state $s \in S_{\mathcal{A}}$, $I(s) \neq \emptyset$.

If s is a state of $\mathcal{A}$ and $(e_i)_{1 \leq i \leq n}$ is a finite sequence of edges of $\mathcal{A}$, the *(symbolic) path* starting from s and determined by $(e_i)_{1 \leq i \leq n}$ is the following set of runs:

$$\pi(s, e_1, \dots, e_n) = \{\varrho = s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \;\mid\; \varrho \in \mathsf{Runs}(\mathcal{A}, s)\}\,.$$

The region automaton abstraction. The well-known region automaton construction is an abstraction of timed automata which can be used for verifying many properties, for instance regular untimed properties.

Let $\mathcal{A}$ be a timed automaton. Define M the largest constant to which clocks are compared in guards or invariants of $\mathcal{A}$. Two clock valuations ν and ν' are said *region-equivalent* (written $\nu \approx \nu'$) whenever the following conditions hold:

- $\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$ or $\nu(x), \nu'(x) > M$, for all $x \in X$;
- $\{\nu(x)\} = 0$ iff $\{\nu'(x)\} = 0$, for all $x \in X$ with $\nu(x) \leq M$;
- $\{\nu(x)\} \leq \{\nu(y)\}$ iff $\{\nu'(x)\} \leq \{\nu'(y)\}$, for all $x, y \in X$ with $\nu(x), \nu(y) \leq M$.

where, $\lfloor \cdot \rfloor$ denotes the integral part, and $\{\cdot\}$ denotes the fractional part.

This equivalence relation on clock valuations has a finite (exponential) index, and extends to the states of $\mathcal{A}$, saying that $(\ell, \nu) \approx (\ell', \nu')$ iff $\ell = \ell'$ and $\nu \approx \nu'$. We use $[\nu]$ (resp. $[(\ell, \nu)]$) to denote the equivalence class to which ν (resp. (ℓ, ν)) belongs. A *region* is an equivalence class of valuations. The set of all the regions is denoted by R. If r is a region, we denote by $\mathsf{cell}(r)$ the smallest guard defined with constants smaller than M, and which contains r.

The original region automaton [AD94] is a finite automaton which is the quotient of the time abstract transition $T_{\mathcal{A}}$ by the equivalence relation $\approx$. Here, we use a slight

[4] When the underlying timed automaton is clear from the context, we write only $\rightarrow$ instead of $\rightarrow_{\mathcal{A}}$.

modification of the original construction, which is still a timed automaton, but which satisfies very strong properties.

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

- $Q = L \times R$; $\quad$ $\kappa((\ell, r)) = \mathcal{I}(\ell)$, *and* $\lambda((\ell, r)) = \mathcal{L}(\ell)$ *for every* $(\ell, r) \in L \times R$;
- $T \subseteq (Q \times \mathsf{cell}(R) \times \Sigma \times 2^X \times Q)$, *and* $(\ell, r) \xrightarrow{\mathsf{cell}(r''), a, Y} (\ell', r')$ *is in* T *iff there exists* $e = \ell \xrightarrow{g, a, Y} \ell'$ *in* E *such that there exists* $\nu \in r$, $\tau \in \mathbb{R}_+$ *with* $(\ell, \nu) \xrightarrow{\tau, e} (\ell', \nu')$, $\nu + \tau \in r''$ *and* $\nu' \in r'$.

We recover the usual region automaton of [AD94] by labelling the transitions "a" instead of "$\mathsf{cell}(r''), a, Y$", and by interpreting $R(\mathcal{A})$ as a finite automaton. However, the above timed interpretation satisfies strong timed bisimulation properties that we do not detail here (we assume the reader is familiar with this construction). To every finite path $\pi((\ell, \nu), e_1, \ldots, e_n)$ in $\mathcal{A}$ corresponds a finite set of paths in $\pi(((\ell, [\nu]), \nu), f_1, \ldots, f_n)$ in $R(\mathcal{A})$, each one corresponding to a choice in the regions that are crossed. If ϱ is a run in $\mathcal{A}$, then we write $\iota(\varrho)$ its (unique) image in $R(\mathcal{A})$. Finally, note that if $\mathcal{A}$ is non-blocking, then so is $R(\mathcal{A})$.

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). 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 [Spr04] 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:

- $\mu_s(I(s)) = \mu_s(\mathbb{R}_+) = 1$;[5]
- If λ is the Lebesgue measure, if $\lambda(I(s)) > 0$, μ_s is equivalent[6] to λ on $I(s)$; Else, μ_s is equivalent on $I(s)$ to the uniform distribution over points of $I(s)$;
- μ_s admits a density function d_s.

Remark 3. The above constraints on probability measures are rather loose and are for instance satisfied by:

[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, $\nu(A) = 0 \Leftrightarrow \nu'(A) = 0$.

– the uniform discrete distribution over $I(s)$ if $I(s)$ is a finite set of points;
– the Lebesgue measure over $I(s)$, normalized to have a probability measure, if $I(s)$ is a finite set of bounded intervals;
– an exponential distribution if $I(s)$ contains an unbounded interval.

3.1 Definition of a Probability Measure over Finite Paths

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

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

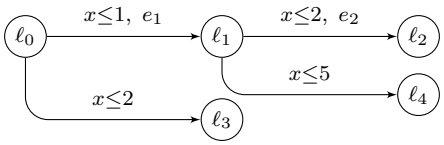
where $s \xrightarrow{t,e_1} s_t$. We initialize with $\mathbb{P}_{\mathcal{A}}(\pi(s)) = \frac{1}{2}$.

Using Fubini's theorem and 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 $\mathcal{A}$.

Intuitively, this formula can be read as follows: the probability of waiting t time units is $\mathrm{d}\mu_s(t)$. Then there is a nondeterministic choice between all transitions enabled at that time, that is why we divide by $\#\{I(s,e) \mid t \in I(s,e)\}$. Hence, the probability of taking transition e_1 at date t is $\frac{1}{\#\{I(s,e) | t \in I(s,e)\}} \mathrm{d}\mu_s(t)$, and the probability that the run starts by such a move is then $\frac{\mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))}{\#\{I(s,e) | t \in I(s,e)\}} \mathrm{d}\mu_s(t)$. We finally just need to sum up over all t's in $I(s, e_1)$. Factor $\frac{1}{2}$ is a normalization factor, that will enforce the probability of all finite runs to be exactly 1 (indeed, the set of paths of fixed length n has probability $\frac{1}{2^{n+1}}$). This can be changed, if we better want the probability of finite runs of length n (for every integer n) to be 1.

Let us illustrate the previous definition on an example. More details on the computation and another example are reported in the appendix.

Example 5. Consider the following timed automaton:



For simplicity, we assume uniform distribution over delays in every state:

$$\begin{aligned}
\mathbb{P}(\pi((\ell_0, 0), e_1, e_2)) &= \frac{1}{2} \int_{t \leq 1} \frac{1}{4} \Big(\int_{t \leq u \leq 2} \frac{1}{4} \mathrm{d}\mu_{(\ell_1, t)}(u) \Big) \mathrm{d}\mu_{(\ell_0, 0)}(t) \\
&= \frac{1}{32} \int_{t \leq 1} \Big(\int_{t \leq u \leq 2} \frac{1}{5-t} \mathrm{d}\lambda(u) \Big) \frac{1}{2} \mathrm{d}\lambda(t) \\
&= \frac{1}{64} \Big(1 - 3 \log \Big(\frac{5}{4} \Big) \Big)
\end{aligned}$$

since $\mu_{(\ell_0,0)} = \frac{\lambda}{2}$ is the uniform distribution on $[0,2]$ and $\mu_{(\ell_1,t)} = \frac{\lambda}{5-t}$ is the uniform distribution on $[t,5]$. □

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

Proof. We prove by induction on n that the probability of the set of paths of length n is $\frac{1}{2^{n+1}}$. As we assume the automaton is non-blocking, there are paths of every length, hence the probability of all paths of finite length is 1. Moreover, for every s in $\mathcal{A}$, $\mathbb{P}_{\mathcal{A}}(\pi(s)) = \frac{1}{2}$. Hence case $n = 0$ holds. Assume $n \geq 1$, and let $\mathsf{Runs}^n(\mathcal{A}, s)$ be the set of runs in $\mathcal{A}$ of length n starting in s. We note $\mathbb{1}_I$ the characteristic function of set $I \subseteq \mathbb{R}_+$. Then,

$$
\begin{aligned}
\mathbb{P}_{\mathcal{A}}(\mathsf{Runs}^n(\mathcal{A}), s) &= \sum_{e_1,\ldots,e_n} \mathbb{P}_{\mathcal{A}}(\pi(s, e_1, \ldots, e_n)) \\
&= \sum_{e_1,\ldots,e_n} \frac{1}{2} \int_{t \in I(s,e_1)} \frac{\mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))}{\#\{I(s,e) \mid t \in I(s,e)\}} \mathrm{d}\mu_s(t) \\
&= \frac{1}{2} \sum_{e_1} \int_{t \in I(s,e_1)} \frac{\sum_{e_2,\ldots,e_n} \mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))}{\#\{I(s,e) \mid t \in I(s,e)\}} \mathrm{d}\mu_s(t) \\
&= \frac{1}{2} \sum_{e_1} \int_{t \in I(s,e_1)} \frac{\mathbb{P}_{\mathcal{A}}(\mathsf{Runs}^{n-1}(\mathcal{A}, s_t))}{\#\{I(s,e) \mid t \in I(s,e)\}} \mathrm{d}\mu_s(t) \\
&= \frac{1}{2} \sum_{e_1} \int_{t \in I(s,e_1)} \frac{\frac{1}{2^n}}{\#\{I(s,e) \mid t \in I(s,e)\}} \mathrm{d}\mu_s(t) \qquad \text{by i.h.} \\
&= \frac{1}{2^{n+1}} \int_{t \in I(s)} \frac{1}{\#\{I(s,e) \mid t \in I(s,e)\}} \left(\sum_{e_1} \mathbb{1}_{I(s,e_1)}(t) \right) \mathrm{d}\mu_s(t) \\
&= \frac{1}{2^{n+1}} \int_{t \in I(s)} \mathrm{d}\mu_s(t) \\
&= \frac{1}{2^{n+1}} \qquad \text{since } \mu_s(I(s)) = 1 \text{ (see page 5).}
\end{aligned}
$$

This concludes the proof. □

Lemma 7. *If $s \approx s'$, then $\mathbb{P}_{\mathcal{A}}(\pi(s, e_1, \ldots, e_n)) > 0$ iff $\mathbb{P}_{\mathcal{A}}(\pi(s', e_1, \ldots, e_n)) > 0$ for all edges $e_1, \ldots, e_n$.*

Proof. We prove this result by induction on n. The case $n = 0$ is obvious. Assume $n \geq 1$, and let s and s' be two states which are in the same region. Then, write:

$$
\begin{aligned}
\mathbb{P}_{\mathcal{A}}(\pi(s, e_1, \ldots, e_n)) > 0 &\Leftrightarrow \int_{t \in I(s,e_1)} \frac{\mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))}{\#\{I(s,e) \mid t \in I(s,e)\}} \mathrm{d}\mu_s(t) > 0 \\
&\Leftrightarrow \exists q,\ \int_{t \in I(s,e_1), s_t \in q} \frac{\mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))}{\#\{I(s,e) \mid t \in I(s,e)\}} \mathrm{d}\mu_s(t) > 0
\end{aligned}
$$

Assume that $\mathbb{P}_{\mathcal{A}}(\pi(s, e_1, \ldots, e_n)) > 0$, and take q such that

$$\int_{t \in I(s,e_1), s_t \in q} \frac{\mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))}{\#\{I(s,e) \mid t \in I(s,e)\}} \mathrm{d}\mu_s(t) > 0\,.$$

It means that there is a μ_s-measurable subset I of $\{t \in I(s,e_1) \mid s_t \in q\}$ such that $\mu_s(I) > 0$ and for all $t \in I$, $\mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n)) > 0$. Applying the induction hypothesis to s_t, we get that for all states s_1 in that region, $\mathbb{P}_{\mathcal{A}}(\pi(s_1, e_2, \ldots, e_n)) > 0$. As $\mu_s(\{t \mid s_t \in q\}) > 0$, we know that $\lambda(\{t \mid s_t \in q\}) > 0$, since the Lebesgue (resp. uniform discrete measure) is equivalent to μ_s (see page 5). By region equivalence, we get that $\lambda(\{t \mid s'_t \in q\}) > 0$. Finally, since μ_s is equivalent to the Lebesgue (resp. uniform discrete measure) (see page 5), we obtain that $\mu_{s'}(\{t \mid s'_t \in q\}) > 0$. Hence, applying the same kind of decomposition for $\mathbb{P}_{\mathcal{A}}(\pi(s', e_1, \ldots, e_n))$, we get that $\mathbb{P}_{\mathcal{A}}(\pi(s', e_1, \ldots, e_n)) > 0$. □

3.2 Probabilistic Semantics

We consider the logic LTL [Pnu77], 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 shortcuts 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, and we assume the reader is familiar with the semantics of LTL. Note that, given a symbolic path π and an LTL formula φ, either all concretizations (*i.e.*, concrete runs $\varrho \in \pi$) of π 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 (resp. positively) satisfies* φ from s_0 and we then write $\mathcal{A}, s_0 \models\!\!\approx_\forall \varphi$ (resp. $\mathcal{A}, s_0 \models\!\!\approx_\exists \varphi$), if

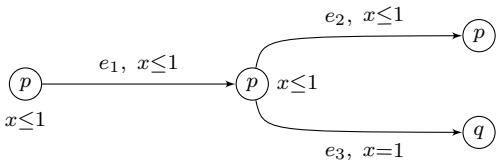
$$\begin{cases} \mathcal{A}, s_0 \models\!\!\approx_\forall \varphi & \stackrel{\text{def}}{\Leftrightarrow} \quad \mathbb{P}_{\mathcal{A}}(s_0, \varphi) = 1, \text{ resp.} \\ \mathcal{A}, s_0 \models\!\!\approx_\exists \varphi & \stackrel{\text{def}}{\Leftrightarrow} \quad \mathbb{P}_{\mathcal{A}}(s_0, \varphi) > 0\,. \end{cases}$$

Note that the positive satisfaction is the dual of the almost-sure satisfaction, as $\mathcal{A}, s_0 \models\!\!\approx_\exists \varphi$ iff $\mathcal{A}, s_0 \not\models\!\!\approx_\forall \neg\varphi$ for every LTL formula φ.

Remark 8. Our model of timed automata has no accepting locations. This is hence 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 $\mathbb{P}_{\mathcal{A}}(\varphi)$ we would rather evaluate the conditional probability

$\mathbb{P}_{\mathcal{A}}(s_0, \varphi \mid \psi)$.[7] Clearly enough, verifying that $\mathbb{P}_{\mathcal{A}}(s_0, \varphi \mid \psi) = 1$ (resp. > 0) in the automaton without accepting locations is equivalent to checking $\mathbb{P}_{\mathcal{A}}(s_0, \varphi) = 1$ (resp. > 0) in the automaton where accepting locations are those labelled with acc. Let us 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. In the next sections, each time this is necessary, we will explain how our constructions need to be changed in order to take into account accepting locations (or even any property ψ expressible in LTL).

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



If s_0 is the initial state (left-most location, and clock x set to 0), then $\mathcal{A}, s_0 \not\models_{\forall} \mathbf{G}\, p$. Indeed, in this example, the transition e_3 will unlikely happen, because its guard $x = 1$ is much too "small" compared to the guard $x \leq 1$ of the transition e_2.

Remark 10. In this paper, we focus on the simple logic LTL, but there would be a natural probabilistic interpretation for the untimed $\mathsf{CTL}^\star$ or μ-Calculus. Indeed, we could for instance say that $\mathcal{A}, s_0 \not\approx \mathbf{A}\, \varphi_1\, \mathbf{U}\, \varphi_2$ iff $\mathbb{P}_{\mathcal{A}}(s_0, \varphi_1\, \mathbf{U}\, \varphi_2) = 1$. And the methodology we develop here could apply for these more general logics as well, but technicalities due to this choice could hide the most important arguments.

4 A Tool: The Dimension

The aim of this section is to provide a tool, the dimension, (i) to solve the model-checking problem for both $\not\approx_{\forall}$ and $\not\approx_{\exists}$, and (ii) to give a topological interpretation of our probabilistic semantics.

Definition 11. *Let $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I}, \mathcal{L})$ be a timed automaton, s be a state of $\mathcal{A}$ and $(e_i)_{1 \leq i \leq n}$ be a finite sequence of edges of $\mathcal{A}$. We define (by induction on n) the dimension of the (symbolic) path $\pi(s, e_1, \ldots, e_n)$ as follows:*

- *If $n = 0$, we let $\dim_{\mathcal{A}}(\pi(s)) = 0$.*
- *Assume $n \geq 1$, the dimension of $\pi = \pi(s, e_1, \ldots, e_n)$ is then defined by:*

$$\dim_{\mathcal{A}}(\pi) = \begin{cases} \sup\{\dim(I(s, e_1)) + \dim_{\mathcal{A}}(\pi(s_1, e_2, \ldots, e_n)) \mid s_1 \in S(s, e_1)\} \\ \qquad\qquad \text{if } \dim(I(s, e_1)) = \dim(I(s)), \\ \bot \qquad\qquad \text{if } \dim(I(s, e_1)) < \dim(I(s)) \end{cases}$$

[7] Which is the probability of verifying φ, under the hypothesis that ψ holds.

where $\dim(I(\cdot))$ *represents the classical dimension of polyhedra embedded in a one-dimensional space, and with the conventions that* $n + \bot = \bot = \bot + n$*, for* $n \in \mathbb{N} \cup \{\bot\}$*,* $\bot < n$ *for every* $n \in \mathbb{N}$*, and* $\sup \emptyset = \bot$*.*

If $\dim_{\mathcal{A}}(\pi(s, e_1, \ldots, e_n)) = \bot$*, we say it is* undefined*, otherwise we say it is* defined*.*

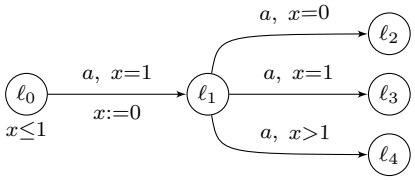
4.1 Properties of the Dimension

Let us notice that the definition of the dimension is *region independent*. Indeed one can easily be convinced that given s, $s' \in S_{\mathcal{A}}$ such that $s \approx s'$ we have that $\dim(I(s, e)) = \dim(I(s', e))$, for any edge e of $\mathcal{A}$. This observation directly leads to the following lemma.

Lemma 12. *Let* $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I}, \mathcal{L})$ *be a timed automaton,* s *and* s' *be two states of* $\mathcal{A}$ *such that* $s \approx s'$*, and* $(e_i)_{1 \leq i \leq n}$ *be a finite sequence of edges of* $\mathcal{A}$*. Then,*

$$\dim_{\mathcal{A}}(\pi(s, e_1, \ldots, e_n)) = \dim_{\mathcal{A}}(\pi(s', e_1, \ldots, e_n))\,.$$

Example 13. Consider the timed automaton on the next figure. For $i = 1, \ldots, 4$ let e_i denote the edge ending in ℓ_i and let s_0 be the state $(\ell_0, 0)$. Let us show that the dimension of the symbolic path $\pi(s_0, e_1, e_2)$ is $\bot$, whereas the dimension of the symbolic path $\pi(s_0, e_1, e_4)$ is 1.



We first establish the dimension of the symbolic path $\pi(s_0, e_1, e_2)$. By definition of the dimension:

$$\begin{aligned}
&\dim_{\mathcal{A}}(\pi(s_0, e_1, e_3)) = \\
&\qquad \sup_{s_1 \in S(s, e_1)} \{\dim(I(s, e_1)) + \dim_{\mathcal{A}}(\pi(s_1, e_3)) \mid \dim(I(s, e_1)) = \dim(I(s))\} \\
&\qquad = \sup\{\dim(\{1\}) + \dim_{\mathcal{A}}(\pi(s_1, e_2)) \mid s_1 = (\ell_1, 0)\} \\
&\qquad = \sup\{0 + \bot\} = \bot,
\end{aligned}$$

since we have that $\dim(I(s_1, e_2)) = \dim(\{0\}) = 0 < 1 = \dim(\mathbb{R}^+) = \dim(I(s_1))$, meaning that $\dim_{\mathcal{A}}(\pi(s_1, e_2)) = \bot$. Let us now compute the dimension of the symbolic path $\pi(s_0, e_1, e_4)$:

$$\begin{aligned}
&\dim_{\mathcal{A}}(\pi(s_0, e_1, e_4)) = \\
&\qquad \sup_{s_1 \in S(s, e_1)} \{\dim(I(s, e_1)) + \dim_{\mathcal{A}}(\pi(s_1, e_4)) \mid \dim(I(s, e_1)) = \dim(I(s))\} \\
&\qquad = \sup\{\dim(\{1\}) + \dim_{\mathcal{A}}(\pi(s_1, e_4)) \mid s_1 = (\ell_1, 0)\} \\
&\qquad = \sup\{0 + 1\} = 1,
\end{aligned}$$

since we have that:

$$\begin{aligned}\dim_{\mathcal{A}}(\pi(s_1,e_4)) &= \\ &\sup_{s_4\in S(s_1,e_4)} \{\dim(I(s_1,e_4)) + \dim_{\mathcal{A}}(\pi(s_4)) \mid \dim(I(s_1,e_4)) = \dim(I(s_1))\} \\ &= \sup\{\dim((1,+\infty)) + 0\} = \sup\{1+0\} = 1.\end{aligned}$$

4.2 Qualitative Probabilities and Dimension "Match"

In Section 2, we defined the region automaton associated to a timed automaton as a timed automaton itself. We can thus apply all machinery developed so far to the timed region automata. First, we prove that dimension and probabilities are closely related, when dealing with timed region automata.

Theorem 14. *Let $\mathcal{A}$ be a non-blocking timed automaton, and π be a symbolic path in $R(\mathcal{A})$. Then,*

$$\mathbb{P}_{R(\mathcal{A})}(\pi) > 0 \Leftrightarrow \dim_{R(\mathcal{A})}(\pi) \neq \bot\,.$$

Proof. We prove the equivalence by induction on the length of the symbolic path π. If the length of π is 0 (*i.e.*, $\pi = \pi(s)$ for some s), then $\dim(\pi) = 0 \neq \bot$. In that case, we also have that $\mathbb{P}_{R(\mathcal{A})}(\pi) = 1$, the equivalence is thus true.

We assume that $n \geq 1$ and that the induction hypothesis has been proved for symbolic paths of length strictly smaller than n, and we now let π be some symbolic path $\pi(s, e_1, \ldots, e_n)$ in $R(\mathcal{A})$. We assume that $\dim_{R(\mathcal{A})}(\pi) = \bot$. Then, either $\dim(I(s,e_1)) < \dim(I(s)) = 1$, or $\dim(I(s)) = 0$ and $I(s,e_1) = \emptyset$, or $\dim(I(s,e_1)) = \dim(I(s))$ (hence $I(s,e_1) \neq \emptyset$) and for every $s_1 \in S(s,e_1)$, $\dim_{R(\mathcal{A})}(\pi(s_1,e_2,\ldots,e_n)) = \bot$.

- In the first case, since μ_s is equivalent to the Lebesgue measure on $I(s)$ (see page 5), this implies that $\mu_s(I(s,e_1)) = 0$. Hence, as

$$\mathbb{P}_{R(\mathcal{A})}(\pi) \leq \int_{t\in I(s,e_1)} \mathbb{P}_{R(\mathcal{A})}(\pi(s_t,e_2,\ldots,e_n))\mathrm{d}\mu_s(t) \leq \mu_s(I(s,e_1))$$

  we get that $\mathbb{P}_{R(\mathcal{A})}(\pi) = 0$.
- In the second case, obviously $\mathbb{P}_{R(\mathcal{A})}(\pi) = 0$ by definition of the probability.
- In the last case, by induction hypothesis, we get that $\mathbb{P}_{R(\mathcal{A})}(\pi(s_1,e_2,\ldots,e_n)) = 0$ for every $s_1 \in S(s,e_1)$. Hence, $\mathbb{P}_{R(\mathcal{A})}(\pi) = 0$ by definition of $\mathbb{P}_{R(\mathcal{A})}$.

Note that this implication could hold for any timed automaton, and not only for region automata.

We assume that $\dim_{R(\mathcal{A})}(\pi) \neq \bot$. It implies that $\dim(I(s,e_1)) = \dim(I(s))$. By the equivalence of the Lebesgue on $I(s)$ (resp. uniform discrete) measure and μ_s (see page 5) we have that $\mu_s(I(s,e_1)) > 0$.

As we consider the timed region automaton $R(\mathcal{A})$, there exists q such that for every $t \in I(s, e_1)$, $s+t \in q$ (where $s \xrightarrow{t} s+t$). Two cases have to be considered: either q is a thin region, or q is a thick region. We call a region q *thin* if $\nu + \tau \not\approx \nu$ for every $\nu \in q$ and $\tau > 0$ (note that this is independent of the choice of $\nu \in q$). Otherwise, it is said *thick*. In case q is a thin region, both sets $I(s, e_1)$ and $S(s, e_1)$ reduce in a singleton. Let us denote by s_1 the unique point of $S(s, e_1)$. The probability of π is thus given by:

$$\mathbb{P}_{R(\mathcal{A})}(\pi) = \frac{1}{2} \cdot \mathbb{P}_{R(\mathcal{A})}(\pi(s_1, e_2, \ldots, e_n)) \cdot \mu_s(I(s, e_1)) \,.$$

Since $\dim_{R(\mathcal{A})}(\pi) \neq \bot$, we necessarily have that $\dim_{R(\mathcal{A})}(\pi(s_1, e_2, \ldots, e_n)) \neq \bot$. By induction hypothesis, $\mathbb{P}_{R(\mathcal{A})}(\pi(s_1, e_2, \ldots, e_n)) > 0$, and thus clearly $\mathbb{P}_{R(\mathcal{A})}(\pi) > 0$. Let us now consider the case where q is a thick region. As $\dim_{R(\mathcal{A})}(\pi) \neq \bot$, there exists $s_1 \in S(s, e_1)$ such that $\dim_{R(\mathcal{A})}(\pi(s_1, e_2, \ldots, e_n)) \neq \bot$. By induction hypothesis, $\mathbb{P}_{R(\mathcal{A})}(\pi(s_1, e_2, \ldots, e_n)) > 0$, and by Lemma 7, for every $s_t \in S(s, e_1)$, $\mathbb{P}_{R(\mathcal{A})}(\pi(s_t, e_2, \ldots, e_n)) > 0$.[8] Assume that $\mathbb{P}_{R(\mathcal{A})}(\pi) = 0$, then for almost every $t \in I(s, e_1)$ (w.r.t. measure μ_s), $\mathbb{P}_{R(\mathcal{A})}(\pi(s_t, e_2, \ldots, e_n)) = 0$, hence a contradiction. We conclude that $\mathbb{P}_{R(\mathcal{A})}(\pi) > 0$. □

Remark 15. Notice that Theorem 14 does not hold when π is a symbolic path in $\mathcal{A}$ (instead of $R(\mathcal{A})$). Indeed one can find a symbolic path π such that $\mathbb{P}_{\mathcal{A}}(\pi) = 0$, but $\dim_{\mathcal{A}}(\pi) \neq \bot$. Figure 1 on page 12 represents an automaton $\mathcal{A}$ (left) and its region automaton (right). Consider *e.g.* the path $\pi(s_0, e_1, e_2)$ in $\mathcal{A}$ where $s_0 = (\ell_0, 0)$ and e_1 (resp. e_2) is the edge ending in ℓ_1 (resp. ℓ_2). In $R(\mathcal{A})$, a single symbolic path corresponds to $\pi(s_0, e_1, e_2)$, namely the path going through $(\ell_1, x = 0)$. It is rather clear that the dimension of this symbolic path is undefined in $R(\mathcal{A})$.

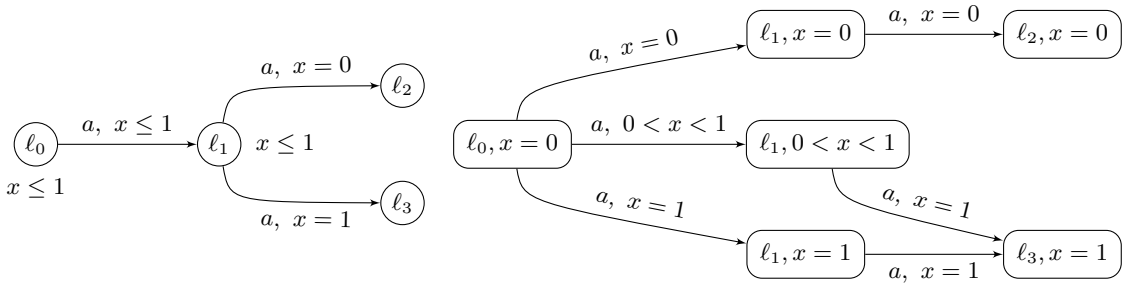


Fig. 1. Timed automaton $\mathcal{A}$ and its region automaton $R(\mathcal{A})$

[8] As $R(\mathcal{A})$ is a timed region automaton, all states in $S(s, e_1)$ are region equivalent.

Remark 16. To any finite symbolic path $\pi = \pi(s, e_1, \ldots, e_n)$ in $R(\mathcal{A})$, one can naturally associate a polyhedron[9] of $(\mathbb{R}_+)^n$:

$$\mathsf{Pol}(\pi) = \{(\tau_1, \ldots, \tau_n) \in (\mathbb{R}_+)^n \mid \varrho = s \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} \cdots s_n \in \mathsf{Runs}(R(\mathcal{A}), s)\}.$$

If defined, the dimension of π is indeed the real dimension of the polyhedron $\mathsf{Pol}(\pi)$.

Remark 17. It is worth noticing here that the definition of the dimension does not depend on the choice of the probabilistic measures that have been assigned to delays in the timed automaton. Thus the previous equivalence holds, whatever the probability measures over the delays are (as long as the hypotheses described on page 5 are satisfied).

4.3 Correctness of the Region Automaton w.r.t. Probabilities

We now compare probabilities in $\mathcal{A}$ and in $R(\mathcal{A})$ and establish that it is legal to consider $R(\mathcal{A})$ instead of $\mathcal{A}$. Of course, the measures we initially assign to $\mathcal{A}$ and $R(\mathcal{A})$ have to be related. Hence, if $\mu^{\mathcal{A}}$ (resp. $\mu^{R(\mathcal{A})}$) is the measure in $\mathcal{A}$ (resp. $R(\mathcal{A})$), we assume that for every state s in $\mathcal{A}$, $\mu^{\mathcal{A}}_s = \mu^{R(\mathcal{A})}_{\iota(s)}$. This is possible as one can easily be convinced that $I(s) = I(\iota(s))$. Under that assumption, we have the following result.

Lemma 18. *Let $\mathcal{A}$ be a non-blocking timed automaton. Assume measures in $\mathcal{A}$ and in $R(\mathcal{A})$ are related as described above. Let π be a symbolic path in $\mathcal{A}$. Then, $\iota(\pi)$[10] is a $\mathbb{P}_{R(\mathcal{A})}$-measurable set of runs in $R(\mathcal{A})$, and $\mathbb{P}_{\mathcal{A}}(\pi) = \mathbb{P}_{R(\mathcal{A})}(\iota(\pi))$.*

Proof. In this proof we will denote by e_i (resp. f_i) the transitions of $\mathcal{A}$ (resp. $R(\mathcal{A})$). The first point of the lemma is obvious. We prove the second property by induction on the length n of symbolic paths. When $n = 0$, this is obvious as, for every (ℓ, ν), there is a single state $((\ell, r), \nu)$ in $R(\mathcal{A})$ such that $\nu \in r$, and in that case, $\iota(\pi((\ell, \nu))) = \{\pi(((\ell, r), \nu))\}$. We assume the induction hypothesis holds for all paths of length strictly smaller than n. Let $\pi = \pi(s, e_1, \ldots, e_n)$ be a symbolic path in $\mathcal{A}$. We need to have some more notations (this will be technical, but rather simple). If s is a state, we write $s + t$ for the state reached from s after a delay of t. If s is a state of $\mathcal{A}$, we write $\iota(s)$ for its image in $R(\mathcal{A})$ (as argued before). We recall that if s is a state of $\mathcal{A}$, then $[s]$ is the region to which s belongs. If q is a state of the region automaton, we write n_q the number of edges that can be taken without delay from q in $R(\mathcal{A})$ (or equivalently in $\mathcal{A}$). If e_1 is a transition that can be taken from q without delay, we write $e_1(q)$ the single image region we reach after firing e_1 from q, and we write $q \models f_1$ if f_1 is the unique transition with guard checking that we are in q and corresponding to e_1 in $R(\mathcal{A})$.

[9] This construction is detailed in [BBBR07].

[10] Recall that, if ϱ is a run in $\mathcal{A}$, then $\iota(\varrho)$ is the image of ϱ in $R(\mathcal{A})$ (see page 5)

$$
\begin{aligned}
\mathbb{P}_{\mathcal{A}}(\pi) &= \frac{1}{2}\int_{t\in I(s,e_1)} \frac{\mathbb{P}_{\mathcal{A}}(\pi(s_t,e_2,\ldots,e_n))}{\#\{I(s,e)\mid t\in I(s,e)\}}\mathrm{d}\mu_s^{\mathcal{A}}(t)\\
&= \frac{1}{2}\int_{t\in I(s,e_1)} \sum_{\pi_t\in\iota(\pi(s_t,e_2,\ldots,e_n))} \frac{\mathbb{P}_{R(\mathcal{A})}(\pi_t)}{\#\{I(s,e)\mid t\in I(s,e)\}}\mathrm{d}\mu_s^{\mathcal{A}}(t) \qquad \text{by i.h.}\\
&= \frac{1}{2}\sum_q \int_{\substack{t\in I(s,e_1)\\ s+t\in q}} \sum_{\pi_t\in\iota(\pi(s_t,e_2,\ldots,e_n))} \frac{\mathbb{P}_{R(\mathcal{A})}(\pi_t)}{n_q}\mathrm{d}\mu_s^{\mathcal{A}}(t)\\
&= \frac{1}{2}\sum_q \int_{\substack{t\in I(s,e_1)\\ s+t\in q}} \sum_{(f_2,\ldots,f_n)\in\iota(e_1(q),e_2,\ldots,e_n)} \frac{\mathbb{P}_{R(\mathcal{A})}(\pi(\iota(s_t),f_2,\ldots,f_n))}{n_q}\mathrm{d}\mu_s^{\mathcal{A}}(t)\\
&= \frac{1}{2}\sum_q \int_{\substack{t\in I(\iota(s),f_1)\\ s+t\in q\\ q\models f_1\\ [s]\xrightarrow{f_1}e_1(q)}} \sum_{(f_2,\ldots,f_n)\in\iota(e_1(q),e_2,\ldots,e_n)} \frac{\mathbb{P}_{R(\mathcal{A})}(\pi(\iota(s)_t,f_2,\ldots,f_n))}{n_q}\mathrm{d}\mu_{\iota(s)}^{R(\mathcal{A})}(t)\\
&\qquad\qquad\qquad\qquad \text{by hypothesis on the measures}\\
&= \frac{1}{2}\sum_{\substack{q\models f_1\\ [s]\xrightarrow{f_1}e_1(q)\\ (f_2,\ldots,f_n)\in\iota(e_1(q),e_2,\ldots,e_n)}} \int_{t\in I(\iota(s),f_1)} \frac{\mathbb{P}_{R(\mathcal{A})}(\pi(\iota(s)_t,f_2,\ldots,f_n))}{n_q}\mathrm{d}\mu_{\iota(s)}^{R(\mathcal{A})}(t)\\
&= \sum_{\substack{q\models f_1\\ [s]\xrightarrow{f_1}e_1(q)\\ (f_2,\ldots,f_n)\in\iota(e_1(q),e_2,\ldots,e_n)}} \mathbb{P}_{R(\mathcal{A})}(\pi(\iota(s),f_1,\ldots,f_n))\\
&= \mathbb{P}_{R(\mathcal{A})}(\iota(\pi))
\end{aligned}
$$

where $(f_2,\ldots,f_n)\in\iota(e_1(q),e_2,\ldots,e_n)$ iff $(f_2,\ldots,f_n)$ is a finite sequence of transitions corresponding to $(e_2,\ldots,e_n)$ and which starts in $(e_1(q))$ (this is a state of $R(\mathcal{A})$). □

5 A Topological Interpretation

To strengthen our probabilistic semantics, we propose an equivalent topological characterization of the almost-sure (resp. positive) satisfiability of formulas. If an automaton almost-surely satisfies a property, the set of paths satisfying the formula should be big, and its complement small. As already pointed out in [VV06], the adequate topological notion corresponding to that requirement is not the notion of density, but better the notion of largeness.

5.1 Largeness and the Banach-Mazur Game

We refer to [Mun00] for bases in topology (topological space, neighbourhoud, *etc*). We recall here the notion of *largeness* and also provide its characterization in terms of *Banach-Mazur games* [Oxt57].

Some topological notions. We recall how topological spaces can be built, and first we define the notion of *neighbourhood system*.

Definition 19. *Let A be a set. A* neighbourhood system on A *is an application, denoted $\mathcal{N}$, which assigns a* filter[11] *$\mathcal{N}(a)$ (on the set A) to each $a \in A$ such that:*

- *for all $a \in A$ and for all $U \in \mathcal{N}(a)$, we have that $a \in U$, and*
- *for all $a \in A$ and for all $U \in \mathcal{N}(a)$, there exists $V \subseteq U$ such that for all $b \in V$ we have that $V \in \mathcal{N}(b)$.*

It is well-known that a topological space $(A, \mathcal{T})$ induces a neighbourhood system. Conversely given any set A together with a neighbourhood system $\mathcal{N}$, one can obtain a topology $\mathcal{T}_{\mathcal{N}}$ on A: a subset O of A is an *open set* if for all $a \in O$ we have that $O \in \mathcal{N}(a)$. It can be shown that $(A, \mathcal{T}_{\mathcal{N}})$ is a *topological space*, and that all topological spaces can be constructed that way.

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 $\mathcal{T}' \subseteq \mathcal{T}$ is called a *basis* for the topology $\mathcal{T}$ if every open set (*i.e.*, elements of $\mathcal{T}$) can be obtained as a union of elements of $\mathcal{T}'$. In this case, the elements of $\mathcal{T}'$ are called *basic opens*. A set $B \subseteq A$ is *nowhere dense* if the interior of the closure of B is empty, *i.e.*, $\mathring{\overline{B}} = \emptyset$. A set is *meager* if it is a countable union of nowhere dense sets. Finally, a set is *large* if its complement is meager.

Example 20. Let $\mathbb{R}$ be the set of real numbers equipped with its natural topology (whose basic open sets are the open intervals). The set of integer numbers $\mathbb{Z}$ is nowhere dense in $\mathbb{R}$. The set of rational numbers $\mathbb{Q}$ is dense (in $\mathbb{R}$) however $\mathbb{Q}$ is meager since it can be seen as a countable union of singletons (which are clearly nowhere dense sets). This implies that $\mathbb{R} \setminus \mathbb{Q}$ is large.

Although the notion of largeness is quite abstract, it admits a very nice characterization in terms of a two-player game, known as *Banach-Mazur game*. Conditions for determinacy of this game have been first conjectured by Mazur and proved by Banach for the set of reals (with the classical topology), and then proved in the more general context of arbitrary topological spaces by Oxtoby [Oxt57].

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) for any non-empty open set O of A, $\exists B \in \mathcal{B},\ B \subseteq O$. 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 [Oxt57, Remark 1]). Still a natural question is to know when the players have winning strategies. The following result gives a partial answer:

[11] In this context, a *filter* $\mathcal{N}$ is a non-empty subset of $\wp(A)$ which does not contain the empty set, which is closed by finite intersection, and $\mathcal{N}$ is closed by superset (*i.e.*, if B is in $\mathcal{N}$ and $B \subseteq C$, then $C \in \mathcal{N}$).

Theorem 21 (Banach-Mazur [Oxt57]). *Player 2 has a winning strategy in the Banach-Mazur game with target set C if and only if C is meager.*

Remark 22. Surprisingly, there is no relation between meager and open sets. Indeed one can identify topological spaces where open sets are meager (non meager open sets obviously exist). Let us consider the topological space $\big((0,1),\mathcal{T}\big)$ where $\mathcal{T} = \{\emptyset\} \cup \{(0,2^{-n}) \mid n \in \mathbb{N}\}$. In this topology all open sets are meager (one can be convinced that Player 2 can always enforce the (Banach-Mazur) game to reach the empty set).

Remark 23. Let us notice that by means of Theorem 21 one can easily be convinced that a non-empty open set is never meager. Indeed let $(A,\mathcal{T})$ be a topological space an $\mathcal{O}$ be a non-empty open set. By Theorem 21, $\mathcal{O}$ is meager if and only Player 2 has a winning strategy. Consider an instance of the Banach-Mazur game where $C = \mathcal{O}$. If the first choice of Player 1 is to pick $B \in \mathcal{B}$ such that $B \subseteq \mathcal{O}$ (such a B always exists by definition of the Banach-Mazur game), it is clear that Player 1 wins the game. In particular, Player 2 has no winning strategy for this game and thus $\mathcal{O}$ is not meager.

5.2 A Topology Based on the Dimension

For $\mathcal{A}$ a timed automaton, and s_0 a state, we define a neighbourhood system on $\mathsf{Runs}(\mathcal{A},s_0)$, which then induces a topology. Let ϱ be a finite run in $\mathsf{Runs}(\mathcal{A},s_0)$, and π_ϱ its unique corresponding symbolic path. We define the neighbourhood of ϱ as:

$$\mathcal{N}(\varrho) \stackrel{\text{def}}{=} \begin{cases} \big\{U \subseteq \mathsf{Runs}(\mathcal{A},s_0) \;\mid\; \pi_\varrho \subseteq U\big\} & \text{if } \dim(\pi_\varrho) \neq \bot, \\ \{\mathsf{Runs}(\mathcal{A},s_0)\} & \text{otherwise.} \end{cases}$$

Lemma 24. *Function $\mathcal{N}$ defines a neighbourhood system.*

Proof. Let us first prove that given $\varrho \in \mathsf{Runs}^{\mathrm{acc}}(\mathcal{A},s_0)$, we have that $\mathcal{N}(\varrho)$ is a filter on $\mathsf{Runs}^{\mathrm{acc}}(\mathcal{A},s_0)$. There are two cases to consider, (i) either $\dim(\pi_\varrho) \neq \bot$, or (ii) $\dim(\pi_\varrho) = \bot$. Case (ii) is obvious since $\mathcal{N}(\varrho)$ reduces in $\{\mathsf{Runs}^{\mathrm{acc}}(\mathcal{A},s_0)\}$. Let us consider case (i):

- $\emptyset \notin \mathcal{N}(\varrho)$. This is clearly true since for all $U \in \mathcal{N}(\varrho)$ we have that $\pi_\varrho \subseteq U$.
- Given U_1 and $U_2 \in \mathcal{N}(\varrho)$, we have to prove that $U_1 \cap U_2 \in \mathcal{N}(\varrho)$. By definition of $\mathcal{N}(\varrho)$, for $i = 1,2$, $\varrho \in \pi_\varrho \subseteq U_i$. Hence $U_1 \cap U_2 \in \mathcal{N}(\varrho)$.
- Given $U \in \mathcal{N}(\varrho)$ and $U \subseteq V \subseteq \mathsf{Runs}^{\mathrm{acc}}(\mathcal{A},s_0)$, we have to prove that $V \in \mathcal{N}(\varrho)$. This clearly holds by definition of $\mathcal{N}(\varrho)$.

Let us now prove that $\mathcal{N}$ defines a neighbourhood system. Given $\varrho \in \mathsf{Runs}^{\mathrm{acc}}(\mathcal{A},s_0)$ and $U \in \mathcal{N}(\varrho)$, we clearly have that $\varrho \in U$. It remains to prove that for all $\varrho \in \mathsf{Runs}^{\mathrm{acc}}(\mathcal{A},s_0)$, for all $U \in \mathcal{N}(\varrho)$, there exists $V \subseteq U$ such that for all $\varrho' \in V$,

$V \in \mathcal{N}(\varrho')$. Again two situations are possible. First let us consider the case where $\dim(\pi_\varrho) \neq \bot$. In that case, fix $V = \pi_\varrho$, then, obviously $V \in \mathcal{N}(\varrho')$ for all $\varrho' \in \pi_\varrho$ (or equivalently $\pi_{\varrho'} = \pi_\varrho$). Then let us consider the case where $\dim(\pi_\varrho) = \bot$. Take then $V = \mathsf{Runs}^{\mathrm{acc}}(\mathcal{A}, s_0)$ to obtain the desired property. □

We write $\mathcal{T}_\mathcal{N}$ for the topology induced by the neighbourhood system $\mathcal{N}$ on $\mathsf{Runs}(\mathcal{A}, s_0)$. The following lemma characterizes the topology $\mathcal{T}_\mathcal{N}$.

Lemma 25. *The neighbourhood system defined by $\mathcal{N}$ induces a topology whose basic open sets are the π's such that $\dim(\pi) \neq \bot$, and the set $\mathsf{Runs}(\mathcal{A}, s_0)$.*

Proof. Let $\mathcal{T}_\mathcal{N}$ be the topology induced by the neighbourhood system defined by $\mathcal{N}$. Let $\mathcal{T}$ be the topology whose basic open sets are the π's such that $\dim(\pi) \neq \bot$, and the set $\mathsf{Runs}^{\mathrm{acc}}(\mathcal{A}, s_0)$. We would like to prove that those two topologies are equivalent, *i.e.*, $\mathcal{T}_\mathcal{N} = \mathcal{T}$.

Let us first prove that $\mathcal{T}_\mathcal{N} \subseteq \mathcal{T}$. Let $\mathcal{O}$ be an element of $\mathcal{T}_\mathcal{N}$. By definition of $\mathcal{T}_\mathcal{N}$, this means that $\forall \varrho \in \mathcal{O}$ we have that $\mathcal{O} \in \mathcal{N}(\varrho)$. Assume $\mathcal{O} \neq \mathsf{Runs}^{\mathrm{acc}}(\mathcal{A}, s_0)$, we want to prove that $\mathcal{O} \in \mathcal{T}$. For a contradiction, assume that $\mathcal{O}$ is not an union of π_i's whose dimension is different from $\bot$. We will distinguish two cases either (i) $\mathcal{O}$ contains a non-empty[12] symbolic path $\pi_\bot$ whose dimension is $\bot$, or (ii) $\mathcal{O}$ does not contain such a path. In case (i) we obtain a contradiction since given $\varrho \in \pi_\bot$ we have that $\mathcal{N}(\varrho) = \mathsf{Runs}^{\mathrm{acc}}(\mathcal{A}, s_0) \neq \mathcal{O}$. In case (ii) there exists a symbolic path π such that $\dim(\pi) \neq \bot$ and $\emptyset \neq \mathcal{O} \cap \pi \subsetneq \pi$. Let ϱ_0 be a run in $\mathcal{O} \cap \pi$, we know that any neighbourhood of ϱ_0 contains π, thus $\mathcal{O} \notin \mathcal{N}(\varrho_0)$. This is a contradiction.

Let us now prove that $\mathcal{T} \subseteq \mathcal{T}_\mathcal{N}$. Given π such that $\dim(\pi) \neq \bot$, it is clear that for all $\varrho \in \pi$ we have that $\varrho \in \mathcal{N}(\varrho)$. The same obviously holds for $\mathsf{Runs}^{\mathrm{acc}}(\mathcal{A}, s_0)$. This proves the second implication since given $\mathcal{O} \in \mathcal{T}$ (*i.e.*, $\mathcal{O} = \cup_i \pi_i$), we have that any π_i is included in $\mathcal{O}$. □

Let us give an example in order to illustrate our topology on finite paths.

Example 26. The timed automaton of Example 13 on page 10 contains five symbolic paths starting from $s_0 = (\ell_0, \nu_0)$: $\pi_0 \stackrel{\text{def}}{=} \pi(s_0)$, $\pi_1 \stackrel{\text{def}}{=} \pi(s_0, e_1)$, and $\pi_j \stackrel{\text{def}}{=} \pi(s_0, e_1, e_j)$ for $j = 2, \ldots, 4$. One can be convinced that only π_0, π_1 and π_4 have a dimension different from $\bot$. In particular, π_0, π_1 and π_4 are open sets. Hence $(\pi_0 \cup \pi_1 \cup \pi_4)^c = (\pi_2 \cup \pi_3)$ is a closed set. Moreover π_2 is nowhere dense, as $\overset{\circ}{\overline{\pi_2}} = \overset{\circ}{\overbrace{\pi_2 \cup \pi_3}} = \emptyset$ and the same holds for π_3. Hence, $(\pi_2 \cup \pi_3)$ is meager and thus, $\pi_0 \cup \pi_1 \cup \pi_4$ is large.

We recall that a topological space $(A, \mathcal{T})$ is a *Baire space* if every non-empty open set in $\mathcal{T}$ is not meager.[13] *E.g.*, $\mathbb{R}$ with the natural topology is a Baire space,

[12] *I.e.*, there exists a symbolic path $\pi_\bot$ such that $\pi_\bot \neq \emptyset$, $\dim(\pi_\bot) = \bot$ and $\pi_\bot \subseteq \mathcal{O}$.

[13] In modern definitions, a topological space is a Baire space if each countable unions of closed sets with an empty interior has an empty interior. However, our definition is equivalent, see [Mun00, p.295].

but $\mathbb{Q}$, with the natural topology, is not a Baire space. However, by means of the Banach-Mazur game, one can show that the topological space $(\mathsf{Runs}(\mathcal{A}, s_0), \mathcal{T}_{\mathcal{N}})$ is a Baire space.

Proposition 27. *Let $\mathcal{A}$ be a timed automaton, and s_0 be a state of $\mathcal{A}$. The topological space $(\mathsf{Runs}(\mathcal{A}, s_0), \mathcal{T}_{\mathcal{N}})$ is a Baire space.*

Proof. We prove that every non-empty basic open set in $\mathcal{T}$ is not meager. A basic open set is a symbolic path π whose dimension is different from $\bot$, or the whole set $\mathsf{Runs}(\mathcal{A}, s_0)$. Using Theorem 21, we prove that π is not meager by proving that Player 2 does not have a winning strategy for the Banach-Mazur game where $C = \pi$. Let us first notice that $\pi \in \mathcal{B}$, the set of sets we play with. Indeed the second point of the definition of the Banach-Mazur game (Definition 5.1) ensures that for any non-empty open set O there exists $B \in \mathcal{B}$ such that $B \subseteq O$, thus there must exist $B \in \mathcal{B}$ such that $B \subseteq \pi$. Moreover the first point of the definition of the Banach-Mazur game imposes that $\mathring{B} \neq \emptyset$, since there is no non-empty open set strictly included in π, we necessarily have that $B = \pi$. Hence if the first move of Player 1 is to choose π, this ends the game since both players will only have to pick π forever. Player 1 thus wins the game. This proves that π is not meager. □

Remark 28. To handle accepting locations, we just need to consider the topology induced by $\mathcal{T}_{\mathcal{N}}$ on the set of runs satisfying ψ (recall Remark 8, page 8).

6 Model-Checking (for the Probabilistic Semantics)

The almost-sure and the positive model-checking problems admit a nice topological characterization, which we summarize in the next theorem.

Theorem 29. *Let $\mathcal{A}$ be a non-blocking timed automaton, s a state of $\mathcal{A}$, and φ an* LTL *formula. Then,*

$$\begin{aligned} \mathcal{A}, s \models\!\!\approx_\forall \varphi \quad &\overset{def}{\Leftrightarrow} \quad \textit{w.r.t. } \mathbb{P}_{\mathcal{A}}\textit{, almost all paths starting in } s \textit{ in } \mathcal{A} \textit{ satisfy } \varphi, \\ &\Leftrightarrow \quad \textit{the paths of } R(\mathcal{A}) \textit{ starting in } \iota(s) \textit{ not satisfying } \varphi \\ &\qquad\quad \textit{have an undefined dimension,} \\ &\Leftrightarrow \quad \textit{the set of paths of } R(\mathcal{A}) \textit{ starting in } \iota(s) \textit{ satisfying } \varphi \\ &\qquad\quad \textit{is (topologically) large.} \end{aligned}$$

Proof. The first equivalence is a corollary of Lemma 18 and Theorem 14.

The second is not a direct consequence of previous lemmas or propositions and requires some more work. Let us first assume that $\mathcal{A}, s_0 \models\!\!\approx_\forall \varphi$. Using the first equivalence, this means that from s_0 every symbolic path π in $R(\mathcal{A})$ such that $\dim_{R(\mathcal{A})}(\pi) \neq \bot$ satisfies φ. We prove that the set $[\![\varphi]\!] = \{\varrho \in \mathsf{Runs}(R(\mathcal{A}), s_0) \mid \varrho \models \varphi\}$ is large. In order to do this, we use a Banach-Mazur game and prove that the complement $[\![\varphi]\!]^c = [\![\neg\varphi]\!]$ is meager. Consider an instance of the Banach-Mazur

game with a set $\mathcal{B}$ of open sets and where $C = [\![\neg\varphi]\!]$. By hypothesis, $[\![\neg\varphi]\!] \subseteq \cup_{\dim_{R(\mathcal{A})}(\pi)=\bot}\pi$. Let $B_1 \in \mathcal{B}$ be Player 1's first move. By definition of the Banach-Mazur game, $\mathring{B}_1 \neq \emptyset$. In particular, B_1 contains some symbolic path π_0 such that $\dim_{R(\mathcal{A})}(\pi_0) \neq \bot$. We necessarily have $\pi_0 \in \mathcal{B}$, since there is no non-empty open set strictly included in π_0. Player 2 is thus forced to pick $B_2 = \pi_0$. This ends the game, since the following B_i's ($i \geq 2$) can only be equal to π_0. Clearly $\bigcap_{i=1}^{\infty} B_i = \pi_0$, hence $\bigcap_{i=1}^{\infty} B_i \cap [\![\neg\varphi]\!] = \emptyset$, and Player 2 has a winning strategy for this game. Theorem 21 implies that $[\![\neg\varphi]\!]$ is meager, thus $[\![\varphi]\!]$ is large.

Let us now prove the second implication. For a contradiction we assume that $\mathcal{A} \not\approx_\forall \varphi$. This means that there exists a symbolic path π with defined dimension which does not satisfy φ. Hence: $\exists\pi,\ \exists\varrho \in \pi\ \big(\dim_{R(\mathcal{A})}(\pi) \neq \bot\big) \wedge \big(\varrho \not\models \varphi\big)$ or, equivalently (since φ is an LTL formula) $\exists\pi,\ \ \forall\varrho \in \pi\ \big(\dim_{R(\mathcal{A})}(\pi) \neq \bot\big) \wedge \big(\varrho \not\models \varphi\big)$. In particular $\{\varrho \in \mathsf{Runs}(R(\mathcal{A}), s_0) \mid \varrho \not\models \varphi\}$ contains an open set, which is not meager by Proposition 27. Since the notion of being meager is closed under subset, the set $\{\varrho \in \mathsf{Runs}(R(\mathcal{A}), s_0) \mid \varrho \not\models \varphi\}$ is not meager. Hence the set $\{\varrho \in \mathsf{Runs}(R(\mathcal{A})) \mid \varrho \models \varphi\}$ is not large which is a contradiction. This concludes the proof. □

Remark 30. – Note that the previous theorem can be rewritten for the satisfaction relation $\not\approx_\exists$ since, as mentioned in Subsection 3.2, it is the dual of $\not\approx_\forall$.
– To handle accepting states in the previous theorem, it would be sufficient to quantify only over paths in $R(\mathcal{A})$ which are accepting.

Theorem 31. *Over finite timed words, the almost-sure and the positive* LTL *model-checking problems over non-blocking timed automata are* PSPACE*-Complete.*

Proof. We only provide the proof for the almost-sure model-checking problem, the positive model-checking problem is dual and can thus be handled similarly.

The PSPACE-Hardness follows from the PSPACE-Hardness of LTL model checking over finite automata.

To describe a PSPACE algorithm for deciding the almost-sure model checking of LTL, we first color each edge of $R(\mathcal{A})$ as follows: if $e = q \xrightarrow{g,a,Y} q'$ is a transition in $R(\mathcal{A})$, then we color it in red whenever $\mu_s(I(s,e)) = 0$ for some $s \in q$ (note that this property is independent on the choice of $s \in q$, and that it is equivalent to $\dim(I(s,e)) < \dim(I(s))$), and we color it in blue otherwise.

Lemma 32. *Let π be a symbolic path. Then,* $\dim_{R(\mathcal{A})}(\pi) = \bot$ *iff at least one of the edges of π is red.*

Proof. We do the proof by induction on the length of π. The case when π has length 0 is obvious (as $\dim_{R(\mathcal{A})}(\pi) = 0 \neq \bot$).

We assume now that $\pi = \pi(s, e_1, \ldots, e_n)$ with $n \geq 1$. If $\dim_{R(\mathcal{A})}(\pi) = \bot$, then either $\dim(I(s,e_1)) < \dim(I(s))$, or for all $s_1 \in S(s,e_1)$, $\dim_{R(\mathcal{A})}(s_1, e_2, e_n) = \bot$. In the first case, this precisely means that e_1 has been colored in red. In the second

case, by induction hypothesis, we get that one of the e_i's for $2 \leq i \leq n$ is colored in red.

Now assume that one of the e_i's is red. We first assume that e_1 is red. Then, by definition of the dimension, we get that $\dim_{R(\mathcal{A})}(\pi) = \bot$. Otherwise, for every $s_1 \in S(s, e_1)$, the symbolic path $\pi(s_1, e_2, \ldots, e_n)$ has a red edge. By induction hypothesis, we get that $\dim_{R(\mathcal{A})}(\pi(s_1, e_2, \ldots, e_n)) = \bot$. By definition of the dimension, we thus get that $\dim_{R(\mathcal{A})}(\pi(s, e_1, \ldots, e_n)) = \bot$. □

Now, applying Theorem 29, to decide whether $\mathcal{A} \not\approx_\forall \varphi$, it is sufficient to guess a path in $R(\mathcal{A})$ which has defined dimension (*i.e.*, does not contain any red edge), and does not satisfy φ. Using what precedes, it is thus sufficient to find a counter-example for φ in $R(\mathcal{A})$ restricted to blue edges. Everything can be guessed non-deterministically in PSPACE. Indeed, there is no need to construct *a priori* the whole graph $R(\mathcal{A})$, a counter-example can be guessed on-the-fly. Moreover, to guess such a path, we only need to store two consecutive locations of $R(\mathcal{A})$ and a counter bounded by the length of a counter-example. In the case of LTL, the size of a counter-example can be bounded by a polynom in the size of the graph $R(\mathcal{A})$ and exponential in the size of the formula φ, hence in that case both exponential in the original timed automaton and in the size of the formula; it can thus be stored in polynomial space. The model-checking problem is thus in coNPSPACE, hence in PSPACE. □

7 Conclusion

In this paper we proposed a probabilistic semantics for timed automata, which rules out unlikely sequences of events in the system. 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. It naturally raises (qualitative) model-checking questions, for instance whether the probability that a property holds is 1. We propose a characterization of this model-checking question using the topological concept of largeness, based on a notion of dimension. This characterization has been used to establish decidability of the model-checking problem with our non-standard semantics, and prove that it is as hard as ordinary LTL model-chacking (PSPACE-Complete).

The method we 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 (we already mentioned the logic CTL* or the μ-Calculus). It could also be applied to various classes of hybrid systems with a finite bisimulation quotient [HMR05].

We have plenty of further works, including the non-trivial extension to infinite timed words, the model checking of timed logics, the quantitative analysis of this model (for instance, computing the exact, or approximate, probability of satisfying a given property, *etc*), control problems, and many more.

References

[ACD91] Rajeev Alur, Costas Courcoubetis, and David Dill. Model-checking for probabilistic real-time systems. In *Proc. 18th International Colloquium on Automata, Languages and Programming (ICALP'91)*, volume 510 of *Lecture Notes in Computer Science*, pages 115–126. Springer, 1991.

[ACD92] Rajeev Alur, Costas Courcoubetis, and David Dill. Verifying automata specifications of probabilistic real-time systems. In *Real-Time: Theory in Practice, Proc. REX Workshop 1991*, volume 600 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 1992.

[AD94] Rajeev Alur and David Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[ALM05] Rajeev Alur, Salvatore La Torre, and P. Madhusudan. Perturbed timed automata. In *Proc. 8th International Workshop on Hybrid Systems: Computation and Control (HSCC'05)*, volume 3414 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 2005.

[BBBR07] Patricia Bouyer, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On the optimal reachability problem. *Formal Methods in System Design*, 2007. To appear.

[BHHK03] Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):524–541, 2003.

[BMR06] Patricia Bouyer, Nicolas Markey, and Pierre-Alain Reynier. Robust model-checking of timed automata. In *Proc. 7th Latin American Symposium on Theoretical Informatics (LATIN'06)*, volume 3887 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2006.

[DDMR04] Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François 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)*, volume 3253 of *Lecture Notes in Computer Science*, pages 118–133. Springer, 2004.

[DDR04] Martin De Wulf, Laurent Doyen, and Jean-François Raskin. Almost ASAP semantics: From timed models to timed implementations. In *Proc. 7th International Workshop on Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *Lecture Notes in Computer Science*, pages 296–310. Springer, 2004.

[DGJP03] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Approximating labelled Markov processes. *Information and Computation*, 184(1):160–200, 2003.

[DGJP04] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.

[Dij71] Edsger W. Dijkstra. Hierarchical ordering of sequencial processes. *Acta Informatica*, 1(2):115–138, 1971.

[DOTY96] Conrado Daws, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. The tool Kronos. In *Proc. Hybrid Systems III: Verification and Control (1995)*, volume 1066 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 1996.

[GHJ97] Vineet Gupta, Thomas A. Henzinger, and Radha Jagadeesan. Robust timed automata. In *Proc. International Workshop on Hybrid and Real-Time Systems (HART'97)*, volume 1201 of *Lecture Notes in Computer Science*, pages 331–345. Springer, 1997.

[HMR05] Thomas A. Henzinger, Rupak Majumdar, and Jean-François Raskin. A classification of symbolic transition systems. *ACM Transactions on Computational Logic*, 6(1):1–32, 2005.

[HR00] Thomas A. Henzinger and Jean-François Raskin. Robust undecidability of timed and hybrid systems. In *Proc. 3rd International Workshop on Hybrid Systems: Computation and Control (HSCC'00)*, volume 1790 of *Lecture Notes in Computer Science*, pages 145–159. Springer, 2000.

[JPQ05] Marcin Jurdiński, Doron Peled, and Hongyang Qu. Calculating probabilities of real-time test cases. In *Proc. 5th International Workshop on Formal Approaches to Software Testing (FATES'05)*, volume 3997 of *Lecture Notes in Computer Science*, pages 134–151. Springer, 2005.

[KNP04] Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM 2.0: A tool for probabilistic model checking. In *Proc. 1st International Conference on the Quantitative Evaluation of Systems*, pages 322–323. IEEE Computer Society Press, 2004.

[KNSS02] Marta Z. Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 282(1):101–150, 2002.

[LPY97] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *Journal of Software Tools for Technology Transfer*, 1(1–2):134–152, 1997.

[Mun00] James R. Munkres. *Topology*. Prentice Hall, 2nd edition, 2000.

[Oxt57] John 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.

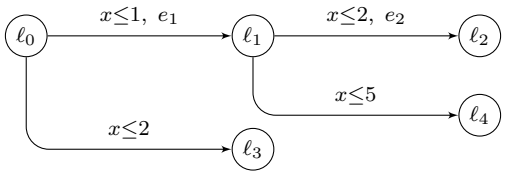
[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proc. 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*, pages 46–57. IEEE Computer Society Press, 1977.

[Pur98] Anuj Puri. Dynamical properties of timed automata. In *Proc. 5th International Symposium on Formal techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98)*, volume 1486 of *Lecture Notes in Computer Science*, pages 210–227. Springer, 1998.

[Spr04] Jeremy Sproston. Model checking for probabilistic timed systems. In *Validation of Stochastic Systems – A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, pages 189–229. Springer, 2004.

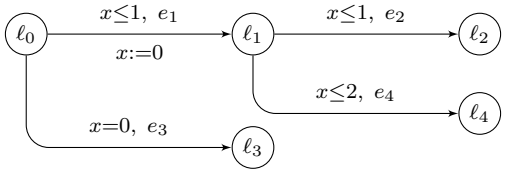
[VV06] Daniele Varacca and Hagen Völzer. Temporal logics and model checking for fairly correct systems. In *Proc. 21st Annual Symposium on Logic in Computer Science (LICS'06)*, pages 389–398. IEEE Computer Society Press, 2006.

Example of probability computation (Example 5 page 6) in more details.



$$\begin{aligned}
\mathbb{P}(\pi((\ell_0,0),e_1,e_2)) &= \frac{1}{2}\int_{t\leq 1}\frac{\mathbb{P}(\pi((\ell_1,t),e_2))}{2}\mathrm{d}\mu_{(\ell_0,0)}(t)\\
&= \frac{1}{4}\int_{t\leq 1}\Big(\frac{1}{2}\int_{t\leq u\leq 2}\frac{\mathbb{P}((\ell_2,u))}{2}\mathrm{d}\mu_{(\ell_1,t)}(u)\Big)\mathrm{d}\mu_{(\ell_0,0)}(t)\\
&= \frac{1}{16}\int_{t\leq 1}\Big(\int_{t\leq u\leq 2}\frac{1}{2}\mathrm{d}\mu_{(\ell_1,t)}(u)\Big)\mathrm{d}\mu_{(\ell_0,0)}(t)\\
&= \frac{1}{32}\int_{t\leq 1}\Big(\int_{t\leq u\leq 2}\frac{1}{5-t}\mathrm{d}\lambda(u)\Big)\frac{1}{2}\mathrm{d}\lambda(t)\\
&= \frac{1}{64}\int_{t\leq 1}\frac{2-t}{5-t}\mathrm{d}\lambda(t)\\
&= \frac{1}{64}\int_{t\leq 1}1-\frac{3}{5-t}\mathrm{d}\lambda(t)\\
&= \frac{1}{64}\Big[t+3\log(5-t)\Big]_0^1\\
&= \frac{1}{64}\Big(1-3\log\Big(\frac{5}{4}\Big)\Big)
\end{aligned}$$

Another example of probability computation. Consider the following timed automaton:



We assume we have the uniform distribution over all delays. Then, we can compute

$$\mathbb{P}(\pi((\ell_0,0),e_1,e_2)) \quad =$$

$$\frac{1}{2}\left(\int_0^0\frac{\mathbb{P}(\pi((\ell_1,0),e_2))}{2}\mathrm{d}\mu_{(\ell_0,0)}(t)+\int_0^1\mathbb{P}(\pi((\ell_1,0),e_2))\mathrm{d}\mu_{(\ell_0,0)}(t)\right)$$

ii

$$
\begin{aligned}
&= \frac{1}{2}\left(\int_0^1 \left(\frac{1}{2}\int_0^1 \frac{\mathbb{P}(\pi((\ell_2,u)))}{2}\mathrm{d}\mu_{(\ell_1,0)}(u)\right)\mathrm{d}\mu_{(\ell_0,0)}(t)\right)\\
&= \frac{1}{2}\left(\int_0^1 \left(\frac{1}{2}\int_0^1 \frac{\mathbb{P}(\pi((\ell_2,u)))}{2}\frac{1}{2}\mathrm{d}\lambda(u)\right)\mathrm{d}\lambda(t)\right)\\
&= \frac{1}{16}\left(\int_0^1 \left(\int_0^1 \frac{1}{2}\mathrm{d}\lambda(u)\right)\mathrm{d}\lambda(t)\right) = \frac{1}{32}
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{P}(\pi((\ell_0,0),e_1,e_4)) \quad &= \\
&\frac{1}{2}\left(\int_0^0 \frac{\mathbb{P}(\pi((\ell_1,0),e_4))}{2}\mathrm{d}\mu_{(\ell_0,0)}(t) + \int_0^1 \mathbb{P}(\pi((\ell_1,0),e_4))\mathrm{d}\mu_{(\ell_0,0)}(t)\right)\\
&= \frac{1}{2}\left(\int_0^1 \left(\frac{1}{2}\left(\int_0^1 \frac{\mathbb{P}(\pi((\ell_4,u)))}{2}\mathrm{d}\mu_{(\ell_1,0)}(u) + \int_1^2 \mathbb{P}(\pi((\ell_4,u)))\mathrm{d}\mu_{(\ell_1,0)}(u)\right)\right)\mathrm{d}\mu_{(\ell_0,0)}(t)\right)\\
&= \frac{1}{2}\left(\int_0^1 \left(\frac{1}{2}\left(\int_0^1 \frac{\mathbb{P}(\pi((\ell_4,u)))}{2}\frac{1}{2}\mathrm{d}\lambda(u) + \int_1^2 \mathbb{P}(\pi((\ell_4,u)))\frac{1}{2}\mathrm{d}\lambda(u)\right)\right)\mathrm{d}\lambda(t)\right)\\
&= \frac{1}{4}\int_0^1 \left(\int_0^1 \frac{1}{8}\mathrm{d}\lambda(u) + \int_1^2 \frac{1}{4}\mathrm{d}\lambda(u)\right)\mathrm{d}t = \frac{3}{32}.
\end{aligned}
$$

since $\mu_{(\ell_0,0)} = \lambda$ and $\mu_{(\ell_1,0)} = \frac{\lambda}{2}$.