Christel Baier, Nathalie Bertrand, Patricia Bouyer,

Thomas Brihaye, and Marcus Größer

Probabilistic and Topological Semantics for Timed Automata

Research Report LSV-07-26

30 August 2007

Laboratoire Spécification et Vérification

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

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

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 [AD94] 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 [DDR04]. 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 [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).

[⋆] 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).

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 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 [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 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 [VV06]) 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 [GHJ97,HR00] 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 [Spr04] 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.

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. 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:

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

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.

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 [AD94].

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 $R_{\mathcal{A}}$. If r is a region, we denote by $\mathsf{cell}(r)$ the smallest guard defined with constants smaller than M, and which contains r. We denote $\mathsf{cell}(\mathsf{R}(\mathcal{A}))$ the set of all $\mathsf{cell}(r)$ for $r \in R_{\mathcal{A}}$.

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 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 $\nu' \in r'$.*

We recover the usual region automaton of [AD94] 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

[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 regions that are crossed. If ϱ is a run in $\mathcal{A}$, then 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 [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:

1. $\mu_s(I(s)) = \mu_s(\mathbb{R}_+) = 1$;[5]
2. 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').

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 $\mathbb{P}_{\mathcal{A}}(\pi(s)) = \frac{1}{2}$.

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

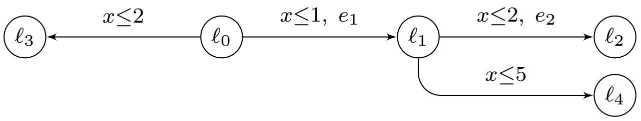
[6] Two measures ν and ν' are *equivalent* whenever for each measurable set A, $\nu(A) = 0 \Leftrightarrow \nu'(A) = 0$.

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 $\mathbb{P}(\pi((\ell_0, 0), e_1e_2)) = \frac{1}{64}\big(1 - 3\log\big(\frac{5}{4}\big)\big)$ 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. $[t, 5]$). The complete computation is presented in the appendix.

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 denote by $\mathbb{1}_I$ the characteristic function of set $I \subseteq \mathbb{R}_+$. Then,

$$\begin{aligned}
\mathbb{P}_{\mathcal{A}}(\mathsf{Runs}^n(\mathcal{A}), s) &= \sum_{e_1,\ldots,e_n} \mathbb{P}_{\mathcal{A}}(\pi(s, e_1, \ldots, e_n)) \\
&= \sum_{e_1,\ldots,e_n} \frac{1}{2} \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))\, \mathrm{d}\mu_s(t) \\
&= \frac{1}{2} \sum_{e_1} \int_{t \in I(s,e_1)} p_{s+t}(e_1) \sum_{e_2,\ldots,e_n} \mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \ldots, e_n))\, \mathrm{d}\mu_s(t) \\
&= \frac{1}{2} \sum_{e_1} \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}_{\mathcal{A}}(\mathsf{Runs}^{n-1}(\mathcal{A}, s_t))\, \mathrm{d}\mu_s(t) \\
&= \frac{1}{2} \sum_{e_1} \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \frac{1}{2^n}\, \mathrm{d}\mu_s(t) \qquad \text{by induction hypothesis}
\end{aligned}$$

[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.

$$
\begin{aligned}
&= \frac{1}{2^{n+1}} \int_{t \in I(s)} \left(\sum_{e_1} p_{s+t}(e_1)\, \mathbb{1}_{I(s,e_1)}(t) \right) \mathrm{d}\mu_s(t) \\
&= \frac{1}{2^{n+1}} \int_{t \in I(s)} \mathrm{d}\mu_s(t) \\
&= \frac{1}{2^{n+1}} \qquad \text{since } \mu_s(I(s)) = 1 \text{ (see condition on page 5).}
\end{aligned}
$$

This concludes the proof. □

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 $\mathcal{A}$, $\mu_s^{\mathcal{A}} = \mu_{\iota(s)}^{\mathsf{R}(\mathcal{A})}$.[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 $t \in \mathbb{R}_+$ $p_{s+t}^{\mathcal{A}} = p_{\iota(s)+t}^{\mathsf{R}(\mathcal{A})}$. Under those assumptions, we have the following result.

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 $\mathbb{P}_{\mathcal{A}}(\pi) = \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\iota(\pi))$.*

Proof. In this proof we will denote by e_i (resp. f_i) the transitions of $\mathcal{A}$ (resp. $\mathsf{R}(\mathcal{A})$). We make the hypothesis on $p^{\mathcal{A}}$ and $p^{\mathsf{R}(\mathcal{A})}$ more precise: if s is a state of $\mathcal{A}$ and $t \in \mathbb{R}_+$, for every edge e, there is a single corresponding transition f which leaves the state $[s]$ in $\mathsf{R}(\mathcal{A})$ ($[s]$ is the single state of $\mathsf{R}(\mathcal{A})$ such that $\iota(s) \in [s]$), and such that this transition f is enabled from $\iota(s) + t$ iff e is enabled from $s + t$; the hypothesis then says that $p_{s+t}^{\mathcal{A}}(e) = p_{\iota(s)+t}^{\mathsf{R}(\mathcal{A})}(f)$.

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 $\mathsf{R}(\mathcal{A})$ 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 $\mathsf{R}(\mathcal{A})$ (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 for the number of edges that can be taken without delay from q in $\mathsf{R}(\mathcal{A})$ (or equivalently in $\mathcal{A}$). If e_1 is a transition that can be taken from q without delay, we denote by $e_1(q)$ the single image region we reach

[8] Here (and in the sequel), we abuse notations and use $\iota(s)$ for $\iota(\pi(s))$.

[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 5).

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 $\mathsf{R}(\mathcal{A})$.

$$\begin{aligned}
\mathbb{P}_{\mathcal{A}}(\pi) &= \frac{1}{2} \int_{t \in I(s,e_1)} p^{\mathcal{A}}_{s+t}(e_1)\, \mathbb{P}_{\mathcal{A}}(\pi(s_t, e_2, \dots, e_n))\, \mathrm{d}\mu^{\mathcal{A}}_s(t) \\
&= \frac{1}{2} \int_{t \in I(s,e_1)} p^{\mathcal{A}}_{s+t}(e_1) \sum_{\pi_t \in \iota(\pi(s_t, e_2, \dots, e_n))} \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi_t)\, \mathrm{d}\mu^{\mathcal{A}}_s(t) \qquad \text{by induction hypothesis} \\
&= \frac{1}{2} \sum_{q} \int_{\substack{t \in I(s,e_1) \\ s+t \in q}} p^{\mathcal{A}}_{s+t}(e_1) \sum_{\pi_t \in \iota(\pi(s_t, e_2, \dots, e_n))} \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi_t)\, \mathrm{d}\mu^{\mathcal{A}}_s(t) \\
&= \frac{1}{2} \sum_{q} \int_{\substack{t \in I(s,e_1) \\ s+t \in q}} p^{\mathcal{A}}_{s+t}(e_1) \sum_{(f_2, \dots, f_n) \in \iota(e_1(q), e_2, \dots, e_n)} \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi(\iota(s_t), f_2, \dots, f_n))\, \mathrm{d}\mu^{\mathcal{A}}_s(t) \\
&= \frac{1}{2} \sum_{q} \int_{\substack{t \in I(\iota(s), f_1) \\ s+t \in q \\ q \models f_1 \\ [s] \xrightarrow{f_1} e_1(q)}} p^{\mathsf{R}(\mathcal{A})}_{\iota(s)+t}(f_1) \sum_{(f_2, \dots, f_n) \in \iota(e_1(q), e_2, \dots, e_n)} \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi(\iota(s)_t, f_2, \dots, f_n))\, \mathrm{d}\mu^{\mathsf{R}(\mathcal{A})}_{\iota(s)}(t) \\
&\qquad \text{by hypothesis on the measures} \\
&= \frac{1}{2} \sum_{\substack{q \models f_1 \\ [s] \xrightarrow{f_1} e_1(q) \\ (f_2, \dots, f_n) \in \iota(e_1(q), e_2, \dots, e_n)}} \int_{t \in I(\iota(s), f_1)} p^{\mathsf{R}(\mathcal{A})}_{\iota(s)+t}(f_1)\, \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi(\iota(s)_t, f_2, \dots, f_n))\, \mathrm{d}\mu^{\mathsf{R}(\mathcal{A})}_{\iota(s)}(t) \\
&= \sum_{\substack{q \models f_1 \\ [s] \xrightarrow{f_1} e_1(q) \\ (f_2, \dots, f_n) \in \iota(e_1(q), e_2, \dots, e_n)}} \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi(\iota(s), f_1, \dots, f_n)) \\
&= \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\iota(\pi))
\end{aligned}$$

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

3.2 Probabilistic Semantics

We consider the logic LTL [Pnu77], defined inductively as:

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

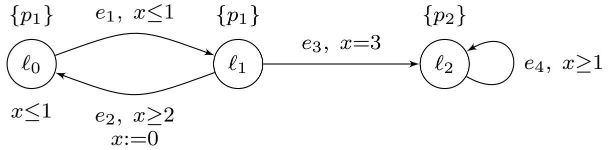
where $p \in \mathsf{AP}$ is an atomic proposition. We use classical shorthands like $\mathtt{tt} \overset{\text{def}}{=} p \vee \neg p$, $\mathtt{ff} \overset{\text{def}}{=} p \wedge \neg p$, $\varphi_1 \Rightarrow \varphi_2 \overset{\text{def}}{=} \neg \varphi_1 \vee \varphi_2$, $\mathbf{F}\, \varphi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

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 $\mathcal{A}, s_0 \mathrel{|}\!\!\approx_{\mathbb{P}} \varphi$, 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.*, $\psi \overset{\text{def}}{=} \mathbf{F}\,\mathbf{G}\,\mathsf{acc}$. 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 $\mathcal{A}, s_0 \mathrel{|}\!\!\approx_{\mathbb{P}} \mathbf{G}\, p_1$. 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,*

$$\mathcal{A}, s \mathrel{|}\!\!\approx_{\mathbb{P}} \varphi \;\Leftrightarrow\; \mathsf{R}(\mathcal{A}), \iota(s) \mathrel{|}\!\!\approx_{\mathbb{P}} \varphi\,.$$

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 [Mun00]. 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 [VV06] 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.

Example 11. 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 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.

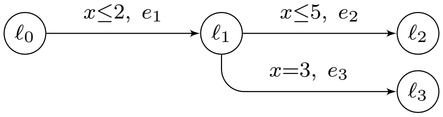
Banach-Mazur games are not always determined, even for simple topological spaces (see [Oxt57, Remark 1]). Still a natural question is to know when the players have winning strategies. The following result gives a partial answer:

Theorem 12 (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.*

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 13. 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 $\pi(s_0, e_1 e_2)$.



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

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

$\mathsf{Pol}(\pi)$ 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 $\pi' = \pi(s_0, e_1 e_3)$. 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 14. *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 15. 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 $\pi(s_0, (e_1 e_2)^*)$ or of the form $\pi(s_0, e_1(e_2 e_1)^*)$. A (constrained) basic open set is then for instance $\pi_{\mathcal{C}}(s_0, e_1 e_2)$ 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 $\pi(s_0, (e_1 e_2)^* e_3 e_4^*)$ is meager.

Remark 16. Rather 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 converge to the empty set).

A topological space in which all non-empty open sets are not meager is called a *Baire space*.[10] As noticed in the previous remark, not all topological spaces are Baire spaces. However, non-Baire spaces have strange properties. For instance, in the set $\mathbb{Q}$ of rationals, all sets are both meager and large.

Proposition 17. *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.*

Before proving this proposition, we first prove that basic open sets really form a basis for $\mathcal{T}_\mathcal{A}$.

Lemma 18. *Let $\mathcal{A}$ be a timed automaton, and s a state of $\mathcal{A}$. The basic open sets and $\mathsf{Runs}(\mathcal{A}, s)$ form a basis for the topological space $(\mathsf{Runs}(\mathcal{A}, s), \mathcal{T}_\mathcal{A})$.*

We postpone the proof of this technical lemma in the appendix, on page ii.

Proof (of Proposition 17). To prove that $(\mathsf{Runs}(\mathcal{A}, s), \mathcal{T}_\mathcal{A})$ is a Baire space, we prove that every non-empty basic open set in $\mathcal{T}_\mathcal{A}$ is not meager. This is sufficient since the basic open sets form a basis of the topological space, see Lemma 18. Let $\pi_\mathcal{C}$ be a basic open set. Using Theorem 12, we prove that $\pi_\mathcal{C}$ is not meager by proving that Player 2 does not have a winning strategy for the Banach-Mazur game where $C = \pi_\mathcal{C}$, and $\mathcal{B}$ is the set of basic open sets (once more, this is legal to play with the basic open sets, because they form a basis of the topological space, see Lemma 18).

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 her moves 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 $\pi(s, e_1 \ldots e_n)$, 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.

By iterating the same process for the strategy of Player 1, we obtain the following sequence:

$$\pi_\mathcal{C} = \pi_1 \supseteq \pi_2 \supseteq F_2 \supseteq \pi_3 \supseteq \pi_4 \supseteq F_3 \supseteq \pi_5 \supseteq \ldots \supseteq F_i \supseteq \pi_{2i-1} \supseteq \pi_{2i} \supseteq \cdots$$

We have that

$$\bigcap_{i=1}^{\infty} \pi_i = \bigcap_{i=1}^{\infty} F_i \, .$$

By compactness of F_2, we can ensure that $\bigcap_{i=1}^{\infty} F_i$ is non-empty (Heine-Borel theorem), implying that $\bigcap_{i=1}^{\infty} \pi_i$ is non-empty. Player 2 has thus no winning strategy, the basic open set $\pi_\mathcal{C}$ is thus non-meager, which concludes the proof. □

[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, the two definitions coincide, see [Mun00, p.295].

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 $\mathcal{A}, s \mathrel{|\!\!\approx_{\mathcal{T}}} \varphi$, 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 19. *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 5). Then ι is continuous, and for every non-empty open set $\mathcal{O} \in \mathcal{T}_{\mathcal{A}}$, $\iota(\overset{\circ}{\mathcal{O}}) \neq \emptyset$.*

Proof. We first prove that ι is continuous. Let $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(\iota(s), f_1 \dots f_n)$ be a basic open set of $(\mathsf{Runs}(\iota(s), \mathsf{R}(\mathcal{A})), \mathcal{T}_{\mathsf{R}(\mathcal{A})})$. We will prove that $\pi' = \iota^{-1}(\pi_{\mathcal{C}})$ is a basic open set of $(\mathsf{Runs}(s, \mathcal{A}), \mathcal{T}_{\mathcal{A}})$. First notice that $\mathsf{Pol}(\pi_{\mathcal{C}}) = \mathsf{Pol}(\pi')$. Let γ be the tightest constraint corresponding to $\pi(\iota(s), f_1 \dots f_n)$. Then, as $\pi_{\mathcal{C}}$ has a defined dimension (and $\mathsf{R}(\mathcal{A})$ is the region automaton), we can prove by induction on the length of the path, that there exists an open constraint $\widetilde{\gamma}$ such that $\mathsf{Pol}(\pi_{\mathcal{C}}) = \mathsf{Pol}(\pi_{\mathcal{C} \wedge \widetilde{\gamma}}(s, e_1 \dots e_n))$. Moreover, there exists some open set $\mathcal{O}$ in $\mathbb{R}^n$ such that $\mathcal{O} \cap \mathsf{Pol}(\pi(\iota(s), f_1 \dots f_n)) = \mathsf{Pol}(\pi_{\mathcal{C}})$. Then, we get that $\mathsf{Pol}(\pi') = \mathsf{Pol}(\pi(s, e_1 \dots e_n)) \cap \mathcal{O} \cap \mathsf{Pol}(\widetilde{\gamma})$. The set $\mathcal{O} \cap \mathsf{Pol}(\widetilde{\gamma})$ is an open set of $\mathbb{R}^n$, hence $\mathsf{Pol}(\pi')$ is an open set of $\pi(s, e_1 \dots e_n)$ for the induced topology. Moreover, $\pi' = \pi_{\mathcal{C}'}(s, e_1 \dots e_n)$ (taking for instance $\mathcal{C}' = \mathcal{C} \wedge \widetilde{\gamma}$), we write it $\pi_{\mathcal{C}'}$ for now.

It remains to prove that $\pi_{\mathcal{C}'}$ has defined dimension in $\mathcal{A}$. Assume it is not the case. Then there exists some $1 \leq i \leq n$ such that

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

By property of the region automaton, we get that

$$\dim\Big(\mathsf{Pol}(\pi_{\mathcal{C}_i}\Big) < \dim\Big(\bigcup_f \mathsf{Pol}(\pi_{\mathcal{C}_{i-1}}(\iota(s), f_1 \dots, f_{i-1} f))\Big).$$

This contradicts the assumption that π has defined dimension in $\mathsf{R}(\mathcal{A})$. We conclude that π' is an open set of $(\mathsf{Runs}(s, \mathcal{A}), \mathcal{T}_{\mathcal{A}})$.

We now prove that for every non-empty open set $\mathcal{O} \in \mathcal{T}_{\mathcal{A}}$, $\iota(\overset{\circ}{\mathcal{O}}) \neq \emptyset$. Let $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1 \dots e_n)$ be a basic open set of $(\mathsf{Runs}(s, \mathcal{A}), \mathcal{T}_{\mathcal{A}})$. We have that

$$\iota(\pi_{\mathcal{C}}) = \bigcup_{f_1, \cdots f_n} \pi_{\mathcal{C}}(\iota(s), f_1 \dots f_n)$$

where the (finite) union is taken over all sequences of edges $f_1, \dots, f_n$ corresponding to $e_1, \dots, e_n$. There exists thus some $f_1, \dots, f_n$ such that

$$\dim(\mathsf{Pol}(\pi_{\mathcal{C}})) = \dim\Big(\mathsf{Pol}\big(\pi_{\mathcal{C}}(\iota(s), f_1 \dots f_n)\big)\Big)$$

and we write $\pi'_{\mathcal{C}} = \pi_{\mathcal{C}}(\iota(s), f_1 \dots f_n)$. We will prove that $\pi'_{\mathcal{C}}$ is an open set. Note that $\mathcal{C}$ characterizes an open set of $\mathbb{R}^n$, hence $\pi'_{\mathcal{C}}$ is open in $\pi(\iota(s), f_1 \dots f_n)$. Assume that it has an undefined dimension. Then, there exists some i such that

$$\dim\Big(\mathsf{Pol}\big(\pi_{\widetilde{\mathcal{C}}_i}(\iota(s), f_1 \dots f_i)\big)\Big) < \dim\Big(\bigcup_f \mathsf{Pol}\big(\pi_{\widetilde{\mathcal{C}}_{i-1}}(\iota(s), f_1 \dots f_{i-1}, f)\big)\Big)$$

where $\widetilde{\mathcal{C}}_i$ corresponds to the projection on the i first components of the tightest constraint defining $\pi'_{\mathcal{C}}$. Moreover, as $\mathsf{Pol}(\pi'_{\mathcal{C}}) \subseteq \mathsf{Pol}\big(\pi_{\mathcal{C}}\big)$ and $\dim(\mathsf{Pol}(\pi'_{\mathcal{C}})) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}}))$, applying Lemma B, we get that for all i's, $\dim(\mathsf{Pol}(\pi'_{\widetilde{\mathcal{C}}_i})) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}_i}))$. Furthermore, $\bigcup_f \mathsf{Pol}\big(\pi_{\widetilde{\mathcal{C}}_{i-1}}(\iota(s), f_1 \dots f_{i-1} f)\big) \subseteq \bigcup_e \mathsf{Pol}\big(\pi_{\mathcal{C}_{i-1}}(s, e_1 \dots e_{i-1} e)\big)$ (this is a property of the region automaton). Finally, we get that

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

which contradicts the hypothesis that π has defined dimension. We deduce that π' is an open set of $(\mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s)), \mathcal{T}_{\mathsf{R}(\mathcal{A})})$, hence the result. □

Corollary 20. *Let $\mathcal{A}$ be a timed automaton, s a state of $\mathcal{A}$, and φ an* LTL *formula. Then,*

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

Proof. We prove both implications using characterisation of meager sets by Banach-Mazur games.

Assume Player 2 has a winning strategy in $\mathcal{A}$ to avoid $[\![\varphi]\!]_{\mathcal{A}}$. We will show that Player 2 also has a winning strategy in $\mathsf{R}(\mathcal{A})$ to avoid $[\![\varphi]\!]_{\mathsf{R}(\mathcal{A})}$. The first move of Player 1 is some path $\pi_{\gamma_1}(s, f_1 \dots f_n)$, that can be transported in $\mathcal{A}$ (thanks to Lemma 19, this is a legal move of the game on $\mathcal{A}$). Then, Player 2 can play his own strategy, *etc.* All moves are legal, thanks to Lemma 19. Finally, the intersection of all moves is equal to the one in $\mathcal{A}$, hence it does not intersect $[\![\varphi]\!]_{\mathsf{R}(\mathcal{A})}$ (because, roughly, up to ι, the same runs are in $[\![\varphi]\!]_{\mathsf{R}(\mathcal{A})}$ and in $[\![\varphi]\!]_{\mathcal{A}}$).

On the contrary, assume that Player 2 has a winning strategy in $\mathsf{R}(\mathcal{A})$ to avoid $[\![\varphi]\!]_{\mathsf{R}(\mathcal{A})}$. We will show that Player 2 also has a winning strategy in $\mathcal{A}$ to avoid $[\![\varphi]\!]_{\mathcal{A}}$. Assume that Player 1 plays $\pi_\gamma(s, e_1 \dots e_n)$, then applying Lemma 19, Player 2 can play as if it was $\pi_\gamma(\iota(s), f_1 \dots f_n)$ in $\mathsf{R}(\mathcal{A})$ for some $f_1, \dots, f_n$. The game then plays as in $\mathsf{R}(\mathcal{A})$, and all moves are legal thanks to Lemma 19. □

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 21. *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]

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

Proof. Let $\pi = \pi(s, e_1 \dots e_n)$ be an unconstrained symbolic path in $\mathsf{R}(\mathcal{A})$.

We first prove that $\mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi) > 0$ implies $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \top$. Towards a contradiction, assume that $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \bot$. Following Corollary G in the appendix, there must exist some index $1 \leq i \leq n$ such that $\mu_{s'}(I(s', e_{i+1})) = 0$ for every $s \xrightarrow{e_1} \cdots \xrightarrow{e_i} s'$. Hence, $\mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi(s', e_{i+1} \dots e_n)) = 0$ by definition of the probability. Thus, it holds that $\mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi(s, e_1 \dots e_n)) = 0$.

Assume now $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \top$. Corollary G in the appendix implies that for any index $i \leq n$, any state s_i reachable from s via $e_1, \dots, e_{i-1}$, satisfies $\mu_{s_i}(I(s_i, e_i)) > 0$. We then use the definition of the probability inductively on suffixes of π starting in some s_i to obtain a sequence of integral computation over non negligible set of a positive function, hence $\mathbb{P}_{\mathsf{R}(\mathcal{A})}(\pi) > 0$. □

The main result of this paper is the following theorem.

Theorem 22. *Let $\mathcal{A}$ be a non-blocking timed automaton, s a state of $\mathcal{A}$, and φ an* LTL *formula. Then,*

$$\mathcal{A}, s \mathrel{|\!\!\approx}_{\mathbb{P}} \varphi \;\Leftrightarrow\; \mathcal{A}, s \mathrel{|\!\!\approx}_{\mathcal{T}} \varphi\,.$$

Proof. Thanks to Corollary 10 and Corollary 20, it is equivalent (and hence sufficient) to prove that $\mathsf{R}(\mathcal{A}), \iota(s) \mathrel{|\!\!\approx}_{\mathcal{T}} \varphi$ iff $\mathsf{R}(\mathcal{A}), \iota(s) \mathrel{|\!\!\approx}_{\mathbb{P}} \varphi$. Moreover, $\mathsf{R}(\mathcal{A}), \iota(s) \mathrel{|\!\!\approx}_{\mathbb{P}} \varphi$ iff $\mathbb{P}_{\mathsf{R}(\mathcal{A})}(\iota(s), \neg\varphi) = 0$, thus applying Proposition 21, $\mathsf{R}(\mathcal{A}), \iota(s) \mathrel{|\!\!\approx}_{\mathbb{P}} \varphi$ 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 $\mathsf{R}(\mathcal{A}), \iota(s) \mathrel{|\!\!\approx}_{\mathcal{T}} \varphi$, *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.

To prove the first implication, we use a Banach-Mazur game and Theorem 12, playing with the set $\mathcal{B}$ of basic open sets together with the $\mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s))$ (which form a basis of the topology, as already said in the proof of Proposition 17). The objective of the game is set to be $[\![\neg\varphi]\!]$. By hypothesis and Proposition 21, $[\![\neg\varphi]\!] \subseteq \cup_{\dim_{\mathsf{R}(\mathcal{A})}(\pi)=\bot} \pi$. As every basic open set has defined dimension it holds for every $B \in \mathcal{B}$ such that $B \neq \mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s))$, that $B \cap [\![\neg\varphi]\!] = \emptyset$. Thus, if the first play of Player 1 is $\mathsf{Runs}(\mathsf{R}(\mathcal{A}), \iota(s))$, Player 2 picks some path of defined dimension. If the first play of Player 1 is a path π, then Player 2 just chooses the same path. Then, Player 2 wins the game by mimicking at each round the choices of Player 1, *i.e.*, whatever set $B_{2\cdot j-1}$ Player 1 chooses in the j-th round, Player 2 answers with the same choice $B_{2\cdot j} = B_{2\cdot j-1}$. For such a play we clearly have $\bigcap_{i=1}^{\infty} B_i \subset [\![\varphi]\!]$, hence $\bigcap_{i=1}^{\infty} B_i \cap [\![\neg\varphi]\!] = \emptyset$, and Player 2 has a winning strategy for the game. Theorem 12 implies that $[\![\neg\varphi]\!]$ is meager.

Let us now prove the other implication. For a contradiction we assume that there exists a symbolic path π in $\mathsf{R}(\mathcal{A})$ with defined dimension which does not satisfy φ. In particular $\{\varrho \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), s_0) \mid \varrho \not\models \varphi\}$ contains an open set, which is not meager by Proposition 17 (($\mathsf{Runs}(\mathcal{A}, s_0), \mathcal{T}_{\mathcal{A}}$) is a Baire space). Since the notion of being meager is closed under subset, the set $\{\varrho \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), s_0) \mid \varrho \not\models \varphi\}$ is not meager. Hence the set $\{\varrho \in \mathsf{Runs}(\mathsf{R}(\mathcal{A})) \mid \varrho \models \varphi\}$ is not large which is a contradiction. □

Remark 23. 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 24. *Over finite timed words, the almost-sure and the large* LTL *model-checking problems over non-blocking timed automata are* PSPACE*-Complete.*

Proof. The two problems are equivalent, due to Theorem 22. 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 5), and we color it in blue otherwise.

Lemma 25. *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.*

Proof. The proof is a consequence of Lemma F and can be done by induction on the length n of π.

Case $n = 0$ is obvious since $\dim_{\mathsf{R}(\mathcal{A})}(\pi(s)) = \top$ and $\pi(s)$ surely contains no red edge.

Assume the induction hypothesis holds for any $i \leq n-1$, and let $\pi = \pi(s, e_1 \dots e_n)$ be a path of length n. Let us first consider the case $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \bot$. If π has a prefix $\pi(s, e_1 \dots e_i)$ of undefined dimension, then some edge in $\{e_1, \dots, e_i\}$ is red by induction hypothesis. Otherwise, thanks to Lemma F, for all configuration s' such that $s \xrightarrow{e_1} \cdots \xrightarrow{e_{n-1}} s'$, $\mu_{s'}(I(s', e_n)) = 0$. By definition, edge e_n is thus a red edge. Consider now that π contains a red edge. If some edge in $\{e_1, \dots, e_{n-1}\}$ is red, by induction hypothesis, some prefix of π has undefined dimension. Since any continuation of a path with undefined dimension has undefined dimension, $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \bot$. Assume now e_n is a red edge, *i.e.*, $\mu_{s'}(I(s', e_n)) = 0$ for all configurations s' with $s \xrightarrow{e_1} \cdots \xrightarrow{e_{n-1}} s'$. By Lemma F, we conclude $\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \bot$. □

Now, applying Proposition 21, to decide whether $\mathcal{A} \not\approx_{\mathbb{P}} \varphi$, 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. Using what precedes, it is thus sufficient to find a counter-example for φ in $\mathsf{R}(\mathcal{A})$ restricted to blue edges. Everything can be guessed non-deterministically in PSPACE. Indeed, there is no need to construct *a priori* the whole graph $\mathsf{R}(\mathcal{A})$, a counter-example can be guessed on-the-fly. Moreover, to guess such a path, we only need to store two consecutive locations of $\mathsf{R}(\mathcal{A})$ 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 $\mathsf{R}(\mathcal{A})$ 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 =PSPACE.

□

In the literature, several pieces of work can be found on real-time and probabilistic systems. We report here the most closely related models, and explain the differences with our approach. Moreover, our probabilistic semantics can somehow be viewed as a notion of *robustness* for timed automata. We also review here work in that direction.

Work on Probabilistic Timed Systems

Probabilistic timed automata. The model of probabilistic timed automata has been extensively studied by the group developing the PRISM tool [KNP04]. Among others, a possible reference for that model is [KNSS02]. Roughly, in this model, no probability is put on delays, but probabilistic and non-deterministic choices are put on discrete transitions. This model is thus somehow orthogonal to our model, but we could as well have a probabilistic choice between discrete transitions (right now, we assume the choice between transitions is uniform). For the model of probabilistic timed automata, the model-cheking problem for TCTL specifications, a timed extension of CTL, is decidable, reducing to the model-checking problem of CTL for (untimed) Markov decision processes.

Continuous-time Markov chains. 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 at each step.

Real-time probabilistic processes. The model of real-time probabilistic processes has been defined in the early 90's in [ACD91,ACD92], and this is probably the model which is the closest to ours. This model gathers a number of independent processes with a single clock, and when a process is launched, its duration is probabilistically chosen in the set of all its possible durations. However, all processes are independent, and a process cannot have a higher priority w.r.t. to another one. In particular, it means that the choice of the transition to be taken is made before choosing probabilistically a delay. As a consequence, even transitions with very small firing intervals can have a high probability to be taken, even though events with much larger firing intervals are possible. This model satisfies intrinsically different properties than ours, and the authors of the above-mentioned papers solve the qualitative model-checking problem for TCTL, a timed extension of CTL over infinite timed words. More recently this model has been used for quantifying test cases [JPQ05].

Work on the Robustness of Timed Systems

Topological acceptance. Let us compare our notion of dimension and the associated topology with the notion of robust timed automata introduced in [GHJ97] and further studied in [HR00]. In [GHJ97], several metrics on finite trajectories[12] of timed automata have been introduced and proved equivalent. These different metrics induce a topology on the set of finite runs (of a given timed automaton). The basic open sets of these topologies are called tubes. From this topology, the authors define the notion of tube acceptance to obtain a *robust semantics* for timed automata: under that semantics, a trajectory ω is (robustly) accepted whenever there is a tube O such that $\omega \in O$, and a dense subset of O is classically accepted by the timed automaton.

We would like to compare the notion of robustness introduced in [GHJ97] with our topological semantics. 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*.

The timed automaton $\mathcal{A}_1$ of Figure 1 accepts robustly no trajectory. Indeed take any trajectory ω of the form $(a, \tau_1)(a, \tau_2)$, clearly ω is accepted by $\mathcal{A}_1$ if and only if $\tau_2 - \tau_1 = 1$, it is thus impossible to find an open set of accepting trajectories (for any reasonable metric). However the (symbolic) path π starting from $(\ell_0, 0)$ and ending in ℓ_2 has a defined dimension. In this example, the difference between the two notions heavily relies on the fact that the notion of "robustness" is based on a topology on $(\Sigma \times \mathbb{R}^+)^*$ and not on a topology induced by the trajectories of the timed automaton.

Let us now consider the timed automaton of Figure 2, where ℓ_0 is the initial location and the three other locations are accepting. The trajectory $(a, 1)$ is robustly accepted in the sense of [GHJ97], since given $\varepsilon > 0$ any trajectory (a, τ) with $|1 - \tau| < \varepsilon$ is also accepted. However the symbolic path π starting from $(\ell_0, 0)$ and determined by the transition from ℓ_0 to ℓ_2 (which consists in the unique run corresponding to the trajectory $(a, 1)$) has no defined dimension.

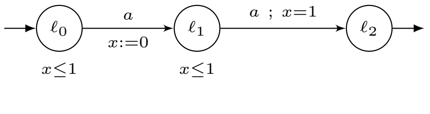


Fig. 1. Timed automaton $\mathcal{A}_1$

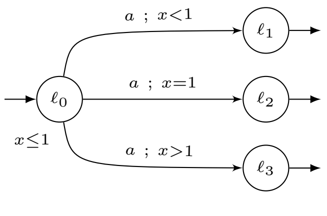


Fig. 2. Timed automaton $\mathcal{A}_2$

[12] A finite trajectory over an alphabet Σ is an element of $(\Sigma \times \mathbb{R}^+)^*$.

Other notions of robustness. Other notions of robustness have been studied, where the model of timed automata is slightly modified in that clocks can drift or guards are somehow enlarged [Pur98,DDR04,DDMR04,ALM05,BMR06]. Hence, such a model accepts more behaviours than the original automaton, and there is absolutely no link between these "enlarged" (or perturbed, following [ALM05]) models and our probabilistic semantics. In their framework, various results have been proved, like the decidability of safety properties [Pur98,DDMR04], the decidability of LTL model checking over infinite words [BMR06], the determinisability of one-clock perturbed timed automata [ALM05].

7 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 $\mathsf{CTL}^\star$ or in the μ-Calculus). It could also be applied to various classes of hybrid systems with a finite bisimulation quotient [HMR05].

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

[ACD91] Rajeev Alur, Costas Courcoubetis, and David Dill. Model-checking for probabilistic real-time systems. In *Proceedings of the 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 *Proceedings of the Workshop on Real-Time: Theory in Practice, 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 *Proceedings of the 8th International Workshop on Hybrid Systems: Computation and Control (HSCC'05)*, volume 3414 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 2005.

[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 *Proceedings of the 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 *Proceedings of the 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 *Proceedings of the 7th International Workshop on Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *Lecture Notes in Computer Science*, pages 296–310. Springer, 2004.

[GHJ97] Vineet Gupta, Thomas A. Henzinger, and Radha Jagadeesan. Robust timed automata. In *Proceedings of the 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 *Proceedings of the 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 *Proceedings of the 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 *Proceedings of the 1st International Conference on the Quantitative Evaluation of Systems (QEST'04)*, pages 322–323. IEEE Computer Society Press, 2004.

[KNSS02] Marta 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.

[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.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 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 *Proceedings of the 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.

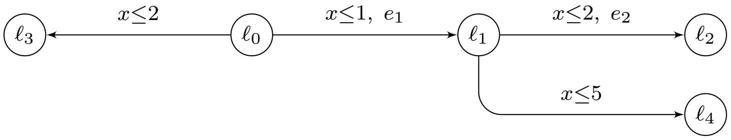
[Spr04] Jeremy Sproston. Model checking for probabilistic timed systems. In *Validation of Stochastic Systems – A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, pages 189–229. Springer, 2004.

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

In this appendix, we present some technical results omitted in the core of the paper, and some tedious computations.

Complements for Section 3

Example of probability computation (Example 5 page 6). In this example, we assume that the probability distributions over delays and enabled edges are uniform.

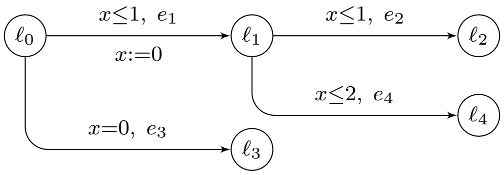


We can then compute:

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

because $\mu_{(\ell_0,0)}=\frac{\lambda}{2}$ (resp. $\mu_{(\ell_1,t)}=\frac{\lambda}{5-t}$) is the uniform distribution over $[0,2]$ (resp. $[t,5]$).

Another example of probability computation. Consider the following timed automaton, and assume uniform probability distributions over both delays and enabled edges:



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

$$\begin{aligned}
\mathbb{P}(\pi((\ell_0,0),e_1,e_2)) \quad = & \\
& \frac{1}{2}\left(\int_0^0 \frac{\mathbb{P}(\pi((\ell_1,0),e_2))}{2}\mathrm{d}\mu_{(\ell_0,0)}(t) + \int_0^1 \mathbb{P}(\pi((\ell_1,0),e_2))\mathrm{d}\mu_{(\ell_0,0)}(t)\right) \\
= & \frac{1}{2}\left(\int_0^1 \left(\frac{1}{2}\int_0^1 \frac{\mathbb{P}(\pi((\ell_2,u)))}{2}\mathrm{d}\mu_{(\ell_1,0)}(u)\right)\mathrm{d}\mu_{(\ell_0,0)}(t)\right) \\
= & \frac{1}{2}\left(\int_0^1 \left(\frac{1}{2}\int_0^1 \frac{\mathbb{P}(\pi((\ell_2,u)))}{2}\frac{1}{2}\mathrm{d}\lambda(u)\right)\mathrm{d}\lambda(t)\right) \\
= & \frac{1}{16}\left(\int_0^1 \left(\int_0^1 \frac{1}{2}\mathrm{d}\lambda(u)\right)\mathrm{d}\lambda(t)\right) = \frac{1}{32}
\end{aligned}$$

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

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

Proof of Lemma 18 (Section 4)

Lemma 18. *Let $\mathcal{A}$ be a timed automaton, and s a state of $\mathcal{A}$. The basic open sets and $\mathsf{Runs}(\mathcal{A}, s)$ form a basis for the topological space $(\mathsf{Runs}(\mathcal{A}, s), \mathcal{T}_{\mathcal{A}})$.*

Proof. To prove this lemma, it is sufficient to prove that the intersection of two basic open sets is still a basic open set.

Lemma A *Let $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}(s, e_1, \ldots, e_n)$ and $\pi_{\mathcal{C}'} = \pi_{\mathcal{C}'}(s, e_1, \ldots, e_n)$ be two basic open sets. Then $\pi_{\mathcal{C}} \cap \pi_{\mathcal{C}'}$ is a basic open set.*

Proof. Let us denote in this proof $\mathcal{C}'' = \mathcal{C} \cap \mathcal{C}'$, and π the unconstrained symbolic path $\pi(s, e_1, \ldots, e_n)$. Write $\pi_{\mathcal{C}''}$ for $\pi_{\mathcal{C}} \cap \pi_{\mathcal{C}'} = \pi_{\mathcal{C}''}(s, e_1, \ldots, e_n)$.

We first show that $\mathsf{Pol}(\pi_{\mathcal{C}} \cap \pi_{\mathcal{C}'})$ is open in $\mathsf{Pol}(\pi)$, which is the second condition for $\pi_{\mathcal{C}} \cap \pi_{\mathcal{C}'}$ to be an open set. We have that $\mathsf{Pol}(\pi_{\mathcal{C}''}) = \mathsf{Pol}(\pi_{\mathcal{C}}) \cap \mathsf{Pol}(\pi_{\mathcal{C}'})$. By assumption both $\mathsf{Pol}(\pi_{\mathcal{C}})$ and $\mathsf{Pol}(\pi_{\mathcal{C}'})$ are open in $\mathsf{Pol}(\pi(s, e_1, \ldots, e_n))$, hence their intersection too.

We now come to the proof that $\dim_{\mathcal{A}}(\pi_{\mathcal{C}''}) = \top$. This relies on Lemma B.

Lemma B *Let $\pi_{\mathcal{C}}$ and $\pi_{\mathcal{C}'}$ be constrained symbolic paths such that $\pi_{\mathcal{C}} \subseteq \pi_{\mathcal{C}'}$ and $\dim(\mathsf{Pol}(\pi_{\mathcal{C}})) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}'}))$. Then for all $i \leq n$ (where n is the length of both paths), $\dim(\mathsf{Pol}(\pi_{\mathcal{C}_i})) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}'_i}))$*

Proof. Assume there exists an index $i \leq n$ such that $\dim(\mathsf{Pol}(\pi_{\mathcal{C}_i})) < \dim(\mathsf{Pol}(\pi_{\mathcal{C}'_i}))$. As $\dim(\mathsf{Pol}(\pi_{\mathcal{C}})) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}'}))$ there must be an index j, such that $\mathsf{Pol}(\pi_{\mathcal{C}})$ gains some dimension in the jth direction, whereas $\mathsf{Pol}(\pi_{\mathcal{C}'})$ does not. But this is not possible since $\pi_{\mathcal{C}} \subseteq \pi_{\mathcal{C}'}$ and therefore $\mathsf{Pol}(\pi_{\mathcal{C}}) \subseteq \mathsf{Pol}(\pi_{\mathcal{C}'})$ □

Corollary C *If $\mathsf{Pol}(\pi_{\mathcal{C}})$ is open in $\mathsf{Pol}(\pi)$, then for all $i \leq n$ (where n is the length of both paths), $\dim(\mathsf{Pol}(\pi_{\mathcal{C}_i})) = \dim(\mathsf{Pol}(\pi_i))$.*

Proof. As $\mathsf{Pol}(\pi_{\mathcal{C}})$ is open in $\mathsf{Pol}(\pi)$, $\dim(\mathsf{Pol}(\pi_{\mathcal{C}})) = \dim(\mathsf{Pol}(\pi))$. Applying Lemma B to $\pi_{\mathcal{C}} = \pi_{\mathcal{C}}$ and $\pi_{\mathcal{C}'} = \pi$ yields the expected result. □

Corollary D *If $\pi_{\mathcal{C}}$ is a non-empty open set of $(\mathit{Runs}(\mathcal{A}, s), \mathcal{T}_{\mathcal{A}})$, if $\pi_{\mathcal{C}'} \subseteq \pi_{\mathcal{C}}$ and $\mathsf{Pol}(\pi_{\mathcal{C}'})$ open in $\mathsf{Pol}(\pi)$, then $\dim_{\mathcal{A}}(\pi_{\mathcal{C}'})$ is defined (or, $\pi_{\mathcal{C}'}$ is a non-empty open set of $(\mathit{Runs}(\mathcal{A}, s), \mathcal{T}_{\mathcal{A}})$).*

Proof. Let $\pi_{\mathcal{C}'}(s, e_1 \ldots, e_n)$ be a constrained symbolic path that is contained in the basic open set $\pi_{\mathcal{C}}(s, e_1 \ldots, e_n)$, such that $\mathsf{Pol}(\pi_{\mathcal{C}'})$ is open. By Corollary C we get $\dim(\mathsf{Pol}(\pi_{\mathcal{C}'_i}(s, e_1, \ldots, e_i))) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}_i}(s, e_1, \ldots, e_i))) = \dim(\mathsf{Pol}(\pi(s, e_1, \ldots, e_i)))$. As $\pi_{\mathcal{C}}$ is a basic open set it holds that for $1 \leq i \leq n$: $\dim(\mathsf{Pol}(\pi_{\mathcal{C}'_i}(s, e_1, \ldots, e_i))) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}_i}(s, e_1, \ldots, e_i))) = \dim(\bigcup_e \mathsf{Pol}(\pi_{\mathcal{C}_{i-1}}(s, e_1, \ldots, e_{i-1}, e)))$. The last expression is greater than or equal to $\dim(\bigcup_e \mathsf{Pol}(\pi_{\mathcal{C}'_{i-1}}(s, e_1, \ldots, e_{i-1}, e)))$ which shows that $\pi_{\mathcal{C}'}$ has a defined dimension. □

We come back to the proof of Lemma A. Since $\mathsf{Pol}(\pi_{\mathcal{C}})$ and $\mathsf{Pol}(\pi_{\mathcal{C}''})$ are both convex and open in $\mathsf{Pol}(\pi)$, and because they intersect non trivially, $\dim(\mathsf{Pol}(\pi_{\mathcal{C}})) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}''}))$[13]. We now use Lemma B to obtain that for every $i \leq n$, $\dim(\mathsf{Pol}(\pi_{\mathcal{C}_i})) = \dim(\mathsf{Pol}(\pi_{\mathcal{C}''_i}))$. Hence, for every $i \leq n$,

[13] We use here the following general topology result: if X is a convex set and O an open set in $\mathbb{R}^n$ such that $X \cap O \neq \emptyset$, then $\dim(X) = \dim(X \cap O)$.

$$\begin{aligned}\dim(\mathsf{Pol}(\pi_{\mathcal{C}''{}_i})) &= \dim(\mathsf{Pol}(\pi_{\mathcal{C}_i})) \\ &= \dim(\bigcup_e \mathsf{Pol}(\pi_{\mathcal{C}_{i-1}}(s, e_1, \ldots, e_{i-1}, e))) \qquad \text{since } \dim(\pi_{\mathcal{C}}) = \top \\ &= \dim(\bigcup_e \mathsf{Pol}(\pi_{\mathcal{C}''{}_{i-1}}(s, e_1, \ldots, e_{i-1}, e)))\end{aligned}$$

The last equality holds because, if we write π_i for the projection of π over the first i components, the following arguments hold:

- $\dim(\mathsf{Pol}(\pi_i)) = \dim\Big(\mathsf{Pol}\big(\pi_{\mathcal{C}''_i}\big)\Big) \leq \dim\Big(\mathsf{Pol}\big(\pi_{\mathcal{C}''_{i-1}}(s, e_1, \ldots, e_{i-1}, e_i)\big)\Big) \leq \dim(\mathsf{Pol}(\pi_i))$ (the equality holds by Corollary C; and both inequalities hold by inclusion of polyhedra).
- Also $\dim\Big(\mathsf{Pol}(\pi_{\mathcal{C}''_i})\Big) \leq \dim\Big(\bigcup_e \mathsf{Pol}\big(\pi_{\mathcal{C}''_{i-1}}(s, e_1, \ldots, e_{i-1}, e)\big)\Big)$, taking e_i as a witness for e.
- For every e, the set $\mathsf{Pol}\big(\pi_{\mathcal{C}''_{i-1}}(s, e_1, \ldots, e_{i-1}, e)\big)$ is open in $\mathsf{Pol}\big(\pi_{i-1}(s, e_1, \ldots, e_{i-1}, e)\big)$, hence $\dim\Big(\mathsf{Pol}\big(\pi_{\mathcal{C}''_{i-1}}(s, e_1, \ldots, e_{i-1}, e)\big)\Big) = \dim\Big(\mathsf{Pol}\big(\pi_{i-1}(s, e_1, \ldots, e_{i-1}, e)\big)\Big)$. It follows that $\dim\Big(\bigcup_e \mathsf{Pol}\big(\pi_{\mathcal{C}''_{i-1}}(s, e_1, \ldots, e_{i-1}, e)\big)\Big) = \dim\Big(\bigcup_e \mathsf{Pol}\big(\pi_{i-1}(s, e_1, \ldots, e_{i-1}, e)\big)\Big)$.
- As the dimension of π is defined, $\dim\Big(\mathsf{Pol}(\pi_i)\Big) = \dim\Big(\bigcup_e \mathsf{Pol}\big(\pi_{i-1}(s, e_1, \ldots, e_{i-1}, e)\big)\Big)$.

Gathering everything, we get that $\dim_{\mathcal{A}}(\pi_{\mathcal{C}''}) = \top$.

We showed that $\mathsf{Pol}(\pi_{\mathcal{C}''})$ is open in $\mathsf{Pol}(\pi)$, and that and $\dim_{\mathcal{A}}(\pi_{\mathcal{C}''}) = \top$, thus $\pi_{\mathcal{C}''}$ is an open set for our topology. □

Complements for Section 5

Lemma E *Let $\mathcal{A}$ be a timed automaton and $\pi(s, e_1, \ldots, e_n)$ be a symbolic path of $\mathsf{R}(\mathcal{A})$. If $\mathsf{Pol}(\pi(s, e_1, \ldots, e_n)) \neq \emptyset$, then $\mathsf{Pol}(\pi(s, e_1, \ldots, e_i))$ is exactly the projection of $\mathsf{Pol}(\pi(s, e_1, \ldots, e_n))$ on the i first coordinates.*

Proof. In order to prove this lemma, we have to prove the following equality:

$$\begin{aligned}&\{(\tau_1, \ldots, \tau_i) \mid s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_i, e_i} s_i \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), s)\} \\ = &\{(\tau_1, \ldots, \tau_i) \mid \exists \tau_{i+1}, \ldots, \exists \tau_n \; s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_i, e_i} s_i \cdots \xrightarrow{\tau_n, e_n} s_n \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), s)\}.\end{aligned}$$

Let us first prove that the polyedron $\mathsf{Pol}(\pi(s, e_1, \ldots, e_i))$ is included in the projection of $\mathsf{Pol}(\pi(s, e_1, \ldots, e_n))$ on the i first coordinates. Let us first notice that since $\mathsf{Pol}(\pi(s, e_1, \ldots, e_n)) \neq \emptyset$, there exists $\tau_1, \ldots, \tau_n$ such that:

$$s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_i, e_i} s_i \cdots \xrightarrow{\tau_n, e_n} s_n \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), s)\,.$$

Let $(\tau'_1, \ldots, \tau'_i)$ be in $\mathsf{Pol}(\pi(s, e_1, \ldots, e_i))$, this means that:

$$s \xrightarrow{\tau'_1, e_1} s'_1 \cdots \xrightarrow{\tau'_i, e_i} s'_i \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), s)\,.$$

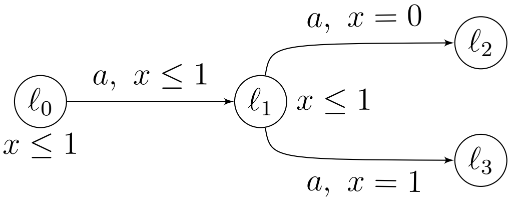
By definition of $\mathsf{R}(\mathcal{A})$, we have that $s_k \approx_t s'_k$ (i.e. s_k and s'_k are region equivalent) for $k = 1, \ldots, i$. Hence, since the region equivalence is a time-abstract bisimulation, there exists $\tau'_{i+1}, \ldots, \tau'_n$ such that

$$s \xrightarrow{\tau'_1, e_1} s'_1 \cdots \xrightarrow{\tau'_i, e_i} s'_i \cdots \xrightarrow{\tau'_n, e_n} s_n \in \mathsf{Runs}(\mathsf{R}(\mathcal{A}), s),$$

with $s_j \approx_t s'_j$ for $j = 1, \ldots, n$. In particular, this implies that $(\tau'_1, \ldots, \tau'_i)$ belongs to the projection of $\mathsf{Pol}(\pi(s, e_1, \ldots, e_n))$ on the i first coordinates.

The other inclusion is straightforward. □

Let us notice that Lemma E is only true in the region automaton. Indeed, let $\mathcal{A}$ be the timed automaton depicted below where e_i denotes the transition ending in ℓ_i, for $i = 1, 2, 3$. Let us consider the (unconstrained) symbolic path $\pi(s, e_1, e_2)$ We have that $\mathsf{Pol}(\pi(s, e_1, e_2)) = \{(0, 0)\}$, hence the projection of $\mathsf{Pol}(\pi(s, e_1, e_2))$ on the first coordinate reduce in the single point $\{0\}$, although $\mathsf{Pol}(\pi(s, e_1)) = [0, 1]$.



Lemma F *Let $\pi = \pi(s, e_1, \ldots, e_n)$ be a symbolic path in $\mathsf{R}(\mathcal{A})$. Then*

$$\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \top \;\Leftrightarrow\; \begin{cases} \dim_{\mathsf{R}(\mathcal{A})}(\pi(s, e_1, \ldots, e_{n-1})) = \top \textit{ and,} \\ \forall s' \textit{ s.t. } s \xrightarrow{e_1} \cdots \xrightarrow{e_{n-1}} s',\ \mu_{s'}(I(s', e_n)) > 0 \end{cases}$$

Proof. Assume $\dim_{\mathsf{R}(\mathcal{A})}(\pi(s, e_1, \ldots, e_n)) = \top$. By Lemma E, we know that in $\mathsf{R}(\mathcal{A})$ the polyedron $\mathsf{Pol}(\pi(s, e_1, \ldots, e_i))$ is exactly the projection of $\mathsf{Pol}(\pi(s, e_1, \ldots, e_n))$ on the i first coordinates. This implies that all prefixes of π also have defined dimension. Hence $\dim_{\mathsf{R}(\mathcal{A})}(\pi(s, e_1, \ldots, e_{n-1})) = \top$. Let s' be a configuration reachable from s via the edges $e_1, \ldots, e_{n-1}$. Assume $\mu_{s'}(I(s', e_n)) = 0$. This means that there exists an edge $e \neq e_n$ such that $\dim(I(s', e_n)) < \dim(I(s', e))$ (thanks to the hypotheses on the measure $\mu_{s'}$). It can only be the case that $I(s', e_n)$ is a singleton, and $I(s', e)$ is an interval, since e_n can be fired in s'. Hence $\dim(\mathsf{Pol}(\pi(s', e_n))) < \dim(\mathsf{Pol}(\pi(s', e)))$. This holds for any configuration s' reachable from s via $e_1 \ldots e_n$ (since the μ measures are assumed to be equivalent inside a region). Thus $\dim(\mathsf{Pol}(\pi(s, e_1, \ldots, e_n))) < \dim(\mathsf{Pol}(\pi(s, e_1, \ldots, e_{n-1}, e)))$, which contradicts $\dim_{\mathsf{R}(\mathcal{A})} = \top$. We therefore conclude $\mu_{s'}(I(s', e_n)) > 0$.

Assume now $\dim_{\mathsf{R}(\mathcal{A})}(\pi(s, e_1, \ldots, e_{n-1})) = \top$ and for all s' such that $s \xrightarrow{e_1} \cdots \xrightarrow{e_{n-1}} s'$, $\mu_{s'}(I(s', e_{n-1})) > 0$. Since the dimension of $\pi(s, e_1, \ldots, e_{n-1})$ is defined, for all $i \leq n-1$, $\dim(\mathsf{Pol}(\pi(s, e_1, \ldots, e_i))) = \dim(\bigcup_e \mathsf{Pol}(\pi(s, e_1, \ldots, e_{i-1}, e)))$. It suffices to show that this still holds for $i = n$. The assumptions on μ yield that, for any edge e, $\dim(I(s', e)) \leq \dim(I(s', e_{n-1}))$. Hence $\dim(\mathsf{Pol}(\pi(s, e_1, \ldots, e_n))) \geq \dim(\mathsf{Pol}(\pi(s, e_1, \ldots, e_{n-1}, e)))$ for all edges e. This concludes the proof. □

Remark. The previous lemma still holds if we replace $\forall s'$ with $\exists s'$, by region equivalence.

The following is an iterated version of Lemma F:

Corollary G *Let $\pi = \pi(s, e_1, \ldots, e_n)$ be a symbolic path in $\mathsf{R}(\mathcal{A})$. Then*

$$\begin{aligned}\dim_{\mathsf{R}(\mathcal{A})}(\pi) = \top &\Leftrightarrow \forall i \leq n\ \forall s_i\ s.t.\ s \xrightarrow{e_1} \cdots \xrightarrow{e_{i-1}} s_i,\ \mu_{s_i}(I(s_i, e_i)) > 0.\\ &\Leftrightarrow \forall i \leq n\ \exists s_i\ s.t.\ s \xrightarrow{e_1} \cdots \xrightarrow{e_{i-1}} s_i,\ \mu_{s_i}(I(s_i, e_i)) > 0.\end{aligned}$$