Almost-Sure Model-Checking of Reactive Timed Automata

Patricia Bouyer*, Thomas Brihaye†, Marcin Jurdziński‡ and Quentin Menet†
* LSV – CNRS & ENS Cachan – France
Email: bouyer@lsv.ens-cachan.fr
† Université de Mons – Belgium
Emails: {thomas.brihaye,quentin.menet}@umons.ac.be
‡ University of Warwick – UK
Email: marcin.jurdzinski@dcs.warwick.ac.uk

***Abstract*—We consider the model of stochastic timed automata, a model in which both delays and discrete choices are made probabilistically. We are interested in the almost-sure model-checking problem, which asks whether the automaton satisfies a given property with probability 1. While this problem was shown decidable for single-clock automata few years ago, it was also proven that the algorithm for this decidability result could not be used for general timed automata. In this paper we describe the subclass of *reactive* timed automata, and we prove decidability of the almost-sure model-checking problem under that restriction. Decidability relies on the fact that this model is almost-surely fair. As a desirable property of real systems, we show that reactive automata are almost-surely non-Zeno. Finally we show that the almost-sure model-checking problem can be decided for specifications given as deterministic timed automata.**

I. Introduction

These last twenty years a huge effort has been made to design expressive models for representing computerized systems. As part of this effort timed automata have been proposed in the early nineties [1] as a suitable model for representing systems with real-time constraints. Numerous works have focused on that model, and it has received an important tool support, with for instance the development of tools like Uppaal [2] or Kronos [3]. Given the success of the timed-automata-based technology for verifying real-time systems, many extensions have been proposed, with the aim of representing the systems more faithfully. They include timed games [4], which can model control problems, priced timed automata [5], [6], [7], which can model energy consumption, stochastic extensions of timed automata [8], [9], [10], [11], [12], [13], which can model randomized aspects of systems or protocols.

In this paper, we are interested in developping verification algorithms for systems which integrate real-time constraints as well as randomized aspects. These two features are important in many applications (see e.g. [14]) but analysis of systems integrating such features is challenging. We distinguish two main different approaches in the literature.

A first way of handling real-time and stochastic features is to assume the systems are modelled as continuous-time Markov chains (CTMCs in short), and timing constraints are given by the properties that are checked. These properties can be given either as formulas of e.g. the logic CSL or extensions thereof [15], [16], [17], [18], or (deterministic) timed automata [19]. This has led to the development of exact and approximated model-checking algorithms.

Another approach is to integrate both features into a complex model (extending e.g. timed automata or Petri nets – here we focus on timed automata), and to analyze this model. Such models include probabilistic timed automata [9] where discrete distributions are assigned to actions and for which tools like Prism [20] have been developed. Delays or durations of events can also be made randomized. This is done for instance in [21], [22] and later in [8], yielding either independent events (in the first papers) and exact model-checking algorithms (for a probabilistic and timed extension of CTL), or approximate model-checking algorithms. The present work is based on the model that was proposed few years ago in [10], [11]. There, timed automata are given a probabilistic semantics, where both delays and discrete choices are randomized. This model has later been extended with non-determinism and interaction [13], but in this paper we focus on the original purely stochastic model. Note that the initial motivation for defining that semantics was robustness of timed systems (since unlikely behaviours are removed by the semantics), but this defines as well an interesting model with real-time constraints and stochastic information.

Our contributions: We are interested in the almost-sure model-checking of stochastic processes defined by timed automata. This problem asks, given a timed automaton $\mathcal{A}$ and a property φ, whether $\mathcal{A}$ satisfies φ with probability 1. This problem has been shown decidable in [11] for *single-clock* timed automata and ω-regular properties. This decidability result relies on the construction of a finite Markov chain $\mathsf{MC}(\mathcal{A})$ such that φ almost-surely holds equivalently in $\mathcal{A}$ and in $\mathsf{MC}(\mathcal{A})$. It was however also shown that the abstraction $\mathsf{MC}(\mathcal{A})$ is *not correct* for two-clock timed automata (the counter-example, depicted on Fig. 3 will be commented later). In this work, we show that if a timed automaton $\mathcal{A}$ is *almost-surely fair* (that is, any edge which is enabled infinitely often is taken infinitely often), then $\mathsf{MC}(\mathcal{A})$ is a correct abstraction. The main result is then a condition on general timed automata under which they are almost-surely fair. The condition ex-

presses that timed automata should be *reactive*, that is, at any time, a discrete transition should be enabled. This assumption is rather natural for modelling real systems. It is interesting to notice that this assumption implies in particular that time-converging (*i.e. Zeno*) behaviours have probability zero, which is a desirable property of real(istic) systems. We can also notice that CTMCs are very simple reactive single-clock timed automata. The proof that reactive automata are almost-surely fair is involved. The key ingredient is to show that almost-surely we visit infinitely often regions where clocks are either very large or equal to 0. Then a judicious use of Borel-Cantelli lemma allows to deduce almost-sure fairness.

The above analysis holds for (state-based) ω-regular properties. Using a product construction, we extend it to properties given as deterministic timed automata (with arbitrarily many clocks), and show that almost-sure model-checking of reactive stochastic automata against properties given as deterministic timed automata is PSPACE-complete. Up to our knowledge this is the first work that establishes a decidability result for stochastic processes when properties are given as deterministic timed automata (with arbitrarily many clocks).

Related work: stochastic processes is huge. We already mentioned several related works, but we would like to discuss a bit more the works [17], [19], which we think are the closest to the present paper. In both papers the model is that of CTMCs. Timing constraints are expressed in the properties, either given as deterministic timed automata [19] or as an extension of CSL called $\mathsf{CSL}_{\mathsf{TA}}$ [17], which extends CSL with properties given as single-clock deterministic timed automata.

Paper [19] is interested in quantitative model-checking, that is, given a CTMC $\mathcal{C}$ and a property given as a determinitic (Muller) timed automaton $\mathcal{A}$, the aim is to compute the probability that runs of $\mathcal{C}$ are accepted by $\mathcal{A}$. This probability is characterized using Volterra integral equations, which can be transformed into linear equations when $\mathcal{A}$ has a unique clock. Therefore quantitative verification can be done for single-clock specifications but can only be approximated in the general case. Our results are somehow incomparable since we allow for a more general model (stochastic timed automata instead of CTMCs) but prove decidability only for the qualitative model-checking problem.

Paper [17] is interested in model-checking of CTMCs against properties expressed as formulas of $\mathsf{CSL}_{\mathsf{TA}}$. This logic involves probability formulas, and uses single-clock deterministic timed automata as predicates. Model-checking of the general logic can be approximated, but if formulas contain only qualitative subformulas, the model-checking can be decided. We do not consider logics, but we allow general deterministic timed automata in our specifications.

Organisation of the paper: Section II presents stochastic (reactive) timed automata, the almost-sure model-checking problem, and explains how it can be reduced to proving almost-sure fairness. part of the paper, and establishes that reactive stochastic automata are almost-surely fair. We give various applications (in terms of decidability of the almost-surel model-checking problem) in Section IV before giving conclusions and further research directions in Section V.

Detailed proofs can be found in the research report [23].

II. Definitions

A. The timed automaton model

Let X be a finite set of variables, called *clocks*. A *clock valuation* over X is a mapping $v\colon X \to \mathbb{R}_+$, where $\mathbb{R}_+$ is the set of nonnegative reals. We write $\mathbb{R}_+^X$ for the set of clock valuations over X. If $v \in \mathbb{R}_+^X$ and $\tau \in \mathbb{R}_+$, we write $v+\tau$ for the clock valuation defined by $(v+\tau)(x) = v(x)+\tau$ if $x \in X$. If $Y \subseteq X$, the valuation $[Y \leftarrow 0]v$ is the valuation assigning 0 to $x \in Y$ and $v(x)$ to $x \notin Y$. A *guard* (or *clock constraint*) 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 $v \models g$, if v satisfies g.

Definition 1: A *timed automaton* is a tuple $\mathcal{A} = (L, \ell_0, X, E)$ such that: (i) L is a finite set of locations, (ii) $\ell_0 \in L$ is the initial location, (iii) X is a finite set of clocks, and (iv) $E \subseteq L \times \mathcal{G}(X) \times 2^X \times L$ is a finite set of edges.

If e is an edge of $\mathcal{A}$, we write $\mathsf{source}(e)$ (resp. $\mathsf{target}(e)$) for the source (resp. target) of e defined by ℓ (resp. ℓ') if $e = (\ell, g, Y, \ell')$. The semantics of a timed automaton $\mathcal{A}$ is a timed transition system whose states are pairs $(\ell, v) \in L \times \mathbb{R}_+^X$, and whose transitions are of the form $(\ell, v) \xrightarrow{\tau, e} (\ell', v')$ when there exists an edge $e = (\ell, g, Y, \ell')$ such that $v+\tau \models g$ and $v' = [Y \leftarrow 0](v+\tau)$. A finite (resp. infinite) *run* ϱ of $\mathcal{A}$ is a finite (resp. infinite) sequence of consecutive transitions, *i.e.*, $\varrho = s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \dots$ where for each $i \geq 0$, $s_i = (\ell_i, v_i)$ is a state. We write $\mathsf{Runs}_f(\mathcal{A}, s_0)$ (resp. $\mathsf{Runs}(\mathcal{A}, s_0)$) for the set of finite runs (resp. infinite runs) of $\mathcal{A}$ from state s_0. If ϱ is a finite run in $\mathcal{A}$, we write $\mathsf{last}(\varrho)$ for the last state of ϱ. Given a state s of $\mathcal{A}$ and an edge e, we define $I(s,e) = \{\tau \in \mathbb{R}_+ \mid s \xrightarrow{\tau, e} s'\}$ and $I(s) = \bigcup_e I(s,e)$. The automaton $\mathcal{A}$ is *reactive* if for every state s, $I(s) = \mathbb{R}_+$.

Symbolic paths: 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 finite runs: $\pi(s, e_1 \dots e_n) = \{\varrho = s \xrightarrow{\tau_1, e_1} s_1 \dots \xrightarrow{\tau_n, e_n} s_n\}$. Given an n-variable constraint $\mathcal{C}$, the *constrained* symbolic path $\pi_{\mathcal{C}}(s, e_1 \dots e_n)$ is the subset of $\pi(s, e_1 \dots e_n)$ where the delays τ_1 to τ_n satisfy the constraint $\mathcal{C}$. Let π be a finite (constrained) symbolic path, we define the *cylinder* generated by π as the set $\mathsf{Cyl}(\pi)$ of infinite paths ϱ such that a prefix ϱ' of ϱ is in π.

B. The timed region automaton

The well-known region automaton construction is a finite abstraction of timed automata which can be used for verifying many properties like ω-regular untimed properties [1]. We recall the notion of region equivalence. Let $\mathcal{A} = (L, \ell_0, X, E)$ be a timed automaton, and let M be the largest integer appearing in a guard of $\mathcal{A}$. Let v and v' be valuations (over X). They are said region equivalent w.r.t. $\mathcal{A}$, which we write $v \cong_{\mathcal{A}} v'$ whenever the following conditions hold:

- for every $x \in X$, either $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ or $v(x), v'(x) > M$, and $\{v(x)\} = 0$ iff $\{v'(x)\} = 0$;[1]
- for every $x, y \in X$ such that $v(x), v(y) \leq M$, $\{v(x)\} \leq \{v(y)\}$ iff $\{v'(x)\} \leq \{v'(y)\}$.

A *region* of $\mathcal{A}$ is an equivalence class of $\cong_{\mathcal{A}}$, and if v is a valuation over X, $[v]_{\mathcal{A}}$ is the equivalence class of v w.r.t. $\cong_{\mathcal{A}}$. We write $R_{\mathcal{A}}$ for the set of regions of automaton $\mathcal{A}$. A region r is said *memoryless* whenever the following holds for every clock $x \in X$: either $v(x) = 0$ for every $v \in r$, or $v(x) > M$ for every $v \in r$. Memoryless regions will play an important role later. For each such region r, we distinguish a canonical valuation $v_r \in r$ defined by $v_r(x) = 0$ or $v_r(x) = M + 1$ for every $x \in X$.

Here we define a timed version of the standard region automaton. Let $\mathcal{A} = (L, \ell_0, X, E)$ be a timed automaton. The *timed region automaton* of $\mathcal{A}$ is the timed automaton $\mathsf{R}(\mathcal{A}) = (Q, q_0, X, T)$ such that:

- $Q = L \times R_{\mathcal{A}}$, $q_0 = (\ell_0, [\mathbf{0}]_{\mathcal{A}})$ where $\mathbf{0}$ is the valuation assigning 0 to every clock;
- $T \subseteq (Q \times \mathsf{cell}(R_{\mathcal{A}}) \times 2^X \times Q)$ (where $\mathsf{cell}(R_{\mathcal{A}}) = \{\mathsf{cell}(r) \mid r \in R_{\mathcal{A}}\}$, writing $\mathsf{cell}(r)$ for the tightest guard containing r) is such that: $(\ell, r) \xrightarrow{\mathsf{cell}(r''), Y} (\ell', r')$ is in T iff there exists $e = \ell \xrightarrow{g, Y} \ell'$ in E s.t. there exist $v \in r$, $\tau \in \mathbb{R}_+$ with $(\ell, v) \xrightarrow{\tau, e} (\ell', v')$, $v + \tau \in r''$ and $v' \in r'$. In that case we say that transition $(\ell, r) \xrightarrow{\mathsf{cell}(r''), Y} (\ell', r')$ maps to e.

We recover the usual region automaton of [1] by interpreting $\mathsf{R}(\mathcal{A})$ as a finite automaton. The above timed interpretation satisfies strong timed bisimulation properties that we do not detail here. To every symbolic path $\pi((\ell, v), e_1 \dots e_n)$ in $\mathcal{A}$ correspond finitely many symbolic paths $\pi(((\ell, [v]_{\mathcal{A}}), v), f_1 \dots f_n)$ in $\mathsf{R}(\mathcal{A})$, each one corresponding to a choice in the regions that are visited. If ϱ is a run in $\mathcal{A}$, we denote by $\iota(\varrho)$ its unique image in $\mathsf{R}(\mathcal{A})$. Note that if $\mathcal{A}$ is reactive, then so is $\mathsf{R}(\mathcal{A})$. If $q = (\ell, r) \in Q$ is such that r is memoryless, we distinguish the canonical configuration $s_q = (\ell, v_r)$ (or $s_q = ((\ell, r), v_r)$ if we speak of $\mathsf{R}(\mathcal{A})$).

C. Stochastic timed automata

1) The general framework: Following [11], we will define a probability measure over sets of infinite runs of a timed automaton, that will quantify in some sense their *likelihood*. We assume that each timed automaton $\mathcal{A}$ comes equipped with probability distributions from every state of $\mathcal{A}$ both over delays and over enabled edges. In [11], we have worked with rather general classes of probability distributions over delays (the only restrictions were to avoid pathological behaviours).

A *stochastic timed automaton* is a tuple $\mathcal{A} = (L, \ell_0, X, E, (\mu_s, p_s)_{s \in L \times \mathbb{R}_+^X})$ where:

- (L, ℓ_0, X, E) is a timed automaton;
- for every $s = (\ell, v) \in L \times \mathbb{R}_+^X$, μ_s is a probability distribution over $I(s)$ and p_s is a probability distribution over $\{(\ell, g, Y, \ell') \in E \mid v \models g\}$.

[1] $\lfloor \cdot \rfloor$ (resp. $\{\cdot\}$) denotes the integral (resp. fractional) part.

We refer to [11, Section 3.1] for the restrictions on the measures that are required in order for the following to be well-defined. We inductively define a measure over finite symbolic paths $\pi(s, e_1 \dots e_n)$ as:

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

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

The value $\mathbb{P}_{\mathcal{A}}(\pi(s, e_1 \dots e_n))$ is the result of n successive one-dimensional integrals, but it can also be viewed as the result of an n-dimensional integral. Hence, we can easily extend the above definition to finite constrained paths $\pi_{\mathcal{C}}(s, e_1 \dots e_n)$ when $\mathcal{C}$ is Borel-measurable. This extension to constrained paths will allow to express (and later, measure) various and rather complex sets of paths. The measure $\mathbb{P}_{\mathcal{A}}$ can then be defined on cylinders, letting $\mathbb{P}_{\mathcal{A}}(\mathsf{Cyl}(\pi)) = \mathbb{P}_{\mathcal{A}}(\pi)$ if π is a finite (constrained) symbolic path from state s. Finally we extend $\mathbb{P}_{\mathcal{A}}$ in a standard and unique way to the σ-algebra generated by these cylinders, which we note $\Omega^s_{\mathcal{A}}$ (see [11] for details).

Proposition 2 ([11]): Let $\mathcal{A}$ be a timed automaton. For every state s, the function $\mathbb{P}_{\mathcal{A}}$ is a probability measure over $(\mathsf{Runs}(\mathcal{A}, s), \Omega^s_{\mathcal{A}})$.

Example 3: The set $\mathsf{Zeno}(s)$ of all the Zeno runs[2] starting from s belongs to $\Omega^s_{\mathcal{A}}$. It can indeed be defined as:

$$\bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \dots, e_n) \in E^n} \mathsf{Cyl}(\pi_{\mathcal{C}_{M,n}}(s, e_1 \dots e_n))$$

where $\mathcal{C}_{M,n}$ is the constraint $\left(\sum_{1 \leq i \leq n} \tau_i\right) \leq M$.

2) The class of reactive stochastic timed automata: In this paper we will be mostly interested in the subclass of reactive stochastic timed automata. A stochastic timed automaton $\mathcal{A} = (L, \ell_0, X, E, (\mu_s, p_s)_{s \in L \times \mathbb{R}^X})$ is reactive whenever the underlying timed automaton is reactive, and:

- for every $\ell \in L$, there exists a probability distribution μ_ℓ over $\mathbb{R}_+$, equivalent to the Lebesgue measure,[3] such that for every $v \in \mathbb{R}_+^X$, $\mu_{(\ell, v)} = \mu_\ell$;
- for every edge e, there exists $w_e \in \mathbb{N}_{>0}$ s.t. for all s:

$$p_s(e) = \begin{cases} w_e / \left(\sum_{e' \text{ enabled at } s} w_{e'}\right) & \text{if } e \text{ enabled at } s \\ 0 & \text{otherwise} \end{cases}$$

Note that for any constant M, for any $\ell \in L$, $\mu_\ell([0, M]) < 1$, this is due to the equivalence of μ_ℓ with the Lebesgue measure. Note also that if $s = (\ell, v)$ and $s' = (\ell, v')$ are such that $v \cong_{\mathcal{A}} v'$, then $p_s(e) = p_{s'}(e)$ for every edge e.

[2] An infinite run ϱ: $s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} \cdots$ is said *Zeno* whenever $\sum_{i=1}^{\infty} \tau_i$ is bounded.

[3] Two measures μ and μ' are equivalent whenever for every measurable set E, $\mu(E) = 0$ iff $\mu'(E) = 0$.

Example 4: Examples of distributions over delays that respect the above conditions are exponential distributions, but we can think of many other kinds of distributions like the gamma distributions. Later, all our examples will use exponential distributions. In that case it is characterized by a positive parameter λ_ℓ, and the density of the distribution is then $t \mapsto \lambda_\ell \cdot e^{-\lambda_\ell \cdot t}$.

Note that reactive stochastic timed automata generalise continuous-time Markov chains (CTMC for short). A CTMC is nothing else than a single-clock reactive stochastic timed automaton in which (i) on all transitions, the guard is trivial, and the clock is reset, and (ii) each location is assigned an exponential distribution over delays.

Remark 5: Note that reactive stochastic timed automata are simple to compose, if we assume all distributions on delays are given by exponential distributions. Indeed applying race conditions to pairs of states, one can rather easily define a composed system. An example of composition is detailed later on an example (see Section II-D below).

3) Reduction to timed region automata: In [11], it is explained how to transfer probabilities from $\mathcal{A}$ to $\mathsf{R}(\mathcal{A})$, thus allowing to prove results on $\mathsf{R}(\mathcal{A})$ and to recover them on the original automaton $\mathcal{A}$. We assume that, for every state s in $\mathcal{A}$, $\mu^{\mathcal{A}}_s = \mu^{\mathsf{R}(\mathcal{A})}_{\iota(s)}$, and for every $t \in \mathbb{R}_+$, $p^{\mathcal{A}}_{s+t} = p^{\mathsf{R}(\mathcal{A})}_{\iota(s)+t}$. Under those assumptions, we have the following transfer result.

Lemma 6 ([11]): Assume measures in $\mathcal{A}$ and in $\mathsf{R}(\mathcal{A})$ are related as above. Then, for every set S of runs in $\mathcal{A}$ we have: $S \in \Omega^s_{\mathcal{A}}$ iff $\iota(S) \in \Omega^{\iota(s)}_{\mathsf{R}(\mathcal{A})}$, and in this case $\mathbb{P}_{\mathcal{A}}(S) = \mathbb{P}_{\mathsf{R}(\mathcal{A})}(\iota(S))$.

D. Two examples

We first describe a naive cooling system, and then briefly discuss the IPv4 Zeroconf protocol that was described in [24, p. 751].

1) A cooling system: In order to illustrate the expressiveness of stochastic timed automata, we design a toy example providing a naive model for a cooling system in a power plant. The cooling system is composed of n similar cooling tanks modelled by the automata $\mathcal{A}_i$, $1 \leq i \leq n$, depicted in Fig. 1. Our model of a tank has two states: $\textsc{Down}_i$ and $\textsc{Up}_i$. The state $\textsc{Up}_i$ models a situation where tank_i is operational and could be used to cool the core if needed. The state $\textsc{Down}_i$ models a situation where tank_i is down and is thus unable to cool the core if needed. The probability to leave the $\textsc{Down}_i$ (resp. $\textsc{Up}_i$) state follows an exponential law of parameter λ^i_D (resp. λ^i_U). Moreover we make the hypothesis that once a cooler tank is down, it takes at least τ^i_r time units to repair it.

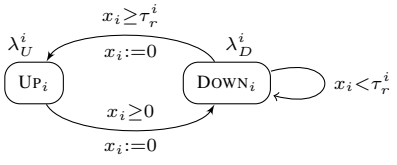

Fig. 1. $\mathcal{A}_i$ for tank_i.

The stochastic timed automaton modelling the cooling system is then given by the product of the n tanks: $\mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$, which contains n independent clocks. Notice that the presence of each clock is necessary, since several tanks can be down at the same time. In particular, when tank_1 is in state $\textsc{Down}_1$, it is needed to keep in memory (thanks to clock x_1) the last time it entered $\textsc{Down}_1$, to evaluate the guard $x_1 \geq \tau^1_r$. In order to illustrate the composition of stochastic automata, we represent the system $\mathcal{A}_1 \times \mathcal{A}_2$ in Fig. 2. Note that this is done in the same spirit as for CTMCs [25].

To keep the picture readable, probability distributions on locations and transitions are omitted on the figure. With any pair of locations (ℓ_1, ℓ_2) in $\mathcal{A}_1 \times \mathcal{A}_2$, we associate the exponential distribution of parameter $\lambda_{\ell_1} + \lambda_{\ell_2}$, where λ_{ℓ_1} and λ_{ℓ_2} are the parameters of the exponential distribution of ℓ_1 and ℓ_2 respectively. To any edge (ℓ_1, g, Y, ℓ'_1) of $\mathcal{A}_1$ corresponds the edges $((\ell_1, \ell_2), g, Y, (\ell'_1, \ell_2))$ of $\mathcal{A}_1 \times \mathcal{A}_2$, where ℓ_2 is any location of $\mathcal{A}_2$. The probability to take an enabled edge $((\ell_1, \ell_2), g, Y, (\ell'_1, \ell_2))$ is given by $\frac{\lambda_{\ell_1}}{\lambda_{\ell_1} + \lambda_{\ell_2}}$. A similar construction occurs for the edges of $\mathcal{A}_2$. Let us observe that in general, the probability of the edges of a product $\mathcal{A} \times \mathcal{B}$ also depends of the probability of the edges in $\mathcal{A}$ and in $\mathcal{B}$. In our example the latter probabilities were all equal to one.

The composition described above is motivated by the following properties of exponential distributions. Given X_1 (resp. X_2) a random variable of exponential law $\exp(\lambda_1)$ (resp. $\exp(\lambda_2)$), the random variable given by $\min(X_1, X_2)$ has a distribution $\exp(\lambda_1 + \lambda_2)$. Moreover we have that

$$\mathbb{P}(X_i = \min(X_1, X_2)) = \frac{\lambda_i}{\lambda_1 + \lambda_2} \ , \ i = 1, 2.$$

The memoryless property of exponential distributions also plays an important role to give a sense to the composition.

From a syntactic point of view composition could also be defined with other laws. However to keep the intended meaning of composition we would need to consider measure μ_s depending not only on the location, but also on the clocks valuations, in order to cope with the absence of the memoryless property. In the case of exponential distributions, it is not hard to get that the composition of two (reactive) stochastic timed automata is a (reactive) stochastic timed automaton.

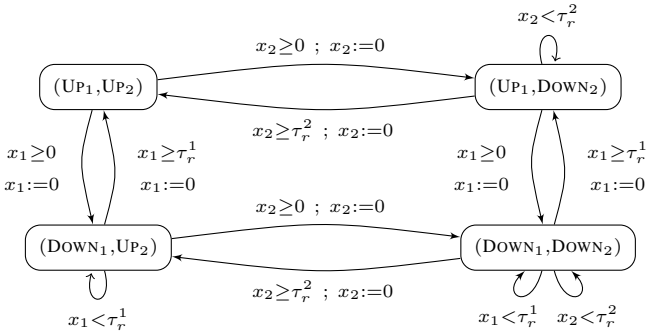

Fig. 2. The product automaton $\mathcal{A}_1 \times \mathcal{A}_2$.

The core of the plant has two modes, LOW and HIGH, modelling respectively a situation where the temperature of the core is *"normal"* and *"too high"*. When in use, the plant warms up and therefore can go from state LOW to state HIGH following an exponential law of parameter λ_L. When in mode HIGH, the plant needs to be cooled down, and its evolution depends on the current state of the cooling system (the more tanks are available the faster the plant goes back to the LOW mode). We do not depict the whole system, but it can be modelled as a reactive stochastic timed automaton.

We now discuss examples of desirable properties for this system. For instance, we could verify that *"At each time, at least two tanks are operational."*. If we assume that the core is highly damaged as soon as it is overheated for more than τ_0 time units, we would like to ensure that *"The core never spends more than τ_0 time units in the* HIGH *state"*. We could also require that *"Each time the core enters in the* HIGH *state, at least two cooling tanks are operational within τ_1 time units."*. The first property can clearly be modelled via an LTL formula, although the two others can be expressed via a deterministic timed automaton. Later, we provide an algorithm which checks whether these kind of properties are almost surely satisfied. More precisely, we design an algorithm which can check whether properties expressed via LTL formula or deterministic timed automata are true with probability one.

2) The IPv4 Zeroconf protocol: This protocol aims at configuring IP addresses in a local network of appliances. It proceeds as follows when a new appliance is plugged: it selects an IP address in a random way, and broadcasts several probe messages to the network to know whether this address is already used or not. If it receives an answer from the network in a bounded delay, then a new IP address is chosen. It may be the case that those messages get lost, in which case there is an error. In [24, p. 751], a very simple model for the IPv4 Zeroconf protocol is given as a finite Markov chain, which abstracts away timing constraints. We can rewrite the model and express those constraints using stochastic timed automata.

E. The model-checking problem

1) Definition: In this paper we are interested in qualitative model-checking. More precisely, we study the almost-sure model-checking problem. This problem asks, given a stochastic timed automaton $\mathcal{A}$ and a measurable linear-time property φ,[4] whether $\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi) = 1$, where $s_0 = (\ell_0, [\mathbf{0}]_{\mathcal{A}})$ is the initial state of $\mathcal{A}$ and $\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi) = \mathbb{P}_{\mathcal{A}}([\![\varphi]\!]_{s_0})$. If the answer is positive, we say that $\mathcal{A}$ *almost-surely satisfies* φ and we write $\mathcal{A} \mid\!\approx \varphi$.

2) Reduction to almost-sure fairness: Although the value $\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi)$ depends on the probability distributions given in $\mathcal{A}$, we have proven in [11] that for single-clock timed automata the almost-sure satisfaction is not affected by the choice of these distributions. The algorithm for deciding the almost-sure model-checking problem in single-clock automata relies on the construction of a finite Markov chain $\mathsf{MC}(\mathcal{A})$ that such that $\mathcal{A} \mid\!\approx \varphi$ iff the probability of satisfying φ in $\mathsf{MC}(\mathcal{A})$ from the initial state is 1 (φ is assumed to be an ω-regular property). The algorithm is described in Algorithm 1. It uses the notion of singularity for an edge, which is defined as follows: an edge e of $\mathsf{R}(\mathcal{A})$ from a region-state q is *singular* whenever there is $s \in q$ such that $I(s, e)$ is a single point, but there is also another edge e' such that $I(s, e')$ is not a single point.[5] It is worth noticing that if e is singular and leaves q, it holds that $\mu_s(I(s, e)) = 0$ for every $s \in q$ (this is actually an equivalence). This is actually the reason why those edges have to be removed: they have probability 0 to be taken. In Algorithm 1, property $\widetilde{\varphi}$ is just the lifting of φ to $\mathsf{MC}(\mathcal{A})$ (which has not the same set of states as $\mathcal{A}$), it is also ω-regular.

Algorithm 1: Algorithm for the qualitative model-checking .

Data: A stochastic timed automaton $\mathcal{A}$ and an ω-regular property φ
Result: Yes iff $\mathcal{A} \mid\!\approx \varphi$?

```
begin
    Build the region automaton R(A) of A;
    Remove singular transitions in R(A) and
    non-reachable states;
    Write MC(A) for the resulting structure, q_0 for its
    initial state, and interpret MC(A) as a finite Markov
    chain (with uniform weights on edges);
    if P_{MC(A)}(q_0 |= φ̃) = 1 then
        answer 'Yes';
    else
        answer 'No';
    end
end
```

Algorithm 1 is rather natural, but its correctness is not obvious. And actually, it is proven in [11] that it is not correct for two-clock automata. The automaton of Figure 3 witnesses the problem: the probability of never visiting the top-most branch in $\mathcal{A}$ is positive, whereas the probability of never visiting the top-most branch in the corresponding finite Markov chain is 0. The intuition is: the more we loop, the closer will be y to 1 when reaching ℓ_0, making the choice of e_1 less and less probable. See [27, Section 4.2] for detailed computations.

We will now give a condition for Algorithm 1 to be correct, which relies on a notion of fairness for infinite paths. An infinite path $q_0 \xrightarrow{g_1, Y_1} q_1 \xrightarrow{g_2, Y_2} \dots$ in $\mathsf{R}(\mathcal{A})$ is *fair* if for every non-singular edge f of $\mathsf{R}(\mathcal{A})$, if there are infinitely many i's such that $\mathsf{source}(f) = q_i$, then there are infinitely many i's such that $f = q_i \xrightarrow{g_{i+1}, Y_{i+1}} q_{i+1}$. Fairness extends

[4] A property φ is said measurable whenever the set $[\![\varphi]\!]_{s_0} = \{\varrho \in \mathsf{Runs}(\mathcal{A}, s_0) \mid \varrho \models \varphi\}$ is in $\Omega_{\mathcal{A}}^{s_0}$. This is for instance the case of ω-regular properties [26].

[5] Note that if $\mathcal{A}$ is reactive the above condition reduces to require that $I(s, e)$ is a single point for some $s \in q$.

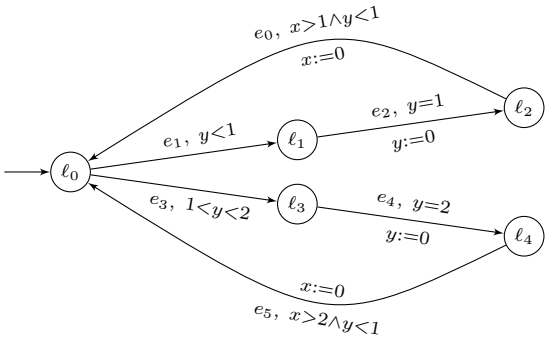

Fig. 3. A two-clock example with non-negligible set of unfair runs .

to underlying (timed) runs and to symbolic infinite paths in an obvious way. Note that if π is an infinite path of $\mathsf{R}(\mathcal{A})$ which does not take any singular transition, then π is fair iff it exists in $\mathsf{MC}(\mathcal{A})$ and it is fair (for the standard notion of fairness in finite Markov chains). Let us also notice that the set of fair paths is measurable since fairness can be expressed via an ω-regular property [26].

The following theorem establishes the correctness of Algorithm 1, under the hypothesis that the automaton is almost-surely fair. It was proven as Theorem 15 in [11] in the case of single-clock automata. However the single-clock hypothesis is only used for proving that fairness is almost-sure. Therefore, the proof of [11] applies here as well and we get the following theorem.

Theorem 7: Let $\mathcal{A}$ be a stochastic timed automaton, and φ an ω-regular property. Assume that $\mathbb{P}_{\mathcal{A}}(s_0 \models \mathsf{fair}) = 1$.[6] Then, $\mathcal{A} \approx \varphi$ iff $\mathbb{P}_{\mathsf{MC}(\mathcal{A})}(q_0 \models \widetilde{\varphi}) = 1$, where $\widetilde{\varphi}$ is the lifting of φ to $\mathsf{MC}(\mathcal{A})$.[7]

Note that almost-sure fairness does not hold for the automaton of Figure 3 at location ℓ_0: each time location ℓ_0 is visited, the value of clock y has increased, and the "weight" of guard $y < 1$ becomes small compared with that of guard $1 < y < 2$. The probability of never taking edge e_1 will therefore be positive (see computations in [27]), which proves that almost-sure fairness fails.

In [11] almost-sure fairness is proven for single-clock timed automata using topological arguments. A topology is put on runs of timed automata, which extends the classical Cantor topology in finite automata, used for instance in [28]. In this topology, large sets coincide with sets of probability 1. The technicalities heavily rely on the characterization of large sets using winning strategies in Banach-Mazur games [29].

The main result of this paper is a proof that almost-sure fairness holds for reactive stochastic automata with arbitrarily many clocks. Contrary to [11], the proof only uses probabilistic arguments. In the next section we state basic properties of probability measures. Section III contains the technical part of the paper, and shows the following result:

[6] $\mathbb{P}_{\mathcal{A}}(s_0 \models \mathsf{fair})$ is the probability of fair runs from s_0.

[7] A run ϱ of $\mathcal{A}$ with only non-singular edges satisfies φ iff its region projection satisfies $\widetilde{\varphi}$ in $\mathsf{MC}(\mathcal{A})$.

Proposition 8: Let $\mathcal{A}$ be a reactive stochastic timed automaton. Then $\mathbb{P}_{\mathcal{A}}(s_0 \models \mathsf{fair}) = 1$.

In Section IV, we discuss applications of this result to the almost-sure model-checking problem, and discuss in particular which properties can be checked.

III. Proving almost-sure fairness

This section is devoted to the proof of Proposition 8. We fix a reactive stochastic timed automaton $\mathcal{A}$ and write $\mathsf{R}(\mathcal{A}) = (Q, q_0, X, E, (\mu_q)_{q\in Q}, (w_e)_{e\in E})$ for its timed region automaton. We will prove that $\mathsf{R}(\mathcal{A})$ is almost-sure fair, which will imply the proposition.

To prove almost-sure fairness in $\mathsf{R}(\mathcal{A})$, we have to show that for every non-singular edge e, the probability to visit e infinitely often, knowing we visit $\mathsf{source}(e)$ infinitely often, is equal to 1. The key point of this proof lies in the fact that as the automaton $\mathsf{R}(\mathcal{A})$ is reactive, the set of runs that visit infinitely often memoryless regions has probability 1 (Subsection III-C).

More precisely, knowing we visit $\mathsf{source}(e)$ infinitely often, the probability to visit infinitely often memoryless regions, from which e is reachable with non-zero probability, will be equal to 1. Then it remains to show that knowing we visit infinitely often such a memoryless region, we visit e infinitely often with probability 1 (Subsection III-D). To this end, we investigate the set of runs that visit infinitely often such a region and e, and we conclude thanks to a judicious decomposition of this set and Borel-Cantelli lemma.

In the following we write M for the maximal constant appearing in $\mathcal{A}$ (or $\mathsf{R}(\mathcal{A})$), and we write $\mathbb{P}$ instead of $\mathbb{P}_{\mathsf{R}(\mathcal{A})}$.

A. Some basic results on conditional probabilities

Let $\mathbb{P}$ be a probability measure on some probabilistic space Ω. We recall that if A and B are measurable and $\mathbb{P}(B) > 0$, then the conditional probability of A given B is defined by

$$\mathbb{P}(A \mid B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Lemma 9: Let A, B and C be measurable sets such that $\mathbb{P}(B) > 0$ and $\mathbb{P}(C) > 0$, then

1) $\mathbb{P}(A \mid B) = 1$ if and only if $\mathbb{P}(A^c \cap B) = 0$.
2) If $\mathbb{P}(A) = 1$, then $\mathbb{P}(A \mid B) = 1$.
3) If $\mathbb{P}(A \mid B) = 1$, $\mathbb{P}(B \mid C) = 1$, then $\mathbb{P}(A \mid C) = 1$.
4) If $\mathbb{P}(A \mid B) = 1$, $\mathbb{P}(A \mid C) = 1$, then $\mathbb{P}(A \mid B \cup C) = 1$.

B. How should we prove almost-sure fairness?

Let $s_0 = (q_0, \mathbf{0})$ be the initial state of $\mathsf{R}(\mathcal{A})$, e an edge in E, and $q \in Q$. We write $\mathfrak{R}^e(s_0)$ for the set of runs in $\mathsf{R}(\mathcal{A})$ that start in s_0 and take e infinitely often, and $\mathfrak{R}^q(s_0)$ for the set of runs of $\mathsf{R}(\mathcal{A})$ that start in s_0 and visit q infinitely often. In particular, we write $\mathfrak{R}^{\mathsf{source}(e)}(s_0)$ for the set of runs that start in s_0 and visit $\mathsf{source}(e)$ infinitely often (hence along which e is enabled infinitely often).

We want to prove that the probability of being fair is 1, hence we want to prove that for every non-singular edge e with $\mathbb{P}(\mathfrak{R}^{\mathsf{source}(e)}(s_0)) > 0$,

$$\mathbb{P}\left(\mathfrak{R}^e(s_0) \mid \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = 1.$$

We let $\mathcal{Q}$ be the set of pairs $q = (\ell, r)$ where r is memoryless and $\mathcal{Q}'_e$ the set of elements $q = (\ell, r) \in \mathcal{Q}$ such that

$$\mathbb{P}(\mathfrak{R}^q(s_0)) > 0 \quad \text{and} \quad \mathbb{P}(\mathfrak{R}_0^{q,e}(s_q)) > 0$$

where $\mathfrak{R}_0^{q,e}(s_q)$ is the set of runs that start from s_q (see page 3 for the definition of s_q) and take e before any other visit to q.

We assume e is a non-singular edge with $\mathbb{P}(\mathfrak{R}^{\mathsf{source}(e)}(s_0)) > 0$. We will prove (Subsection III-D) that for any $q \in \mathcal{Q}'_e$,

$$\mathbb{P}\left(\mathfrak{R}^e(s_0) \mid \mathfrak{R}^q(s_0)\right) = 1 \tag{1}$$

and (Subsection III-C) that

$$\mathbb{P}\left(\bigcup_{q\in\mathcal{Q}} \mathfrak{R}^q(s_0)\right) = 1. \tag{2}$$

Assume that Equations (1) and (2) have been proven. Applying Lemma 9 (point 4.), we deduce from Equation (1) that

$$\mathbb{P}\left(\mathfrak{R}^e(s_0) \mid \bigcup_{q\in\mathcal{Q}'_e} \mathfrak{R}^q(s_0)\right) = 1 \tag{3}$$

and applying Lemma 9 (point 2.), we deduce from Equation (2) that:

$$\mathbb{P}\left(\bigcup_{q\in\mathcal{Q}} \mathfrak{R}^q(s_0) \mid \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = 1. \tag{4}$$

Moreover, we can easily show that

$$\mathbb{P}\left(\bigcup_{q\in\mathcal{Q}} \mathfrak{R}^q(s_0) \mid \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = \mathbb{P}\left(\bigcup_{q\in\mathcal{Q}'_e} \mathfrak{R}^q(s_0) \mid \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right). \tag{5}$$

Indeed, we just have to prove that

$$\mathbb{P}\left(\bigcup_{q\in\mathcal{Q}} \mathfrak{R}^q(s_0) \cap \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = \mathbb{P}\left(\bigcup_{q\in\mathcal{Q}'_e} \mathfrak{R}^q(s_0) \cap \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right)$$

and it is thus sufficient to prove that

$$\mathbb{P}\left(\bigcup_{q\in\mathcal{Q}\setminus\mathcal{Q}'_e} \mathfrak{R}^q(s_0) \cap \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = 0.$$

However, if $q \in \mathcal{Q}\setminus\mathcal{Q}'_e$, we have $\mathbb{P}(\mathfrak{R}^q(s_0)) = 0$ or $\mathbb{P}(\mathfrak{R}_0^{q,e}(s_q)) = 0$. Now, if $\mathbb{P}(\mathfrak{R}^q(s_0)) = 0$, we have

$$\mathbb{P}\left(\mathfrak{R}^q(s_0) \cap \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = 0$$

and if $\mathbb{P}(\mathfrak{R}_0^{q,e}(s_q)) = 0$, we also have

$$\mathbb{P}\left(\mathfrak{R}^q(s_0) \cap \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = 0.$$

We therefore deduce from Equations (4) and (5) that

$$\mathbb{P}\left(\bigcup_{q\in\mathcal{Q}'_e} \mathfrak{R}^q(s_0) \mid \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = 1. \tag{6}$$

Applying Lemma 9 (point 3.), we get the expected result from Equations (3) and (6) :

$$\mathbb{P}\left(\mathfrak{R}^e(s_0) \mid \mathfrak{R}^{\mathsf{source}(e)}(s_0)\right) = 1.$$

It remains to prove the two intermediary results that were used, *i.e.* to prove Equations (1) and (2).

C. Proof of Equation (2)

Lemma 10: $\mathbb{P}\left(\bigcup_{q\in\mathcal{Q}} \mathfrak{R}^q(s_0)\right) = 1.$

Sketch of proof: We notice that the set of runs that delay infinitely many times more than M time units before taking a transition is a subset of $\bigcup_{q\in\mathcal{Q}} \mathfrak{R}^q(s_0)$. Indeed, if a run ϱ delays more than M time units before taking the n-th transition then each clock is either reset on the n-transition (hence its value is 0), or it is above M. Now, as for every $\ell \in L$, we have assumed μ_ℓ is equivalent to the Lebesgue measure on $\mathbb{R}_+$, it holds that $\mu_\ell([0, M]) < 1$. We can then prove that the probability of the set of runs that delay only finitely many times more than M time units is zero, since L is finite, which concludes the proof. ∎

Remark 11: A side-result of the proof of this lemma is that $\mathbb{P}(\mathsf{Zeno}(s_0)) = 0$, where $\mathsf{Zeno}(s_0)$ is the set of Zeno runs from s_0.

D. Proof of Equation (1)

Let $q \in \mathcal{Q}'_e$. We want prove that

$$\mathbb{P}\left(\mathfrak{R}^e(s_0) \mid \mathfrak{R}^q(s_0)\right) = 1$$

or equivalently that

$$\mathbb{P}\left(\mathfrak{R}^e(s_0) \cap \mathfrak{R}^q(s_0) \mid \mathfrak{R}^q(s_0)\right) = 1.$$

We notice that the event $\mathfrak{R}^e(s_0) \cap \mathfrak{R}^q(s_0)$ coincides with

$$\bigcap_{n\in\mathbb{N}} \bigcup_{k\geq n} \mathfrak{R}_k^{q,e}(s_0)$$

where $\mathfrak{R}_k^{q,e}(s_0)$ is the set of runs starting in s_0 along which an occurrence of edge e is preceded by precisely k visits to q, i.e. $\mathfrak{R}_k^{q,e}(s_0) = \{\varrho \in \mathsf{Runs}(\mathcal{A}, s_0) \mid \varrho = s_0 \xrightarrow{\tau_1,e_1} s_1 \dots \xrightarrow{\tau_m,e_m} s_m \dots$ and there exists j s.t. $e_j = e$ and $\#\{1 \leq i < j \mid \mathsf{loc}(s_i) = q\} = k\}$, where $\mathsf{loc}(s_i)$ is the location of state s_i. We recall the following lemma, which is well-known in probability theory (see for example [30]):

Lemma 12 (Second Borel-Cantelli Lemma): Assume $(\mathcal{E}, \mathbb{P})$ is a probabilistic space, and that the measurable events $(E_k)_{k\in\mathbb{N}}$ are independent. If $\sum_{k\in\mathbb{N}} \mathbb{P}(E_k) = +\infty$, then

$$\mathbb{P}\left(\bigcap_{n\in\mathbb{N}} \bigcup_{k\geq n} E_k\right) = 1.$$

With the aim to apply this lemma, we will prove that the events $\mathfrak{R}_k^{q,e}(s_0)$ are independent in the $\mathfrak{R}^q(s_0)$-conditional σ-algebra, and that $\sum_{k\in\mathbb{N}} \mathbb{P}(\mathfrak{R}_k^{q,e}(s_0) \mid \mathfrak{R}^q(s_0)) = +\infty$, which will imply Equation (1). This is non-trivial and will require several technical lemmas that we present now. The following arguments rely on result that will be given as Corollary 18 (which is technical, and therefore postponed).

a) Independence of events:

Lemma 13: The events $\mathfrak{R}_k^{q,e}(s_0)$ are conditionally independent given $\mathfrak{R}^q(s_0)$.

Sketch of proof: Defining $\mathfrak{R}^q_{\geq n}(s_0)$ as the set of runs starting in s_0 and visiting q at least n times, we notice that $\mathfrak{R}^q(s_0) = \bigcap_{n>k} \mathfrak{R}^q_{\geq n}(s_0)$. Thanks to equalities of Corollary 18, we can therefore compute that

$$\mathbb{P}\left(\mathfrak{R}_k^{q,e}(s_0) \mid \mathfrak{R}^q(s_0)\right) = \mathbb{P}\left(\mathfrak{R}_0^{q,e}(s_q)\right)$$

and that for every $k \neq k'$,

$$\mathbb{P}\left(\mathfrak{R}_k^{q,e}(s_0) \cap \mathfrak{R}_{k'}^{q,e}(s_0) \mid \mathfrak{R}^q(s_0)\right) = \mathbb{P}\left(\mathfrak{R}_0^{q,e}(s_q)\right)^2.$$

We deduce that the two events $\mathfrak{R}_k^{q,e}(s_0)$ and $\mathfrak{R}_{k'}^{q,e}(s_0)$ are conditionally independent given $\mathfrak{R}^q(s_0)$. ∎

b) Divergence of the series:

Lemma 14: $\sum_{k\in\mathbb{N}} \mathbb{P}(\mathfrak{R}_k^{q,e}(s_0) \mid \mathfrak{R}^q(s_0)) = +\infty$.

Proof: As in the previous lemma, we can deduce from equalities of Corollary 18 that

$$\mathbb{P}\left(\mathfrak{R}_k^{q,e}(s_0) \mid \mathfrak{R}^q(s_0)\right) = \mathbb{P}\left(\mathfrak{R}_0^{q,e}(s_q)\right).$$

However, as $q \in \mathcal{Q}'_e$, we know by definition that we have

$$\mathbb{P}(\mathfrak{R}_0^{q,e}(s_q)) > 0.$$

The result follows. ∎

c) Decomposition using basic sets: This section aims to prove Corollary 18 and so to complete the previous proofs.

Lemma 15: Let r be a memoryless region, $v \in r$, and $s = (\ell, v)$ a configuration of $\mathcal{A}$. Writing q for region-state (ℓ, r), we have for every sequence $(e_1, \ldots, e_n) \in E^n$,

$$\mathbb{P}(\pi(s, e_1 \ldots e_n)) = \mathbb{P}(\pi(s_q, e_1 \ldots e_n)).$$

Sketch of proof: We can prove a stronger result. We can show that for every pair $s = (\ell, v)$, $s' = (\ell, v')$ satisfying for every $x \in X$, $v(x) = v'(x)$ or $\min(v(x), v'(x)) > M$, we have for every sequence $(e_1, \ldots, e_n) \in E^n$,

$$\mathbb{P}(\pi(s, e_1 \ldots e_n)) = \mathbb{P}(\pi(s', e_1 \ldots e_n)).$$

Such pairs s and s' cannot be discerned by the automaton and for every $s \xrightarrow{t,e_1} s_t^{e_1}$ and $s' \xrightarrow{t,e_1} {s'}_t^{e_1}$, the pairs $s_t^{e_1}$ and ${s'}_t^{e_1}$ have still the same property. We can then show the result by induction on the length n of the sequence of edges. ∎

We define $\mathfrak{E}_q(s_0)$ the set of runs starting in s_0 and visiting q at least once.

Proposition 16: Let $q \in \mathcal{Q}'_e$. The following equalities hold true:

1) For every $n \geq 1$,
$$\mathbb{P}\left(\mathfrak{R}^q_{\geq n}(s_0)\right) = \mathbb{P}\left(\mathfrak{E}_q(s_0)\right) \cdot \mathbb{P}\left(\mathfrak{R}^q_{\geq 1}(s_q)\right)^{n-1}.$$
2) For every $1 \leq k < n$,
$$\mathbb{P}\left(\mathfrak{R}_k^{q,e}(s_0) \cap \mathfrak{R}^q_{\geq n}(s_0)\right) = \mathbb{P}\left(\mathfrak{E}_q(s_0)\right) \cdot \mathbb{P}\left(\mathfrak{R}^q_{\geq 1}(s_q)\right)^{n-2} \cdot \mathbb{P}\left(\mathfrak{R}^q_{\geq 1}(s_q) \cap \mathfrak{R}_0^{q,e}(s_q)\right).$$
3) For every $1 \leq k < k' < n$,
$$\mathbb{P}\left(\mathfrak{R}_k^{q,e}(s_0) \cap \mathfrak{R}_{k'}^{q,e}(s_0) \cap \mathfrak{R}^q_{\geq n}(s_0)\right) = \mathbb{P}\left(\mathfrak{E}_q(s_0)\right) \cdot \mathbb{P}\left(\mathfrak{R}^q_{\geq 1}(s_q)\right)^{n-3} \cdot \mathbb{P}\left(\mathfrak{R}^q_{\geq 1}(s_q) \cap \mathfrak{R}_0^{q,e}(s_q)\right)^2.$$

Sketch of proof: We proceed by decomposition. By example, for the first equality, we decompose the runs $\varrho \in \mathfrak{R}^q_{\geq n}(s_0)$ in $\varrho' \cdot \varrho''$ where ϱ' is a finite run such that $\mathsf{last}(\varrho') \in q$ (first visit) and $\varrho'' \in \mathfrak{R}^q_{\geq n-1}(\mathsf{last}(\varrho'))$. Moreover, $\varrho'' \in \mathfrak{R}^q_{\geq n-1}(\mathsf{last}(\varrho'))$ if and only if

$$\varrho'' \in \mathsf{Cyl}\left(\pi(\mathsf{last}(\varrho'), e_1, \ldots, e_k)\right)$$

where $e_1, \ldots, e_k$ are such that $\mathsf{target}(e_k) = q$, and $\#\{1 \leq i \leq k \mid \mathsf{target}(e_i) = q\} = n - 1$. Taking into account all these finite families $(e_1, \ldots, e_k)$, we can then prove our equality by induction on n thanks to Lemma 15. ∎

We can simplify equalities of the previous proposition thanks to the following lemma:

Lemma 17: Let $q \in \mathcal{Q}'_e$. We have $\mathbb{P}(\mathfrak{R}^q_{\geq 1}(s_q)) = 1$.

Proof: By contradiction, we assume that $\mathbb{P}(\mathfrak{R}^q_{\geq 1}(s_q)) = \alpha_0 < 1$. By Proposition 16, we thus have that:

$$\begin{aligned}\mathbb{P}(\mathfrak{R}^q(s_0)) &= \mathbb{P}\left(\bigcap_n \mathfrak{R}^q_{\geq n}(s_0)\right) \\ &= \lim_{n\to\infty} \mathbb{P}\left(\mathfrak{R}^q_{\geq n}(s_0)\right) \\ &= \lim_{n\to\infty} \mathbb{P}\left(\mathfrak{E}_q(s_0)\right) \cdot \mathbb{P}\left(\mathfrak{R}^q_{\geq 1}(s_q)\right)^{n-1} \\ &= \mathbb{P}\left(\mathfrak{E}_q(s_0)\right) \lim_{n\to\infty} (\alpha_0)^{n-1} = 0\end{aligned}$$

which contradicts the fact that $q \in \mathcal{Q}'_e$. ∎

Corollary 18: Let $q \in \mathcal{Q}'_e$. The following equalities hold true:

1) For every $n \geq 1$,
$$\mathbb{P}\left(\mathfrak{R}^q_{\geq n}(s_0)\right) = \mathbb{P}\left(\mathfrak{E}_q(s_0)\right).$$
2) For every $1 \leq k < n$,
$$\mathbb{P}\left(\mathfrak{R}_k^{q,e}(s_0) \cap \mathfrak{R}^q_{\geq n}(s_0)\right) = \mathbb{P}\left(\mathfrak{E}_q(s_0)\right) \cdot \mathbb{P}\left(\mathfrak{R}_0^{q,e}(s_q)\right).$$

3) For every $1 \leq k < k' < n$,

$$\mathbb{P}\left(\mathfrak{R}_k^{q,e}(s_0) \cap \mathfrak{R}_{k'}^{q,e}(s_0) \cap \mathfrak{R}_{\geq n}^{q}(s_0)\right) = \mathbb{P}\left(\mathfrak{E}_q(s_0)\right) \cdot \mathbb{P}\left(\mathfrak{R}_0^{q,e}(s_q)\right)^2.$$

IV. Applications

A. ω-regular properties

In the previous sections, we have reduced the almost-sure model-checking of an ω-regular property in a reactive stochastic timed automaton to the almost-sure model-checking of an ω-regular property in a finite Markov chain. From that reduction we can therefore infer a decision procedure and compute an upper bound for the almost-sure model-checking problem. More precisely, we have the following result.

Theorem 19: The almost-sure model-checking of reactive stochastic timed automata against Büchi, co-Büchi, Streett, Rabin, Muller or parity conditions is PSPACE-complete.

Proof: We first show the PSPACE upper bound. Let $\mathcal{A}$ be a reactive stochastic timed automaton and φ be a state-based ω-regular property. Note first that we cannot assume $\mathcal{A} = \mathsf{R}(\mathcal{A})$ since $\mathsf{R}(\mathcal{A})$ is exponential-size w.r.t. $\mathcal{A}$. Thus, for complexity analysis, it is important to start from $\mathcal{A}$. We have proven in the previous sections that:

$$\neg\Big(\mathcal{A} \bowtie \varphi\Big) \Leftrightarrow \neg\Big(\mathsf{R}(\mathcal{A}) \bowtie \widetilde{\varphi}\Big) \Leftrightarrow \mathbb{P}_{\mathsf{MC}(\mathcal{A})}(q_0 \models \neg\widetilde{\varphi}) > 0$$

where $\widetilde{\varphi}$ is the lifting of φ in $\mathsf{R}(\mathcal{A})$ (it is obviously ω-regular).

The size of $\mathsf{MC}(\mathcal{A})$ is that of $\mathsf{R}(\mathcal{A})$, which is exponential in the size of $\mathcal{A}$. However we will not construct $\mathsf{MC}(\mathcal{A})$, and we will proceed as in [27, Corollary 34] for guessing reachable BSCCs[8] that satisfy the property $\neg\widetilde{\varphi}$. Indeed in a given BSCC, almost-surely runs visit all states of the BSCC: the property $\neg\widetilde{\varphi}$ has therefore probability 0 or 1 in a BSCC. The property $\mathbb{P}_{\mathsf{MC}(\mathcal{A})}(q_0 \models \neg\widetilde{\varphi}) > 0$ holds iff there is a reachable BSCC which satisfies $\neg\widetilde{\varphi}$ with probability 1. Globally this can be done using a PSPACE decision procedure.

Hardness already holds for simple safety properties, and can be proven with a rather standard reduction from the halting problem of linarly-bounded Turing machines. ∎

B. Properties given as deterministic timed automata

We now extend the previous analysis to the almost-sure model-checking of properties given as deterministic timed automata. Let $\mathcal{A} = (L, \ell_0, X, E, \lambda, (\mu_\ell)_{\ell\in L}, (w_e)_{e\in E})$ be a labelled reactive stochastic timed automaton, that is, a reactive stochastic timed automaton with a labelling function $\lambda : L \to 2^{\mathsf{AP}}$, where AP is a finite set of atomic propositions. Let $\mathcal{B} = (\mathsf{S}, \mathsf{s}_0, \mathsf{X}, \mathsf{E}, \mathcal{F})$ be a deterministic (labelled) timed automaton, whose transitions are furthermore labelled by elements of 2^{AP} and with accepting condition given by an ω-regular condition $\mathcal{F}$. We assume w.l.o.g. that $\mathcal{B}$ is complete w.r.t. time[9] and that X and X are disjoint sets of clocks. We say that an infinite run $\varrho = (\ell_0, v_0) \xrightarrow{\tau_1,e_1} (\ell_1, v_1) \xrightarrow{\tau_2,e_2} \dots$ of $\mathcal{A}$ satisfies $\mathcal{B}$ whenever the (unique) infinite run $(\mathsf{s}_0, \mathbf{0}) \xrightarrow{\tau_1,\lambda(\ell_0)} (\mathsf{s}_1, \mathsf{v}_1) \xrightarrow{\tau_2,\lambda(\ell_1)} \dots$ is accepted by $\mathcal{B}$. In that case we write $\varrho \models \mathcal{B}$. The set of infinite runs of $\mathcal{A}$ satisfying the specification $\mathcal{B}$ is obviously measurable (for $\mathbb{P}_\mathcal{A}$). See Appendix **??**.

To solve the almost-sure model-checking problem for specifications given by $\mathcal{B}$, we define the product $\mathcal{A} \ltimes \mathcal{B}$ as the stochastic timed automaton $(\overline{L}, \overline{\ell}_0, X \cup \mathsf{X}, \overline{E}, (\overline{\mu_\ell})_{\overline{\ell}\in\overline{L}}, (\overline{w_{\overline{e}}})_{\overline{e}\in\overline{E}})$ where:

- $\overline{L} = L \times \mathsf{S}$, $\overline{\ell}_0 = (\ell_0, \mathsf{s}_0)$;
- $\overline{E}$ is made of the following edges: if $(\ell \xrightarrow{g,Y} \ell') \in E$ then for all $(\mathsf{s} \xrightarrow{\mathsf{g},\lambda(\ell),\mathsf{Y}} \mathsf{s}') \in \mathsf{E}$, there is an edge $((\ell, \mathsf{s}) \xrightarrow{g\wedge\mathsf{g},Y\cup\mathsf{Y}} (\ell', \mathsf{s}'))$ in $\overline{E}$;
- $\overline{\mu}_{(\ell,\mathsf{s})} = \mu_\ell$ for every $(\ell, \mathsf{s}) \in \overline{L}$, and $\overline{w}_{\overline{e}} = w_e$ for every edge $\overline{e} \in \overline{E}$ which comes from edge e.

Note that $\mathcal{A} \ltimes \mathcal{B}$ is reactive since $\mathcal{A}$ is reactive and $\mathcal{B}$ is complete. Note that any run in $\mathcal{A}$ has a unique image in $\mathcal{A} \ltimes \mathcal{B}$. We define the ω-regular property $\varphi_\mathcal{B}$ in $\mathcal{A} \ltimes \mathcal{B}$ as the lifting of $\mathcal{F}$ in $\mathcal{A} \ltimes \mathcal{B}$ (an infinite run in $\mathcal{A} \ltimes \mathcal{B}$ satisfies $\varphi_\mathcal{B}$ whenever its projection on $\mathcal{B}$ satisfies the accepting condition $\mathcal{F}$). As $\mathcal{F}$ is an ω-regular condition on states of $\mathcal{B}$, $\varphi_\mathcal{B}$ is a location-based ω-regular condition in $\mathcal{A} \ltimes \mathcal{B}$.

Following arguments similar to those of [27, Lemma 31], we get the following correspondence between $\mathcal{A}$ and $\mathcal{A} \ltimes \mathcal{B}$:

Lemma 20: $\mathbb{P}_\mathcal{A}(s_0 \models \mathcal{B}) = \mathbb{P}_{\mathcal{A}\ltimes\mathcal{B}}((\overline{\ell}_0, \mathbf{0}) \models \varphi_\mathcal{B})$.

Therefore, almost-sure model-checking in $\mathcal{A}$ against properties given as deterministic timed automata is reduced to almost-sure model-checking of a (location-based) ω-regular condition in the product automaton $\mathcal{A} \ltimes \mathcal{B}$. Thus, it can be done in polynomial space. Note that in the above, $\mathcal{B}$ can be untimed and can represent an LTL formula.

Corollary 21: Almost-sure model-checking of reactive stochastic timed automata against specifications given as deterministic timed automata with an ω-regular accepting condition is PSPACE-complete.

C. What about branching-time logics?

Model-checking the qualitative fragment of PCTL [31], that is, the restriction to probabilistic formulas involving only constants 0 and 1 is possible using a tableau-method, yielding an EXPTIME upper bound for the algorithm (the Markov chain $\mathsf{MC}(\mathcal{A})$ is exponential-size). We do not enter into the details here.

V. Conclusion

In this paper we have shown that we can solve the almost-sure model-checking problem against a large class of properties for stochastic timed automata with several clocks, provided they are reactive. An interesting property of reactive timed automata is that the probability of Zeno behaviours is 0, which is a desirable property of real systems. The main ingredient for the decidability is that these automata are *almost-sure fair*, which is a natural property satisfied by most stochastic processes, but which was unluckily not the case in the (too) general framework proposed in [11].

[8] BSCC stands for "bottom strongly-connected component".

[9] That is, for every location $\mathsf{s} \in \mathsf{S}$, for every clock valuation v, there is an outgoing transition from s that is enabled at v.

We believe this is a big step towards the modelling and verification of stochastic real-time systems based on automata. As suggested in Section II-D, stochastic timed automata are a natural model for representing time-dependent stochastic systems. Decidability of qualitative model-checking (with a reasonable complexity) therefore gives a solid basis for their use in a verification context.

Further work includes (approximate) quantitative verification. We think that the memoryless states could be used as checkpoints for decoupling the computation of probabilities, as reset-states and states with large value for the clock in [12]. As other challenges, the study of decision processes and stochastic games [13] could be well pursued, under the hypothesis that systems are reactive.

Acknowledgments: The first author is supported by the ANR project ImpRo (ANR-2010-BLAN-0317). The second author is supported by the ARC project (AUWB-2010–10/15-UMONS-3) and a grant from the the National Bank of Belgium. The fourth author is supported by an FRIA grant.

REFERENCES

[1] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[2] G. Behrmann, A. David, K. G. Larsen, J. Håkansson, P. Pettersson, W. Yi, and M. Hendriks, "Uppaal 4.0," in *Proc. 3rd International Conference on Quantitative Evaluation of Systems (QEST'06)*. IEEE Computer Society Press, 2006, pp. 125–126.

[3] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine, "Kronos: a model-checking tool for real-time systems," in *Proc. 10th International Conference on Computer Aided Verification (CAV'98)*, ser. Lecture Notes in Computer Science, vol. 1427. Springer, 1998, pp. 546–550.

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

[5] R. Alur, S. La Torre, and G. J. Pappas, "Optimal paths in weighted timed automata," in *Proc. 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, ser. Lecture Notes in Computer Science, vol. 2034. Springer, 2001, pp. 49–62.

[6] G. Behrmann, A. Fehnker, Th. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager, "Minimum-cost reachability for priced timed automata," in *Proc. 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, ser. Lecture Notes in Computer Science, vol. 2034. Springer, 2001, pp. 147–161.

[7] P. Bouyer, U. Fahrenberg, K. G. Larsen, and N. Markey, "Quantitative analysis of real-time systems using priced timed automata," *Communication of the ACM*, vol. 54, no. 9, pp. 78–87, 2011.

[8] M. Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston, "Verifying quantitative properties of continuous probabilistic timed automata," in *Proc. 11th International Conference on Concurrency Theory (CONCUR'00)*, ser. Lecture Notes in Computer Science, vol. 1877. Springer, 2000, pp. 123–137.

[9] ——, "Automatic verification of real-time systems with discrete probability distributions," *Theoretical Computer Science*, vol. 282, no. 1, pp. 101–150, 2002.

[10] C. Baier, N. Bertrand, P. Bouyer, Th. Brihaye, and M. Größer, "Probabilistic and topological semantics for timed automata," in *Proc. 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)*, ser. Lecture Notes in Computer Science, vol. 4855. Springer, 2007, pp. 179–191.

[11] ——, "Almost-sure model checking of infinite paths in one-clock timed automata," in *Proc. 23rd Annual Symposium on Logic in Computer Science (LICS'08)*. IEEE Computer Society Press, 2008, pp. 217–226.

[12] N. Bertrand, P. Bouyer, Th. Brihaye, and N. Markey, "Quantitative model-checking of one-clock timed automata under probabilistic semantics," in *Proc. 5th International Conference on Quantitative Evaluation of Systems (QEST'08)*. IEEE Computer Society Press, 2008.

[13] P. Bouyer and V. Forejt, "Reachability in stochastic timed games," in *Proc. 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, ser. Lecture Notes in Computer Science, vol. 5556. Springer, 2009, pp. 103–114.

[14] M. Stoelinga, "Fun with FireWire: A comparative study of formal verification methods applied to the IEEE 1394 root contention protocol," *Formal Aspects of Computing*, vol. 14, no. 3, pp. 328–337, 2003.

[15] A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton, "Model-checking continuous-time Markov chains," *ACM Transactions on Computational Logic*, vol. 1, no. 1, pp. 162–170, 2000.

[16] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Model-checking algorithms for continuous-time Markov chains," *IEEE Transactions on Software Engineering*, vol. 29, no. 7, pp. 524–541, 2003.

[17] S. Donatelli, S. Haddad, and J. Sproston, "Model checking timed and stochastic properties with CSL^{TA}," *IEEE Transactions on Software Engineering*, vol. 35, no. 2, pp. 224–240, 2009.

[18] L. Zhang, D. N. Jansen, F. Nielson, and H. Hermanns, "Automata-based CSL model checking," in *Proc. 38th International Colloquium on Automata, Languages and Programming (ICALP'11)*, ser. Lecture Notes in Computer Science, vol. 6756. Springer, 2011, pp. 271–282.

[19] T. Chen, T. Han, J.-P. Katoen, and A. Mereacre, "Model-checking of continuous-time Markov chains against timed automata specifications," *Logical Methods in Computer Science*, vol. 7, no. 1:12, pp. 1–34, 2011.

[20] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: verification of probabilistic real-time systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, ser. Lecture Notes in Computer Science, vol. 6806. Springer, 2011, pp. 585–591.

[21] R. Alur, C. Courcoubetis, and D. L. Dill, "Model-checking for probabilistic real-time systems," in *Proc. 18th International Colloquium on Automata, Languages and Programming (ICALP'91)*, ser. Lecture Notes in Computer Science, vol. 510. Springer, 1991, pp. 115–126.

[22] ——, "Verifying automata specifications of probabilistic real-time systems," in *Proc. REX Workshop on Real-Time: Theory in Practice*, ser. Lecture Notes in Computer Science, vol. 600. Springer, 1992, pp. 28–44.

[23] P. Bouyer, T. Brihaye, M. Jurdziński, and Q. Menet, "Almost-sure model-checking of reactive timed automata," Laboratoire Spécification & Vérification, ENS de Cachan, France, Research Report LSV-12-09, 2012.

[24] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[25] H. Hermanns, *Interactive Markov Chains: The Quest for Quantified Quality*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2428.

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

[27] C. Baier, N. Bertrand, P. Bouyer, Th. Brihaye, and M. Größer, "A probabilistic semantics for timed automata," Laboratoire Spécification & Vérification, ENS de Cachan, France, Research Report LSV-08-13, 2008.

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

[29] J. C. Oxtoby, "The Banach-Mazur game and Banach category theorem," *Annals of Mathematical Studies*, vol. 39, pp. 159–163, 1957, contributions to the Theory of Games, volume 3.

[30] P. Billingsley, *Probability and measure*, 3rd ed., ser. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1995.

[31] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects of Computing*, vol. 6, no. 5, pp. 512–535, 1994.