Compositional Design of Stochastic Timed Automata*

Patricia Bouyer[1], Thomas Brihaye[2], Pierre Carlier[1,2], and Quentin Menet[2]

[1] LSV, CNRS, ENS Cachan, Université Paris-Saclay, France
[2] Université de Mons, Belgium

Abstract. In this paper, we study the model of stochastic timed automata and we target the definition of adequate composition operators that will allow a compositional approach to the design of stochastic systems with hard real-time constraints. This paper achieves the first step towards that goal. Firstly, we define a parallel composition operator that (we prove) corresponds to the interleaving semantics for that model; we give conditions over probability distributions, which ensure that the operator is well-defined; and we exhibit problematic behaviours when this condition is not satisfied. We furthermore identify a large and natural subclass which is closed under parallel composition. Secondly, we define a bisimulation notion which naturally extends that for continuous-time Markov chains. Finally, we importantly show that the defined bisimulation is a congruence w.r.t. the parallel composition, which is an expected property for a proper modular approach to system design.

1 Introduction

Compositional design and compositional verification are two crucial aspects of the development of computerised systems for which correctness needs to be guaranteed or quantified. It is indeed convenient and natural to model separately each component of a system and model their interaction, and it is easier and probably less error-prone than to model at once the complete system.

In the last twenty years a huge effort has been made to design expressive models, with the aim to faithfully represent computerised systems. This is for instance the case of systems with real-time constraints for which the model of timed automata [1,2] is successfully used. Many applications like communication protocols require models integrating both real-time constraints and randomised aspects (see e.g. [25]), which requires the development of specific models. Recently, a model of stochastic timed automata (STA) has been proposed as a natural extension of timed automata with stochastic delays and stochastic edge choices (see [8] for a survey of the results so far concerning this model). Advantages of the STA model are twofold: (i) it is based on the well-understood and powerful

* The first and the third authors are supported by ERC project EQualIS. The second author is partly supported by FP7-EU project Cassting. The fourth author was a postdoctoral researcher at the Belgian National Fund for Scientific Research (FNRS).

model of timed automata, allowing to express hard real-time constraints like deadlines (unlike for the widely used model of *continuous-time Markov chains* (CTMCs in short)); (ii) it enjoys nice decidability properties (see [8,9]). On the other hand, there is no obvious way of designing in a compositional manner a complex system using this model.

In this paper we are inspired by the approach of [24], and we target the definition of (parallel) composition operators allowing for a component-based modelling framework for STA. This paper achieves the first steps towards that goal:

1. We define a parallel composition operator that (we prove) corresponds to the interleaving semantics for that model; we give conditions over families of distributions over delays, which ensure that the operator is well-defined; we exhibit problematic behaviours when this condition is not satisfied. We furthermore identify a class of such well-behaving STA that is closed under parallel composition. Note that this class of well-behaving systems encompasses the class of CTMCs.
2. We define a bisimulation notion which naturally extends that for CTMCs [5,6,17], and we importantly show that the bisimulation is a congruence w.r.t. parallel composition; this is an expected property for a proper modular approach to system design.

The next step will be to extend the current composition operator with some synchronisation between components. For CTMCs, this has required much effort over the years to come up with a satisfactory solution, yielding for instance the model of *interactive Markov chains* (IMCs) [21,22]. We believe we will benefit a lot from this solution and plan to follow a similar approach for STA; we leave it as further work (the current work focuses on races between components and establishes all useful properties at the level of STA).

Related works We do not list all works concerned with the verification of stochastic real-time systems, but will focus on those interested in compositional design. The first natural related work is that on interactive Markov chains (IMCs in short) [21,22], which extend CTMCs with interaction, and for which compositional verification methods have been investigated [13,23]. However in this model, only soft real-time constraints can be evaluated (that is, they may not be always satisfied by the system, but their likelihood is then quantified), and the model cannot evolve differently, depending on constraints over clocks. Our ultimate goal is to extend the elegant approach of IMCs to a model based on timed automata.

Other related approaches are based on process algebras (note that originally IMCs presented as a process algebra as well [21]). There have been several proposals, among which the IGSMP calculus [12], whose semantics is given as generalised semi-Markov processes (GSMPs); and the stochastic process algebra ♤ [15,16], whose semantics is given as ♤-stochastic timed automata (we write ♤-STA). Our model very much compares to the latter, so we will briefly describe it. In such a system, when a clock variable is activated, it is sampled according to a predefined distribution, and then it acts as a countdown timer: when time elapses,

the clock variables decrease down to 0. Transitions can be fired once all clocks specified on the transition have reached value 0. First notice that both STA and ♤-STA allow to express hard real-time constraints, e.g. strict deadlines to be satisfied by the system (which is not the case of CTMCs or IMCs). Then the ♤-STA model is at the basis of several modelling languages like Modest [10] and comes with several notions of bisimulations with nice congruence properties, and with a complete equational theory. It is interesting to mention as well that ♤-STA allow for infinitely many states and clock variables, whereas STA do not (they have been defined on top of timed automata, with desirable decidability properties in mind). Similarly to ♤-STA, STA extend (finite-state and finite-variable) GSMPs,[1] but for different reasons: ♤-STA allows for fixed-delay events and non-determinism, whereas STA allows for more intricate timing constraints and branchings.[2] Finally, it is worth mentioning the modelling language Modest [10], whose semantics is given as a very general notion of stochastic timed automata (we call them Modest-STA), which comes with an interesting tool suite [19,20], and which encompasses all the models we have mentioned. STA in general, and the subclass that is closed under parallel composition while enjoying decidability properties, can be viewed as a fragment of Modest-STA.

The full version of this work and detailed proofs are given in [11].

2 Stochastic Timed Automata

In this section, we recall the notion of *timed automaton* [2], and that of *stochastic timed automaton* [8]. Let $X = \{x_1, \ldots, x_n\}$ be a finite set of real-valued variables called *clocks*. A *clock valuation* over X is a mapping $\nu : X \to \mathbb{R}_+$ where $\mathbb{R}_+$ is the set of nonnegative real numbers. We write $\mathbb{R}_+^X$ for the set of clock valuations over X. If $\nu \in \mathbb{R}_+^X$, we write ν_i for $\nu(x_i)$ and we then denote ν by $(\nu_1, \ldots, \nu_n)$. If $\tau \in \mathbb{R}_+$, we write $\nu + \tau$ for the clock valuation defined by $(\nu_1 + \tau, \ldots, \nu_n + \tau)$. If $Y \in 2^X$ (the power set of X), $[Y \leftarrow 0]\nu$ is the valuation that assigns to x, 0 if $x \in Y$ and $\nu(x)$ otherwise. A *guard*[3] over X is a finite conjunction of expressions of the form $x_i \sim c$ where $c \in \mathbb{N}$ and $\sim \in \{<, >\}$. We denote by $\mathcal{G}(X)$ the set of guards over X. We write $\nu \models g$ if ν satisfies g, which is defined in a natural way.

Definition 1. *A timed automaton (TA in short) is a tuple* $\mathcal{A} = (L, L_0, X, E, \mathsf{AP}, \mathcal{L})$ *where: (i)* L *is a finite set of locations, (ii)* $L_0 \subseteq L$ *is a set of initial locations, (iii)* X *is a finite set of clocks, (iv)* $E \subseteq L \times \mathcal{G}(X) \times 2^X \times L$ *is a finite set of edges, (v)* AP *is a set of atomic propositions and (vi)* $\mathcal{L} : L \to 2^{\mathsf{AP}}$ *is a labelling function.*

The semantics of a TA is a labelled timed transition system $T_{\mathcal{A}} = (Q, Q_0, \mathbb{R}_+ \times E, \to, \mathsf{AP}, \mathcal{L})$ where $Q = L \times \mathbb{R}_+^X$ is the set of states, $Q_0 = L_0 \times \mathbf{0}_X$ is the set

[1] This can be seen using the residual-time semantics given in [18,14].

[2] Somehow, the clock behaviour in GSMPs and in ♤-STA is that of countdown timers (which can be seen as event-predicting clocks of [3]), which is not as rich as general clocks in standard timed automata.

[3] We restrict to open guards for technical reasons due to stochastic aspects.

of initial states (valuation $\mathbf{0}_X$ assigns 0 to each clock), $\mathcal{L} : Q \to 2^{\mathsf{AP}}$ labels each state $q = (l, \nu) \in Q$ by $\mathcal{L}(l)$ and $\to \subseteq Q \times (\mathbb{R}_+ \times E) \times Q$ is the transition relation defined as follows: if $e = (l, g, Y, l') \in E$ and $\tau \in \mathbb{R}_+$, then we have $(l, \nu) \xrightarrow{\tau, e} (l', \nu')$ if $(\nu + \tau) \models g$ and $\nu' = [Y \leftarrow 0](\nu + \tau)$. If $q = (\ell, \nu)$, for every $\tau \geq 0$, $q + \tau$ denotes $(\ell, \nu + \tau)$. A *finite* (resp. *infinite*) *run* ρ is a finite (resp. infinite) sequence $\rho = q_1 \xrightarrow{\tau_1, e_1} q_2 \xrightarrow{\tau_2, e_2} \ldots$. Given $q \in Q$, we write $\mathrm{Runs}(\mathcal{A}, q)$ for the set of infinite runs in $\mathcal{A}$ from q. Given $q \in Q$ and $e \in E$ we define $I(q, e) = \{\tau \in \mathbb{R}_+ \mid \exists q' \in Q \text{ s.t. } q \xrightarrow{\tau, e} q'\}$ and $I(q) = \bigcup_{e \in E} I(q, e)$.

We now define the notion of *stochastic timed automaton* [8], by equipping every state of a TA with probabity measures over both delays and edges.

Definition 2. *A stochastic timed automaton (STA in short) is a tuple* $\mathcal{A} = (L, L_0, X, E, AP, \mathcal{L}, (\mu_q, p_q)_{q \in L \times \mathbb{R}_+^X})$ *where* $(L, L_0, X, E, AP, \mathcal{L})$ *is a timed automaton and for every* $q = (l, \nu) \in L \times \mathbb{R}_+^X$,

(i) μ_q *is a probability distribution over* $I(q)$ *and* p_q *is a probability distribution over* E *such that for each* $e = (l, g, Y, l') \in E$, $p_q(e) > 0$ *iff* $\nu \models g$,
(ii) μ_q *is equivalent to the restriction of the Lebesgue measure on* $I(q)$,[4] *and*
(iii) for each edge e, *the function* $p_{q+\bullet}(e) : \mathbb{R}_+ \to [0, 1]$ *that assigns to each* $t \geq 0$ *the value* $p_{q+t}(e)$, *is measurable.*

We fix $\mathcal{A}$ a STA, with the notations of the definition. We let $Q = L \times \mathbb{R}_+^X$ be the set of states of $\mathcal{A}$, and pick $q \in Q$. We aim at defining a probability distribution $\mathbb{P}_{\mathcal{A}}$ over $\mathrm{Runs}(\mathcal{A}, q)$. Let $e_1, \ldots, e_k$ be edges of $\mathcal{A}$, and $\mathcal{C} \subseteq \mathbb{R}_+^k$ be a Borel set. The *(constrained) symbolic path* starting from q and determined by $e_1, \ldots, e_k$ and $\mathcal{C}$ is the following set of finite runs: $\pi_{\mathcal{C}}(q, e_1, \ldots, e_k) = \{\rho = q \xrightarrow{\tau_1, e_1} q_1 \cdots \xrightarrow{\tau_k, e_k} q_k \mid (\tau_1, \ldots, \tau_k) \in \mathcal{C}\}$. Given a symbolic path π, we define the cylinder generated by π as the subset $\mathrm{Cyl}(\pi)$ of $\mathrm{Runs}(\mathcal{A}, q)$ containing all runs ρ with a prefix ρ' in π.

We inductively define a measure over the set of symbolic paths as follows:

$$\mathbb{P}_{\mathcal{A}}(\pi_{\mathcal{C}}(q, e_1, \ldots, e_k)) = \int_{t_1 \in I(q, e_1)} p_{q+t_1}(e_1)\, \mathbb{P}_{\mathcal{A}}(\pi_{\mathcal{C}_{[\tau_1/t_1]}}(q_{t_1}, e_2, \ldots, e_k))\, \mathrm{d}\mu_q(t_1),$$

where for every $t_1 \geq 0$, q_{t_1} is such that $q \xrightarrow{t_1, e_1} q_{t_1}$ and $\mathcal{C}_{[\tau_1/t_1]}$ replaces variable τ_1 by t_1 in $\mathcal{C}$; we initialise with $\mathbb{P}_{\mathcal{A}}(\pi(q)) = 1$. The formula for $\mathbb{P}_{\mathcal{A}}$ relies on the fact that the probability of taking transition e_1 at time t_1 coincides with the probability of waiting t_1 time units and then choosing e_1 among the enabled transitions, i.e. $p_{q+t_1}(e_1)\, \mathrm{d}\mu_q(t_1)$. Now, one can extend $\mathbb{P}_{\mathcal{A}}$ to the cylinders by $\mathbb{P}_{\mathcal{A}}(\mathrm{Cyl}(\pi)) = \mathbb{P}_{\mathcal{A}}(\pi)$, where π is a symbolic path. Using some extension theorem as Carathéodory's theorem, we can extend $\mathbb{P}_{\mathcal{A}}$ in a unique way to the σ-algebra generated by the cylinders starting in q, which we denote $\Omega_{\mathcal{A}}^q$.

Proposition 1 ([8]). *Let* $\mathcal{A} = (L, L_0, X, E, \mathsf{AP}, \mathcal{L}, (\mu_q, p_q)_{q \in L \times \mathbb{R}_+^X})$ *be a STA. For every state* $q \in Q$, $\mathbb{P}_{\mathcal{A}}$ *is a probability measure over* $(\mathrm{Runs}(\mathcal{A}, q), \Omega_{\mathcal{A}}^q)$.

[4] Two measures μ and ν on the same measurable space are equivalent whenever for every measurable set A, $\mu(A) > 0$ iff $\nu(A) > 0$.

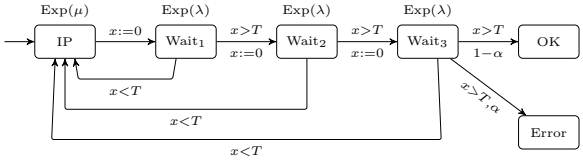

Fig. 1. The IPv4 Zeroconf STA.

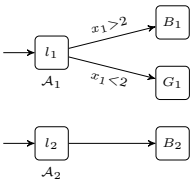

Fig. 2. $\mathcal{A}_1 \notin$ CSTA.

Remark 1. Among others, the set of Zeno runs is measurable in $\Omega^q_{\mathcal{A}}$;[5] writing $\mathcal{C}_{M,k}$ for $\{(\tau_1, \ldots, \tau_k) \in \mathbb{R}^k_+ \mid \tau_1 + \ldots + \tau_k \leq M\}$ it is indeed expressible as follows:

$$\bigcup_{M \in \mathbb{N}} \bigcap_{k \in \mathbb{N}_0} \bigcup_{(e_1,\ldots,e_k) \in E^k} \mathrm{Cyl}(\pi_{\mathcal{C}_{M,k}}(q, e_1, \ldots, e_k)).$$

Remark 2. A CTMC can be viewed as a STA with trivial guards on transitions and exponential distributions over delays.

We now give an example of STA.

Example 1. We model the IPv4 Zeroconf protocol using STA as done in [8] (see Figure 1). This protocol aims at configuring IP addresses in a local network of appliances. When a new appliance is plugged, it selects an IP address at random, and broadcasts several probe messages to the network to know whether this address is already used or not. If it receives in a bounded delay an answer from the network informing that the IP is already used, then a new IP address is chosen. It may be the case that messages get lost, in which case there is an error. In [7], a simple model for the IPv4 Zeroconf protocol is given as a discrete-time Markov chain, which abstracts away timing constraints. In Figure 1, we model the protocol as a STA with a single clock x, and exponential distributions (of parameters μ and λ) and this allows us to explicitly express the delay bound.

Discussion on the model. STA have been defined and studied in a series of papers from 2007, with a complete journal version published as [8]. They can be used for modelling systems with stochastic aspects and real-time constraints (they are based on the standard model of timed automata [2] and extend the model of CTMCs) and are amenable to automatic verification. The class of *almost-surely fair STA*[6] is of particular interest. Indeed:

Theorem 1 ([8]). *The almost-sure model-checking problem is decidable for the class of almost-surely fair STA, with regards to ω-regular properties or properties given as deterministic timed automata.*

[5] We recall that a run $\rho = q \xrightarrow{\tau_1,e_1} q_1 \xrightarrow{\tau_2,e_2} \ldots$ is *Zeno* if $\sum_{i \geq 1} \tau_i < +\infty$.

[6] A STA is said almost-surely fair whenever $\mathbb{P}_{\mathcal{A}}(\mathsf{fair}) = 1$, where a run is fair if and only if (roughly speaking) any edge enabled infinitely often is taken infinitely often.

There exists surprisingly simple examples of STA which are not almost-surely fair (see for example [8, Figure 9]), but large classes of STA have been identified in [8], that are almost-surely fair (they include single-clock STA and (weak-)reactive STA). Deciding whether a STA is almost-surely fair is an open problem

The approach adopted so far for modelling and verifying is monolithic. We target modular design of STA and describe a class of STA in which composition can safely be applied.

3 Parallel Composition of Stochastic Timed Automata

Compositional design is desirable for building computerised systems. Inspired by the approach of [24], we first define a parallel composition operator for STA, which corresponds to an interleaving semantics. This operator involves complex behaviours that are due to races between components. We therefore give conditions under which STA can be safely composed.

Remark 3. As already mentioned earlier, we focus here on an interleaving parallel composition operator between STA, and study the races between components. Extension to a parallel composition operator with some synchronisation is part of our future work, and we plan to adopt the idea of interactive Markov chains [21,22], which extend CTMCs with interactive actions, for the purpose of synchronisation.

3.1 Definition of the parallel composition

We consider two STA $\mathcal{A}_i = (L_i, L_0^{(i)}, X_i, E_i, \mathsf{AP}_i, \mathcal{L}_i, (\mu_q^{(i)}, p_q^{(i)})_{q \in L_i \times \mathbb{R}_+^{X_i}})$ for $i = 1, 2$ with $X_1 \cap X_2 = \emptyset$, and we first recall the standard (interleaving) parallel composition for the underlying TA. It is the TA $(L, L_0, X, E, \mathsf{AP}, \mathcal{L})$ where $L = L_1 \times L_2$, $L_0 = L_0^{(1)} \times L_0^{(2)}$, $X = X_1 \cup X_2$, $\mathsf{AP} = \mathsf{AP}_1 \cup \mathsf{AP}_2$, $\mathcal{L} : L \to 2^{\mathsf{AP}}$ is such that $\mathcal{L}((l_1, l_2)) = \mathcal{L}_1(l_1) \cup \mathcal{L}_2(l_2)$ and where $E = E_{1,\bullet} \cup E_{\bullet,2}$ with $E_{1,\bullet} = \{((l_1, l_2), g, Y, (l_1', l_2)) \mid (l_1, g, Y, l_1') \in E_1,\ l_2 \in L_2\}$.

Back to the STA, the parallel composition $\mathcal{A}_1 \parallel \mathcal{A}_2$ has as underlying TA the one above; it remains to equip each state $q = (q_1, q_2) \in Q_1 \times Q_2$ with probability distributions over both delays and edges, with the following constraints:

- distributions over delays from state (q_1, q_2) should reflect a *race* between the two components $\mathcal{A}_1$ and $\mathcal{A}_2$ from respectively states q_1 and q_2;
- distributions over edges should be state-based (or memoryless), that is, should not depend on how long has been waited before taking that edge, or which other actions have been done meanwhile by other components;
- globally, the product-automaton should correspond to the interleaving of $\mathcal{A}_1$ and $\mathcal{A}_2$, which we express as follows: given a property φ_1 that only concerns $\mathcal{A}_1$ and a property φ_2 that only concerns $\mathcal{A}_2$, $\mathbb{P}_{\mathcal{A}_1 \parallel \mathcal{A}_2}(\varphi_1 \wedge \varphi_2) = \mathbb{P}_{\mathcal{A}_1}(\varphi_1) \cdot \mathbb{P}_{\mathcal{A}_2}(\varphi_2)$.

Example 2 will illustrate the intricacy of getting these conditions satisfied.

Let $\mathcal{A}$ be a STA and let $q = (l, \nu) \in Q$ be a state of $\mathcal{A}$. We write f_q for the density function of μ_q w.r.t. the Lebesgue measure. We write F_q for the cumulative function associated to f_q.

We now define a first class of STA, called CSTA, which is suitable to define a parallel composition. We say that a STA $\mathcal{A}$ is in CSTA if:

(A) for every state q of $\mathcal{A}$, the density function associated with μ_q, denoted by f_q, is continuous everywhere on $\mathbb{R}_+$ except in a finite number of points, and
(B) the family of probability distributions $(\mu_q)_{q \in Q}$ is *weakly-memoryless*, i.e. for every $t, t' \geq 0$, $\mathbb{P}_{\mathcal{A}}(\mathbb{X}_q \geq t + t' \mid \mathbb{X}_q \geq t) = \mathbb{P}_{\mathcal{A}}(\mathbb{X}_{q+t} \geq t')$, where $\mathbb{X}_q$ (resp. $\mathbb{X}_{q+t}$) is a random variable with density function f_q (resp. f_{q+t}).

This second condition is a consistency condition between states which belong to the same 'time-elapsing fiber', that is, sets of the form $F = \{q + t \mid t \in \mathbb{R} \text{ and } q + t \in Q\}$. Indeed, $\mathbb{X}_q$ (resp. $\mathbb{X}_{q+t}$) represents the delay after which we leave state q (resp. $q+t$) via an edge. Hence if q_0 is the minimal (for time-elapsing) element of F, then for every $q = q_0 + t \in F$, the law of $\mathbb{X}_q$ has to be equal to the law of $\mathbb{X}_{q_0}$ conditioned by the fact that t time units have already passed. The distribution in q_0 can be taken arbitrary (satisfying condition (A)), and distributions for $q \in F$ can then be inferred.

Condition (B) can equivalently be written as: for every $t, t' \geq 0$,

$$f_q(t + t') = (1 - F_q(t)) f_{q+t}(t') \tag{1}$$

Remark 4. Let q_0 be an initial element of a fiber, we can check that for instance,

- if $I(q_0)$ is a bounded subset of $\mathbb{R}_+$ and if μ_{q_0} is a uniform distribution over $I(q_0)$, then for every $t \in \mathbb{R}_+$, (B) imposes that μ_{q_0+t} is also uniform over $I(q_0 + t)$;
- similarly, if $I(q_0) = \mathbb{R}_+$, and if μ_{q_0} is an exponential distribution with parameter λ (denoted $\mathrm{Exp}(\lambda)$), then for every $t \in \mathbb{R}_+$, (B) imposes that μ_{q_0+t} is also an $\mathrm{Exp}(\lambda)$-distribution. This corresponds to the classical memoryless property assumed in CTMCs.

We can now explain how to build the probability distributions associated with a state $q = (q_1, q_2)$ of $\mathcal{A}_1 \parallel \mathcal{A}_2$. Since we leave state $q = (q_1, q_2)$ as soon as we leave q_1 or q_2, we naturally define the distribution over the delays from q as the minimum of the distributions over delays from q_1 and q_2. Under hypothesis (A) for the distributions from q_1 and q_2, one can show that the density function f_q for the minimum satisfies $f_q(t) = f_{q_1}(t)(1 - F_{q_2}(t)) + f_{q_2}(t)(1 - F_{q_1}(t))$ almost-surely for every $t \geq 0$ (w.r.t. the Lebesgue measure).

In order to define the probability distribution p_q over the enabled edges in q, one could consider that from state q, both systems $\mathcal{A}_1$ and $\mathcal{A}_2$ are in a race to win the next edge, i.e. $\mathcal{A}_1$ wins the race if the first edge taken from q is in E_1. Hence, given $t \in I(q)$, and an edge $e \in E_1$ enabled in $q + t$, one would like that $p_{q+t}(e) = w_q^1(t) p_{q_1+t}(e)$ where $w_q^1(t)$ is the probability that, starting from

q, $\mathcal{A}_1$ wins the race knowing that it was won after a delay of t time units. This can be formalized, and under hypothesis (A) for f_{q_1} and f_{q_2}, we can show that if $f_q(t) \neq 0$, then $w_q^1(t) = \frac{f_{q_1}(t)(1-F_{q_2}(t))}{f_q(t)}$ almost-surely.

Definition 3. *Let* $\mathcal{A}_i = (L_i, L_0^{(i)}, X_i, E_i, AP_i, \mathcal{L}_i, (\mu_q^{(i)}, p_q^{(i)})_{q \in L_i \times \mathbb{R}_+^{X_i}})$ *for* $i = 1, 2$ *be two STA. We say that* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *are composable if* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *are in* CSTA *and* $X_1 \cap X_2 = \emptyset$*.In that case, we define the parallel composition of* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *as the STA* $\mathcal{A}_1 \parallel \mathcal{A}_2 = (L, L_0, X, E, AP, \mathcal{L}, (\mu_q, p_q)_{q \in L \times \mathbb{R}_+^X})$*, where for any state* $q = (q_1, q_2)$ *of* $\mathcal{A}_1 \parallel \mathcal{A}_2$*,*

(i) $(L, L_0, X, E, AP, \mathcal{L})$ *is the composition of the underlying TA* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*,*
(ii) μ_q *is defined by its density function* $f_q = f_{q_1}(1 - F_{q_2}) + f_{q_2}(1 - F_{q_1})$*, and*
(iii) for any $t \in I(q)$*,* p_{q+t} *is defined as follows:*

$$p_{q+t}(e) = \mathbb{1}_{E_1}(e) w_q^1(t) p_{q_1+t}(e) + \mathbb{1}_{E_2}(e) w_q^2(t) p_{q_2+t}(e)$$

for every $e \in E$*, where* $w_q^i = \frac{f_{q_i}}{f_q}(1 - F_{q_{3-i}})$ *on* $I(q)$*, for* $i = 1, 2$*.*

3.2 Properties of the parallel composition

We are now ready to prove that this parallel composition operator satisfies all the expected properties. We assume the notations of Definition 3. First:

Lemma 1. *The distributions* μ_q *and* p_q *are well-defined, and the STA* $\mathcal{A}_1 \parallel \mathcal{A}_2$ *belongs to the class* CSTA*.*

We now give an example of a family of probability measures that do not satisfy hypothesis (B), which yields undesirable properties in the parallel composition.

Example 2 (Counter-example for condition (B)). We consider the single-clock STA $\mathcal{A}_1$ depicted in Figure 2 (page 5). We assume μ_{q_1} is an exponential distribution of parameter λ_1 (resp. λ_1') if $q_1 = (l_1, \nu_1)$ with $\nu_1 < 1$ (resp. $\nu_1 \geq 1$), and with $\lambda_1 \neq \lambda_1'$. Then for each $\nu_1 \in [0, 1[$, μ_{q_1} does not satisfy hypothesis (B). We then compose $\mathcal{A}_1$ with the STA $\mathcal{A}_2$. Each state $q_2 = (l_2, \nu_2)$ is equipped with an exponential distribution of parameter $\lambda_2 = \lambda_1'$ over the delays. It can be shown that the probability to reach B_1 in $\mathcal{A}_1$ corresponds to the probability to reach (B_1, B_2) in $\mathcal{A}_1 \parallel \mathcal{A}_2$ iff $\ln(\lambda_1) - \ln(\lambda_2) = \lambda_1 - \lambda_2$, which is not true in general.

Example 3. In order to illustrate the notion of composition, we composed two independent copies of the STA modelling the IPv4 Zeroconf protocol (see Example 1). Part of the composed STA is depicted in Figure 3.

It remains to identify when the parallel composition really coincides with an interleaving semantics. This is in general not true, as already shown in Example 2 (which does not satisfy Condition (B)), and witnessed further by Example 4 below (which satisfies both conditions (A) and (B)).

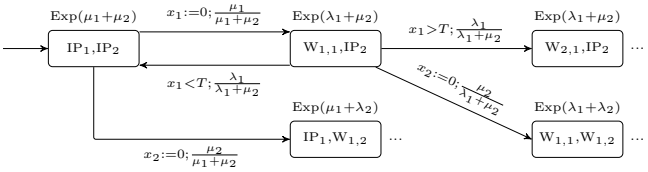

Fig. 3. The product of two STA modelling the IPv4 Zeroconf

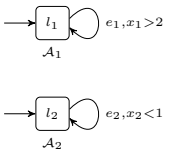

Fig. 4. $\mathcal{A}_2$ is Zeno

Example 4. We consider the STA $\mathcal{A}_1$ and $\mathcal{A}_2$ of Figure 4, equipped resp. with an Exp(λ)-distribution and a uniform distribution. Let $q = (q_1, q_2)$ be a state of $\mathcal{A}_1 \parallel \mathcal{A}_2$, with $q_i = (l_i, 0)$. One can easily check that $\mathbb{P}_{\mathcal{A}_1 \parallel \mathcal{A}_2}(q \to^* \xrightarrow{e_1}) = 0$ while $\mathbb{P}_{\mathcal{A}_1}(\mathrm{Cyl}(\pi(q_1, e_1))) = 1$ which contradicts the independence property we expect. One can notice that $\mathcal{A}_2$ is Zeno with probability 1.

Hence we define a subclass CSTA* of CSTA; $\mathcal{A} \in$ CSTA will be in CSTA* if:

(C) $\mathcal{A}$ is almost-surely non-Zeno.

Remark 5. Hypothesis (C) is not too restrictive since Zeno runs can be seen as faulty behaviours (they perform infinitely many actions in a finite amount of time, which is not realistic). We will see that hypothesis (C) is sufficient (together with (A) and (B)) to show that the parallel composition really coincides with an interleaving semantics. Note that condition (C) can be decided in various subclasses of STA [8].

We give some more notations. Let $\mathcal{A}$ be a STA and let φ be a property for $\mathcal{A}$. Given a state q, we say that φ is measurable from q if the set of runs starting from q satisfying φ is in $\Omega^q_{\mathcal{A}}$; we write this set $\{q \models \varphi\}$. Now let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two composable STA. For $i = 1, 2$, we write ι_i for the natural projection of $\mathrm{Runs}(\mathcal{A}_1 \parallel \mathcal{A}_2, (q_1, q_2))$ onto $\mathrm{Runs}(\mathcal{A}_i, q_i)$, and given a measurable property φ_i in $\mathcal{A}_i$ from q_i, we write $\{(q_1, q_2) \models \widetilde{\varphi}_i\}$ for the set $\iota_i^{-1}(\{q_i \models \varphi_i\})$. The following theorem states that the defined parallel composition is indeed interleaving.

Theorem 2. *Let $\mathcal{A}_1, \mathcal{A}_2 \in$ CSTA* be composable. Then $\mathcal{A}_1 \parallel \mathcal{A}_2 \in$ CSTA*. Moreover, for every state $q = (q_1, q_2)$ of $\mathcal{A}_1 \parallel \mathcal{A}_2$, for every properties φ_1 measurable in $\mathcal{A}_1$ from q_1 and φ_2 measurable in $\mathcal{A}_2$ from q_2, we have*

$$\mathbb{P}_{\mathcal{A}_1 \parallel \mathcal{A}_2}(\{q \models \widetilde{\varphi}_1\} \cap \{q \models \widetilde{\varphi}_2\}) = \mathbb{P}_{\mathcal{A}_1}(\{q_1 \models \varphi_1\}) \cdot \mathbb{P}_{\mathcal{A}_2}(\{q_2 \models \varphi_2\}). \quad (2)$$

Proof (Sketch). Given $\mathcal{A}_1$ and $\mathcal{A}_2$ in CSTA*, thanks to Lemma 1, it suffices to prove that $\mathcal{A}_1 \parallel \mathcal{A}_2$ is almost-surely non-Zeno. This will be ensured by (2) and the fact that *non-Zenoness* is a measurable property.

The important first step to prove (2) consists in showing that, given an edge e_1 of $\mathcal{A}_1$, the probability in $\mathcal{A}_1 \parallel \mathcal{A}_2$ that e_1 is the first edge from $\mathcal{A}_1$ (with possibly edges from $\mathcal{A}_2$ taken before) performed from $q = (q_1, q_2)$ in a given set of delays Δ corresponds to the probability in $\mathcal{A}_1$ that e_1 is the first edge performed from q_1 in the same set of delays Δ. In order to do so, hypothesis (B) is crucial. The rest of the proof is long and technical but does not contain major difficulties. □

Remark 6. Note that an almost-surely non-Zeno STA $\mathcal{A}$ equipped with uniform or exponential distributions such that it satisfies conditions (A) and (B) (i.e. as in Remark 4), it holds that $\mathcal{A}$ is in CSTA*. As said before, we have large classes of STA that are almost-surely fair. For (weak-)reactive STA, it holds that they are almost-surely non-Zeno. Equipping them with uniform or exponential distributions as in Remark 4) make them also composable.

4 Bisimulation and Congruence

In this section, we define a notion of bisimulation for STA which naturally extends that for CTMCs [4,6,17]. We importantly show that the defined bisimulation is a congruence w.r.t. parallel composition: this means that, in a complex system, a component can be replaced by an equivalent one without affecting the global behaviour of the system.

4.1 Bisimulation

To define a bisimulation relation between STA, we are inspired by the approach of [17], which considers continuous-time Markov processes (CTMPs) – CTMPs generalize CTMCs to general continuous state-spaces; this definition of bisimulation that is given for CTMPs can be adapted to our context (note however that STA cannot be seen as particular CTMPs).

We first define some notions. A subset $P \subseteq \mathbb{R}^n$ is a *polyhedral set* if it is defined by a (finite) boolean combination of constraints of the form $A_1 x \leq b_1$ or $A_2 x < b_2$, where $x = (x_1, \ldots, x_n)$ is a variable, $A_1 \in \mathbb{R}^{m_1 \times n}$, $b_1 \in \mathbb{R}^{m_1}$, $A_2 \in \mathbb{R}^{m_2 \times n}$ and $b_2 \in \mathbb{R}^{m_2}$.

Let $\mathcal{A}$ be a STA, Q be its set of states, and $P(Q) = \{\cup_{l \in L}\{l\} \times C_l \mid \forall l \in L,\ C_l$ polyhedral set of $\mathbb{R}^n_+\}$ where n is the number of clocks of $\mathcal{A}$. The set $P(Q)$ is a proper subset of the Borel σ-algebra over $L \times \mathbb{R}^n_+$, which is closed by projection (contrary to the Borel σ-algebra). We then define the *closure of* $\mathcal{R}$ w.r.t. polyhedral sets, and we write $\mathrm{pcl}(\mathcal{R})$ as the following set $\mathrm{pcl}(\mathcal{R}) = \{A \in P(Q) \mid (a \in A \wedge a\mathcal{R}b) \Rightarrow b \in A\}$. One can notice that $\mathrm{pcl}(\mathcal{R})$ corresponds to the set of all polyhedral unions of equivalence classes. Given two equivalence relations $\mathcal{R}$ and $\mathcal{R}'$ over S we say that $\mathcal{R}'$ is *coarser* than $\mathcal{R}$ or that $\mathcal{R}$ is *finer* than $\mathcal{R}'$ if $\mathcal{R} \subseteq \mathcal{R}'$.

Definition 4. *Let $\mathcal{A} = (L, L_0, X, E, AP, \mathcal{L}, (\mu_q, p_q)_{q \in L \times \mathbb{R}^X_+})$ be a STA. An equivalence relation $\mathcal{R}$ over $Q = L \times \mathbb{R}^X_+$ is a bisimulation for $\mathcal{A}$ if for all $q, q' \in Q$ with $q\mathcal{R}q'$: (i) $\mathcal{L}(q) = \mathcal{L}(q')$, and (ii) for every $I \in \mathcal{B}(\mathbb{R}_+)$, for every $C \in \mathrm{pcl}(\mathcal{R})$,*

$$\mathbb{P}_{\mathcal{A}}\big(\{q \models \xrightarrow{I,E} C\}\big) = \mathbb{P}_{\mathcal{A}}\big(\{q' \models \xrightarrow{I,E} C\}\big),$$

where $\{q \models \xrightarrow{I,E} C\}$ stands for $\{\rho \in \mathrm{Runs}(\mathcal{A}, q) \mid \exists \tau \in I,\ \exists e \in E,\ \rho = q \xrightarrow{\tau,e} q_1 \to \cdots \wedge q_1 \in C\}$. States q and q' are bisimilar (written $q \sim q'$) if there is a bisimulation that contains (q, q').

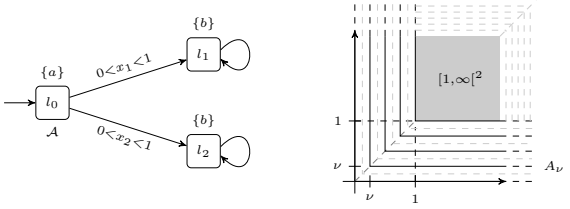

Fig. 5. A simple example for bisimulation.

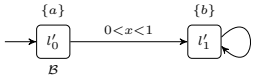

Fig. 6. $\mathcal{B}$ is bisimilar to $\mathcal{A}$.

Given $q \in Q$, $I \in \mathcal{B}(\mathbb{R}_+)$ and $C \in \mathrm{pcl}(\mathcal{R})$ the value $\mathbb{P}_{\mathcal{A}}\big(\{q \models \xrightarrow{I,E} C\}\big)$ can be expressed:

$$\mathbb{P}_{\mathcal{A}}\big(\{q \models \xrightarrow{I,E} C\}\big) = \int_{t\in I} P_{q+t}(C) f_q(t)\,\mathrm{d}t$$

where the value $P_{q+t}(C)$ corresponds to the probability to reach instantaneously C from state $q+t$. Formally: $P_{q+t}(C) = \sum_{l'\in L}\sum_{e\in E_{l'}} p_{q+t}(e)\mathbb{1}_{\mathcal{C}_{l'}(e,\nu)}(t)$ for each $t \geq 0$ and each $C \in \mathrm{pcl}(\mathcal{R})$, where, given $l' \in L$, $E_{l'}$ is the set of edges with target l', and given $e = (l, g, Y, l')$, $\mathcal{C}_{l'}(e,\nu) = \{t \in \mathbb{R}_+ \mid [Y \leftarrow 0](\nu + t) \in C_{l'}\}$. It can be shown that for every $t \geq 0$, P_{q+t} is a probability measure over Q.

Also, given a STA $\mathcal{A}$, one can show that $\sim$ is the coarsest bisimulation for $\mathcal{A}$.

The above natural definition enjoys the following very nice characterization, which shows that our definition is conservative w.r.t. bisimulation over CTMCs [4,6].

Proposition 2. *Let $\mathcal{A}$ be a STA and let $\mathcal{R}$ be a bisimulation for $\mathcal{A}$. Then for all $q, q' \in Q$, $q\mathcal{R}q'$ if and only if (i) $\mathcal{L}(q) = \mathcal{L}(q')$, (ii) $\mu_q = \mu_{q'}$, and (iii) for every $C \in \mathrm{pcl}(\mathcal{R})$, $P_{q+t}(C) = P_{q'+t}(C)$ almost-surely for every $t \geq 0$.*

Proof (Sketch). Point (i) is obvious, and points (ii) and (iii) come from the fact that $q\mathcal{R}q'$ if for each $C \in \mathrm{pcl}(\mathcal{R})$ and for each $I \in \mathcal{B}(\mathbb{R}_+)$,

$$\int_{t\in I} P_{q+t}(C) f_q(t)\,\mathrm{d}t = \int_{t\in I} P_{q'+t}(C) f_{q'}(t)\,\mathrm{d}t.$$

With $C = L \times \mathcal{B}(\mathbb{R}_+^n)$, where n is the number of clocks, we get that $P_{q+t}(C) = 1$ and thus $f_q = f_{q'}$ almost-surely, i.e. $\mu_q = \mu'_q$. It can then be easily shown that point (iii) holds. □

We now illustrate the notion of bisimulation on a simple example.

Example 5. Let us consider the simple STA $\mathcal{A}$ with two clocks on Figure 5. We assume exponential distributions with parameter λ for every state at l_1 or l_2, and from a state of the form $q = (l_0, (\nu_1, \nu_2))$ with $\nu_1 < 1$ or $\nu_2 < 1$, $I(q) = [0, 1 - \min(\nu_1, \nu_2)[$ and so we can equip q with a uniform distribution on the interval $I(q)$ for the delays.

The coarsest bisimulation $\sim$ can easily be computed and is shown on the right part of Figure 5: at location l_0, it is described by the following equivalence classes, for each $\nu \in [0, 1[$: $A_\nu = \{l_0\} \times \big(\{(\nu_1, \nu) \mid \nu_1 \geq \nu\} \cup \{(\nu, \nu_2) \mid \nu_2 \geq \nu\}\big)$.

We extend the previous notion of bisimulation to two STA in a standard way (see [7]), by considering the union of the two STA, and a bisimulation relation between the initial states. If $\mathcal{A}_1$ and $\mathcal{A}_2$ are two STA, we write $\mathcal{A}_1 \sim \mathcal{A}_2$ when the two STA are bisimilar.

Example 6. Let us consider the one-clock STA $\mathcal{B}$ (Figure 6). Assuming that we have the same probability distributions as STA $\mathcal{A}$ of Figure 5, it can be easily established that $\mathcal{B} \sim \mathcal{A}$ by noticing that for each $\nu \in [0, 1[$, (l'_0, ν) is bisimilar to each state of A_ν.

4.2 Congruence

One of the main objectives of defining behavioural equivalences is to aim at modular design and proof of correctness. This is only possible if bisimulation is a *congruence w.r.t. parallel composition*, that is, if $\mathcal{A}_1 \sim \mathcal{A}_2$, then for every $\mathcal{B}$, $\mathcal{A}_1 \parallel \mathcal{B} \sim \mathcal{A}_2 \parallel \mathcal{B}$. We first prove the following natural lemma which is a key point for proving the congruence of the bisimulation w.r.t. parallel composition. Though very intuitive, the result is surprisingly quite technical to prove.

Lemma 2. *Let $\mathcal{A}, \mathcal{B} \in$ CSTA* with sets of states resp. Q_A and Q_B. If $\mathcal{R}$ is a bisimulation for $\mathcal{A}$ then the equivalence relation $\mathcal{R}'$ over $Q_A \times Q_B$ defined by $\mathcal{R}' = \{((q_1, q), (q_2, q)) \mid q_1 \mathcal{R} q_2$ and $q \in Q_B\}$, is a bisimulation for $\mathcal{A} \parallel \mathcal{B}$.*

We can now state the main result of this section:

Theorem 3. *Bisimulation is a congruence w.r.t. parallel composition. That is: if $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{B}$ are three STA in* CSTA**, if $\mathcal{A}_1 \sim \mathcal{A}_2$ then $\mathcal{A}_1 \parallel \mathcal{B} \sim \mathcal{A}_2 \parallel \mathcal{B}$.*

5 Conclusion

In this paper we have described a formal framework for compositional design of stochastic timed automata. We have established properties that should be satisfied by distributions over delays for well-defined parallel composition between components. We have proposed a natural notion of bisimulation and proven that it is a congruence w.r.t. parallel composition. We have also identified a subclass of STA which is closed under parallel composition.

We plan to extend our current work to so-called *interactive* STA (following [21,22]): the idea will be to add non-guarded interactive synchronizing events which take priority over delays when they are enabled. We hope that a parallel composition with synchronisation can be nicely defined in that setting, and that the model will enjoy nice properties as is the case in this paper.

There are many other plans for the future:

- Following the approach of [17,5], we would like to give a logical characterization of the bisimulation using (a subset of) CSL;
- We would like to be able, given a STA, to compute a small quotient automaton that would allow reduce the size of the system;

– All algorithms that have been developed so far for analyzing STA require a unique STA describing the system under analysis; we target the development of compositional verification (or approximation) methods, as it is done for instance for interactive Markov chains [13,23]. We would then like to see how this performs in practice.

References

1. R. Alur and D. Dill. Automata for modeling real-time systems. In *Proc. 17th Int. Coll. Automata, Languages and Programming (ICALP'90)*, volume 443 of *LNCS*, pages 322–335. Springer, 1990.
2. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
3. R. Alur, L. Fix, and T.A. Henzinger. A determinizable class of timed automata. In *Proc. 6th International Conference on Computer Aided Verification (CAV'94)*, volume 818 of *LNCS*, pages 1–13. Springer, 1994.
4. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Engineering*, 29(7):524–541, 2003.
5. C. Baier, H. Hermanns, J.-P. Katoen, and V. Wolf. Comparative branching-time semantics for Markov chains. *Information and Computation*, 200:149–214, 2005.
6. C. Baier, H. Hermanns, J.-P. Katoen, and V. Wolf. Bisimulation and simulation relations for Markov chains. In *Proc. Workshop Essays on Algebraic Process Calculi*, volume 162 of *ENTCS*, pages 73–78, 2006.
7. C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
8. N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, Ch. Baier, M. Größer, and M. Jurdziński. Stochastic timed automata. *Logical Methods in Computer Science*, 10(4):1–73, 2014.
9. N. Bertrand, P. Bouyer, Th. Brihaye, and N. Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics. In *Proc. 5th Int. Conf. Quantitative Evaluation of Systems (QEST'08)*. IEEE Comp. Soc. Press, 2008.
10. H. Bohnenkamp, P D'Argenio, H. Hermanns, and J.-P. Katoen. MODEST: A compositional modeling formalism for hard and softly timed systems. *IEEE Trans. Software Engineering*, 32(10):812–830, 2006.
11. P. Bouyer, T. Brihaye, P. Carlier, and Q. Menet. Compositional design of stochastic timed automata. Research Report LSV-15-06, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2015. 51 pages.
12. M. Bravetti and R. Gorrieri. The theory of interactive generalized semi-Markov processes. *Theoretical Computer Science*, 282(1):5–32, 2002.
13. T. Brázdil, H. Hermanns, J. Krcál, J. Kretínský, and V. Rehák. Verification of open interactive Markov chains. In *Proc. 31sth Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS'12)*, volume 18 of *LIPIcs*, pages 474–485. Springer, 2012.
14. T. Brázdil, J. Krčál, J. Křetínský, and V. Řehák. Fixed-delay events in generalized semi-Markov processes revisited. In *Proc. 22nd Int. Conf. Concurrency Theory (CONCUR'11)*, volume 6901 of *LNCS*, pages 140–155. Springer, 2011.
15. P. D'Argenio and J.-P. Katoen. A theory of stochastic systems Part I: Stochastic automata. *Information and Computation*, 203(1):1–38, 2005.

16. P. D'Argenio and J.-P. Katoen. A theory of stochastic systems Part II: Process algebra. *Information and Computation*, 203(1):39–74, 2005.
17. J. Desharnais and P. Panangaden. Continuous stochastic logic characterizes bisimulation of continuous-time Markov processes. *Journal of Logic and Algebraic Programming*, 56:99–115, 2003.
18. P.W. Glynn. A GSMP formalism for discrete event systems. *Proc. of the IEEE*, 77(1):14–23, 1989.
19. A. Hartmanns. Modest – A unified language for quantitative models. In *Proc. 2012 Forum on Specification and Design Languages (FDL'12)*, pages 44–51. IEEE Comp. Soc. Press, 2012.
20. A. Hartmanns and H. Hermanns. The Modest toolset: An integrated environment for quantitative modelling and verification. In *Proc. 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 8413 of *LNCS*, pages 593–598. Springer, 2014.
21. H. Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *LNCS*. Springer, 2002.
22. H. Hermanns and J.-P. Katoen. The how and why of interactive Markov chains. In *Proc. 8th Int. Symp. Formal Methods for Components and Objects (FMCO'09)*, volume 6286 of *LNCS*, pages 311–337. Springer, 2009.
23. H. Hermanns and J. Krcál, J. anf Kretínský. Compositional verification and optimization of interactive markov chains. In *Proc. 24th Int. Conf. Concurrency Theory (CONCUR'13)*, volume 8052 of *LNCS*, pages 364–379. Springer, 2013.
24. H. Hermanns and L. Zhang. From concurrency models to numbers – Performance and dependability. In *Software and Systems Safety – Specification and Verification*, volume 30 of *NATO Science for Peace and Security Series*, pages 182–210. IOS Press, 2011.
25. M. Stoelinga. Fun with FireWire: A comparative study of formal verification methods applied to the IEEE 1394 root contention protocol. *Formal Aspects of Computing*, 14(3):328–337, 2003.