Expressiveness and Conciseness of Timed Automata for the Verification of Stochastic Models

Susanna Donatelli[1](✉) and Serge Haddad[2]

[1] Dipartimento di Informatica, Università di Torino, Turin, Italy
donatelli@di.unito.it
[2] LSV, ENS Paris-Saclay, CNRS, Inria, Université Paris-Saclay, Cachan, France
haddad@lsv.fr

Abstract. Timed Automata are a well-known formalism for specifying timed behaviours. In this paper we are concerned with Timed Automata for the specification of timed behaviour of Continuous Time Markov Chains (CTMC), as used in the stochastic temporal logic CSL$^{\mathrm{TA}}$. A timed path formula of CSL$^{\mathrm{TA}}$ is specified by a Deterministic Timed Automaton (DTA) that features two kinds of transitions: *synchronizing* transitions (triggered by CTMC transitions) and *autonomous* transitions (triggered when a clock reaches a given threshold). Other definitions of CSL$^{\mathrm{TA}}$ are based on DTAs that do not include autonomous transitions. This raises the natural question: do autonomous transitions enhance expressiveness and/or conciseness of DTAs? We prove that this is the case and we provide a syntactical characterization of DTAs for which autonomous transitions do not add expressive power, but allow one to define exponentially more concise DTAs.

1 Introduction

Stochastic logics like CSL [5] allow one to express assertions about the probability of timed executions of Continuous Time Markov Chains (CTMC). In CSL, model executions (typically called "paths") are specified by two operators: timed neXt and timed Until. CSL has been extended in several ways to include action names (name of the events in the paths) and path properties specified using regular expressions leading to asCSL [6], or rewards, leading to CSRL [7]. Note that asCSL can specify rather complex path behaviour, expressed by regular expressions, but the timing requirements cannot be mixed within these expressions. GCSRL [14] is an extension of CSRL for model checking of CTMC generated by Generalized Stochastic Petri nets (GSPN) [1] taking into account both stochastic and immediate events.

Automata with time constraints have been used to specify path-based performance indices [16] for Stochastic Activity Networks [15], while hybrid automata have been used to define rather complex forms of passage of time [2] for GSPN, as well as generic performance properties [9] that are estimated using simulation. The use of a Deterministic Timed Automaton (DTA) in the stochastic

A. Leporati et al. (Eds.): LATA 2020, LNCS 12038, pp. 170–183, 2020.
https://doi.org/10.1007/978-3-030-40608-0_11

logic CSL^{TA} [12] allows to specify paths in terms of state propositions and action names associated to CTMC states and transitions (respectively) and in terms of the timed behaviour of *portions of the paths*. The CTMC actions are the input symbols for the DTA, and two types of transitions are distinguished: *synchronizing* transitions that read the input symbols of the CTMC, and *autonomous* transitions, that are taken by the DTA when the clock reaches some threshold, with priority over synchronizing ones. The determinism requirement ensures that the synchronized product of the DTA and the CTMC is still a stochastic process as all sources of non-determinism are eliminated. CSL^{TA} strictly includes [12] CSL and asCSL. Various extensions of CSL^{TA} have been presented in the literature. DTA with multiple clocks have been used for defining an extension of CSL^{TA} [10, 13] but autonomous transitions are not allowed. In this paper we concentrate on single-clock CSL^{TA} with autonomous transitions, as in the original definition of CSL^{TA}. Indeed the single-clock limitation is a necessary requirement to reduce the CSL^{TA} model-checking problem to the (steady-state) solution of a Markov Regenerative Process, which is the largest class of stochastic processes for which we can compute an exact numerical solution, supported by efficient solution tools [3, 4]. The single-clock setting allows also to investigate whether the definition of CSL^{TA} in [10, 13], once limited to a single clock, is equivalent to the original definition of CSL^{TA} (introduced in [12]).

Paper Contributions. This paper addresses two research questions. The first one (Sect. 3) is *whether the presence of autonomous transitions enhances the expressiveness of DTAs* both in terms of timed languages (qualitative comparison) and in terms of probability of accepting the random path of a CTMC (quantitative comparison). We establish that autonomous transitions do enhance expressiveness. Given that eliminating autonomous transitions from a DTA is not always feasible, the second question (Sect. 4) is *which are the uses of autonomous transitions that can be emulated by DTA w/o autonomous transitions.* We have identified a hierarchy of subclasses of DTA in which the presence of autonomous transitions does not extend expressiveness (and autonomous transitions can therefore be eliminated), but that exponentially improves the DTA size. Only the most interesting proofs and properties have been included in this paper. Missing proofs and the full set of properties can be found in [11].

2 Context and Definitions

Although our motivations rely on the acceptance of paths of CTMCs featuring atomic propositions that label states and actions that label transitions, we set our work in the general context of acceptance of timed paths, where the $i+1$-th state of a timed path is identified by v_i (we count indices from 0), the boolean evaluation of the atomic propositions in that state. δ_i indicates a delay, or a sojourn time in state i, and τ_i indicates the time elapsed until exiting state i. A timed path leaves state v_i with action a_i after a sojourn time in the state equal to δ_i. The elapsed time can be computed as: $\tau_i = \delta_i + \tau_{i-1}$, with $\tau_{-1} = 0$.

Definition 1 (Timed Path). *Given a set* AP *of atomic propositions and a set* Act *of actions, a* timed (infinite) path *is a sequence* $(v_0, \delta_0) \xrightarrow{a_0} (v_1, \delta_1) \xrightarrow{a_1} \cdots (v_i, \delta_i) \xrightarrow{a_i} \cdots$ *such that for all* $i \in \mathbb{N} : v_i \in \{\top, \bot\}^{AP}, a_i \in Act, \delta_i \in \mathbb{R}_{\geqslant 0}$.

Example 1 ***(Timed Path).*** In writing timed paths we indicate functions v_i as the set of elements in AP that evaluate to $\top$. Given $AP = \{p, q\}$ and $Act = \{a, b, c\}$, a timed path $(\{p, q\}, 0.5) \xrightarrow{a} (\{q\}, 1.3) \xrightarrow{b} \cdots$, is interpreted as the system staying in a state that satisfies $p \wedge q$ in the time interval $[0, 0.5[$, at time 0.5 action a takes place and the system moves to a state that satisfies $\neg p \wedge q$, stays there for 1.3 time units and then action b takes place (at the global time $\tau = 1.8$).

DTA definition includes a clock x and two types of constraints: boundary ones, $\mathsf{BoundC} = \{x = \alpha, \alpha \in \mathbb{N}\}$ and inner ones, $\mathsf{InC} = \{\alpha \bowtie x \bowtie' \beta\}$, with $\bowtie, \bowtie' \in \{<, \leqslant, \}, \alpha \in \mathbb{N}$, and $\beta \in \mathbb{N} \cup \{\infty\}$. In the sequel, C is the largest time constant occurring in a DTA. Before formally defining the syntax and semantic of a DTA (Definitions 2, 3 and 4), let us introduce its main ingredients. During the execution of a stochastic discrete event system (e.g. a Markov chain) that can be represented by a timed path, one manages (1) an index i of the timed path (2) a location, say ℓ, is matched with the current state of the path indexed by i, and (3) a delay $\delta \leqslant \delta_i$ until the next state change from i to $i + 1$. The function Λ mapping the set of locations to the set of boolean expressions over atomic propositions, $\mathcal{B}_{AP}$, restricts the possible matchings since the valuation v_i must satisfy the formula $\Lambda(\ell)$. This matching evolves in three ways depending on the delay δ, elapsed until the next transition $(v_i, \delta_i) \xrightarrow{a_i} (v_{i+1}, \delta_{i+1})$ of the path.

- Either after some delay $\delta' \leqslant \delta$, there is an outgoing *autonomous transition* from ℓ whose boundary condition (say $x = \alpha$) is satisfied and such that v_i fulfills $\Lambda(\ell')$ where ℓ' is the target location of the transition. Then ℓ' is matched with i, delay δ becomes $\delta - \delta'$, the clock x is increased by δ' and the index i is unchanged.
- Else if there is a *synchronizing transition* outgoing from ℓ such that (1) after time δ has elapsed its inner condition (say $\alpha \bowtie x \bowtie' \beta$) is satisfied, (2) the action a_i belongs to the subset of actions associated with the synchronizing transition, and (3) v_{i+1} satisfies $\Lambda(\ell')$ where ℓ' is the target location of the transition. Then ℓ' is matched with $i + 1$, the new delay δ is set to δ_{i+1}, the clock x is either increased by δ or reset depending on the transition, and the index becomes $i + 1$.
- Otherwise there is no possible matching and the timed path is rejected by the DTA.

In the first two cases above, when $\ell' = \ell_f$, the *final location*, the timed path is accepted by the DTA whatever its future. This is ensured due to $\Lambda(\ell_f) = \top$ and the existence of the unique (looping) synchronizing transition from ℓ_f with no timing and action conditions. Observe that the synchronization may last forever without visiting ℓ_f: in this case the timed path is rejected.

Furthermore the synchronization of the stochastic system with the DTA should not introduce non determinism. So (1) the formulas associated with the *initial locations* are mutually exclusive, (2) synchronizing transitions outgoing from the same location are never simultaneously enabled, (3) autonomous transitions outgoing from the same location are never simultaneously enabled, and (4) autonomous transitions have priority over synchronizing transitions.

Definition 2 (DTA). *A* single-clock Deterministic Timed Automaton with autonomous transitions *is defined by a tuple* $\mathcal{A} = \langle L, \Lambda, L_0, \ell_f, AP, Synch, Aut\rangle$ *where* L *is a finite set of* locations, $L_0 \subseteq L$ *is the set of* initial locations, $\ell_f \in L$ *is the* final location, $\Lambda : L \rightarrow \mathcal{B}_{AP}$ *is a function that assigns to each location a boolean expression over the set of propositions* AP, $Synch \subseteq L \times \mathsf{InC} \times 2^{Act} \times \{\varnothing, \downarrow\} \times L$ *is the set of* synchronizing transitions, *and* $Aut \subseteq L \times \mathsf{BoundC} \times \sharp \times \{\varnothing, \downarrow\} \times L$ *is the set of* autonomous transitions, *with* $E = Synch \cup Aut$. $\ell \xrightarrow{\gamma, B, r} \ell'$ *denotes the transition* $(\ell, \gamma, B, r, \ell')$.
Furthermore $\mathcal{A}$ *fulfills the following conditions.*

- **Initial determinism.** $\forall \ell, \ell' \in L_0, \Lambda(l) \wedge \Lambda(l') \Leftrightarrow \bot$.
- **Determinism on actions.** $\forall B, B' \subseteq Act\, s.t.\, B \cap B' \neq \varnothing, \forall \ell, \ell', \ell'' \in L$, *if* $\ell \xrightarrow{\gamma, B, r} \ell'$ *and* $\ell \xrightarrow{\gamma', B', r'} \ell''$ *then* $\Lambda(\ell') \wedge \Lambda(\ell'') \Leftrightarrow \bot$ *or* $\gamma \wedge \gamma' \Leftrightarrow \bot$.
- **Determinism on autonomous transitions.** $\forall \ell, \ell', \ell'' \in L$, *if* $\ell \xrightarrow{x=\alpha, \sharp, r} \ell'$ *and* $\ell \xrightarrow{x=\alpha', \sharp, r'} \ell''$ *then* $\Lambda(\ell') \wedge \Lambda(\ell'') \Leftrightarrow \bot$ *or* $\alpha \neq \alpha'$.
- **Conditions on the final location** ℓ_f**.** $\Lambda(\ell_f) = \top$ *and* $(\ell_f, \top, Act, \varnothing, \ell_f) \in$ *Synch.*

Given a clock constraint γ and a clock valuation $\bar{x}$, $\bar{x} \models \gamma$ denotes the satisfaction of γ by $\bar{x}$. Similarly given a boolean formula φ and a valuation of atomic propositions v, $v \models \varphi$ denotes the satisfaction of φ by v.

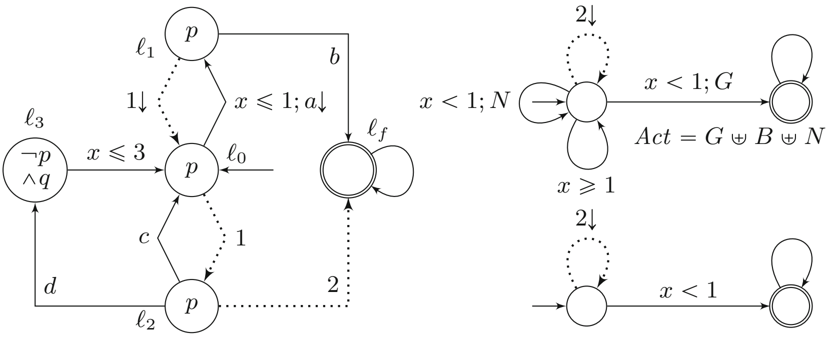

Fig. 1. Some examples of DTA.

Example 2 ***(DTA).*** Figure 1, left, shows a DTA with five locations: $\ell_0, \ell_1, \ell_2, \ell_3$ and ℓ_f. There is a single initial location, ℓ_0. Autonomous transitions are depicted as dotted arcs, while synchronizing are depicted as solid arcs. For readability we omit: (1) the symbol $\sharp$ on autonomous transitions; (2) the set r when there is no reset; (3) Act if a transition accepts all actions; (4) trivially true guards (like $x \geqslant 0$) and boolean conditions; (5) the name x of the clock in $x = \alpha$ guards. As a result an autonomous transition is depicted as either $l \xrightarrow{\alpha,\downarrow} l'$, as between ℓ_1 and ℓ_0, or as $l \xrightarrow{\alpha} l'$, as between ℓ_0 and ℓ_2. We informally write "a transition with reset" or "a transition without reset" to indicate the condition $r = \downarrow$ and $r = \varnothing$ respectively. The arc from ℓ_0 to ℓ_1 represents a synchronizing transition with a clock reset. The arc from ℓ_0 to ℓ_2 represents an autonomous transition to be taken when the clock is equal to 1, with no clock reset. Boolean expression of locations are: p, associated with ℓ_0, ℓ_1, ℓ_2 and $(\neg p \wedge q)$, associated with ℓ_3.

Let us describe a possible run of this DTA. At time 0.5, it goes from ℓ_0 to ℓ_1 by performing action a and resets x. Then at time 1.5, it autonomously comes back to location ℓ_0 and clock x is again reset. Then it autonomously goes to ℓ_2 at time 2.5 and later to ℓ_f at time 3.5. While irrelevant, x has current value 2.

Definition 3 (Run of $\mathcal{A}$). *A* run *of a DTA $\mathcal{A}$ is a sequence: $(\ell_0, v_0, \bar{x}_0, \delta_0) \xrightarrow{\gamma_0,B_0,r_0} (\ell_1, v_1, \bar{x}_1, \delta_1) \cdots (\ell_i, v_i, \bar{x}_i, \delta_i) \xrightarrow{\gamma_i,B_i,r_i} \cdots$ such that for all $i \in \mathbb{N}$: $\ell_i \in L, l_0 \in L_0, v_i \in \{\top, \bot\}^{AP}, \delta_i \in \mathbb{R}_{\geqslant 0}$:*

$$\ell_i \xrightarrow{\gamma_i,B_i,r_i} \ell_{i+1} \in E,\ v_i \models \Lambda(\ell_i),\ \bar{x}_i + \delta_i \models \gamma_i,\ \bar{x}_{i+1} = \begin{cases} 0 & \textit{if } r = \downarrow \\ \bar{x}_i + \delta_i & \textit{otherwise} \end{cases}$$

To enforce priority of autonomous transitions,

let $\bar{x}_\sharp = min\{\alpha \mid \exists \ell_i \xrightarrow{x=\alpha,\sharp,r} \ell \in E \wedge \bar{x}_i \leqslant \alpha \wedge v_i \models \Lambda(\ell)\}$ $(min(\varnothing) = \infty)$
If $B_i = \sharp$ then $\bar{x}_i + \delta_i = \bar{x}_\sharp$ and $v_{i+1} = v_i$ else $\bar{x}_i + \delta_i < x_\sharp$.

A run is therefore a path in the DTA where the visited locations are coupled with a valuation of propositions, a clock value and a delay in a consistent way w.r.t. the DTA.

Example 3 ***(DTA run).*** Given that v is described in terms of the subset of AP that evaluate to $\top$, a run for the DTA of Fig. 1, left, is: $0{:}(\ell_0, \{p\}, \bar{x}_0 = 0.0, \delta_0 = 0.2) \xrightarrow{x\leqslant 1,\{a\},\downarrow} 1{:}(\ell_1, \{p, q\}, 0.0, 1.0) \xrightarrow{x=1,\sharp,\downarrow} 2{:}(\ell_0, \{p, q\}, 0.0, 1.0) \xrightarrow{x=1,\sharp,\varnothing} 3{:}(\ell_2, \{p\}, 1.0, 1.0) \xrightarrow{x=2,\sharp,\varnothing} 4{:}(\ell_f, \{p\}, 2.0, 3.1) \xrightarrow{x\geqslant 0,Act,\varnothing} 5{:}(\ell_f, \{q\}, 5.1, 0.5) \xrightarrow{x\geqslant 0,Act,\varnothing} 6{:}(\ell_f, \{q\}, 5.6, \delta) \cdots$

A timed path σ is recognized by a run ρ of $\mathcal{A}$ such that the occurrences of the actions in σ are matched by the synchronizing transitions in ρ. This requires to define a mapping to match the points in the paths in which synchronizing transitions take place. This can be done by identifying a strictly increasing mapping for the indices of the timed path σ to the subset of the indices of the run ρ that correspond to a synchronizing transition. Note that, due to determinism, if such a run exists, it is unique.

Definition 4 (Path recognized by $\mathcal{A}$ and $\mathcal{L}(\mathcal{A})$). *Let* $\sigma = (v_0, \delta_0) \xrightarrow{a_0} (v_1, \delta_1) \xrightarrow{a_1} \cdots (v_i, \delta_i) \xrightarrow{a_i} \cdots$ *be a timed path and* $\rho = (\ell_0, v'_0, \bar{x}_0, \delta'_0) \xrightarrow{\gamma_0, B_0, r_0} \cdots (\ell_i, v'_i, \bar{x}_i, \delta'_i) \xrightarrow{\gamma_i, B_i, r_i} \cdots$ *be a run of a DTA* $\mathcal{A}$*. Then* σ *is recognized by* ρ *if there is a strictly increasing mapping* $\kappa : \mathbb{N} \to \mathbb{N}$ *(extended to* $\kappa(-1) = -1$*), such that for all* $i \in \mathbb{N}$

- $a_i \in B_{\kappa(i)}$ *and* $\delta_i = \sum_{\kappa(i-1)<h\leqslant\kappa(i)} \delta'_h$
- $\forall h,\ \kappa(i-1) < h \leqslant \kappa(i) \Rightarrow v'_h = v_i$ *and* $h \notin \kappa(\mathbb{N}) \Rightarrow B_h = \sharp$

A timed path σ *is accepted by* $\mathcal{A}$ *if* σ *is recognized by a run* ρ *and* ρ *visits* ℓ_f*. The language* $\mathcal{L}(\mathcal{A})$ *of* $\mathcal{A}$ *is the set of the timed paths* σ *accepted by* $\mathcal{A}$*.*

Example 4 ***(Path recognized by a run).*** A timed path $\sigma = 0{:}(p, 0.2) \xrightarrow{a} 1{:}(\{p, q\}, 6.1) \xrightarrow{b} 2{:}(q, 0.5) \xrightarrow{d} 3\ {:}(p, \delta) \cdots$ is recognized by the run of Example 3 with mapping κ: $\kappa(0) = 0, \kappa(1) = 4$, $\kappa(2) = 5, \kappa(3) = 6, \ldots$. The run visits ℓ_f and the path is accepted.

We consider timed paths generated by a CTMC with state properties and actions.

Definition 5 (CTMC representation). *A* continuous time Markov chain *with state and action labels is represented by a tuple* $\mathcal{M} = \langle S, s_0, Act, AP, lab, \boldsymbol{R}\rangle$*, where* S *is a finite set of* states*,* $s_0 \in S$ *the initial state,* Act *is a finite set of* action names*,* AP *is a finite set of* atomic propositions*,* $lab : S \to \{\top, \bot\}^{AP}$ *is a* state-labeling *function that assigns to each state* s *a valuation of the atomic propositions,* $\boldsymbol{R} \subseteq S \times Act \times S \to \mathbb{R}_{\geqslant 0}$ *is a* rate function*. If* $\lambda = \boldsymbol{R}(s, a, s') \wedge \lambda > 0$*, we write* $s \xrightarrow{a,\lambda} s'$*.*

We assume that each state has at least one successor: for all $s \in S$, exists $a \in Act$, $s' \in S$ such that $\boldsymbol{R}(s, a, s') > 0$. CTMC executions lead to timed paths, and a CTMC is a generator of a random path. We define by $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A})$ the probability that the random path of $\mathcal{M}$ is accepted by $\mathcal{A}$ (probability measure of all paths accepted by $\mathcal{A}$ as in [8]).

3 Autonomous Transitions and Expressiveness

We indicate with $\mathbb{A}$ the whole family of automata of Definition 2 and with $\mathbb{A}^{na}$ the subclass of automata with no autonomous transitions: $\mathbb{A}^{na} = \{\mathcal{A} \in \mathbb{A} \mid Aut(\mathcal{A}) = \varnothing\}$ The comparison of the expressive power of $\mathbb{A}$ and $\mathbb{A}^{na}$ is both qualitative (based on the timed path language) and quantitative (based on accepting probabilities).

Definition 6. *Let* $\mathbb{A}_1$ *and* $\mathbb{A}_2$ *be families of DTA. Then:*

- $\mathbb{A}_2$ *is* at least as expressive as $\mathbb{A}_1$ *w.r.t. language, denoted* $\mathbb{A}_1 \prec_{\mathcal{L}} \mathbb{A}_2$*, if for all* $\mathcal{A}_1 \in \mathbb{A}_1$ *there exists* $\mathcal{A}_2 \in \mathbb{A}_2$ *such that* $\mathcal{L}(\mathcal{A}_2) = \mathcal{L}(\mathcal{A}_1)$*;*

- $\mathbb{A}_2$ *is* at least as expressive *as* $\mathbb{A}_1$ *w.r.t. Markov chains, denoted* $\mathbb{A}_1 \prec_{\mathcal{M}} \mathbb{A}_2$, *if for all* $\mathcal{A}_1 \in \mathbb{A}_1$ *there exists* $\mathcal{A}_2 \in \mathbb{A}_2$ *such that for all Markov chains* $\mathcal{M}$, $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}_2) = \mathbf{Pr}_{\mathcal{M}}(\mathcal{A}_1)$.

As usual, we derive other relations between such families. $\mathbb{A}_1$ and $\mathbb{A}_2$ are *equally expressive* w.r.t. language (resp. Markov chains), denoted $\mathbb{A}_1 \sim_{\mathcal{L}} \mathbb{A}_2$ (resp. $\mathbb{A}_1 \sim_{\mathcal{M}} \mathbb{A}_2$) if $\mathbb{A}_1 \prec_{\mathcal{L}} \mathbb{A}_2$ and $\mathbb{A}_2 \prec_{\mathcal{L}} \mathbb{A}_1$ (resp. $\mathbb{A}_1 \prec_{\mathcal{M}} \mathbb{A}_2$ and $\mathbb{A}_2 \prec_{\mathcal{M}} \mathbb{A}_1$). $\mathbb{A}_2$ is *strictly more expressive than* $\mathbb{A}_1$ w.r.t. language (resp. Markov chains), denoted $\mathbb{A}_1 \precneqq_{\mathcal{L}} \mathbb{A}_2$ (resp. $\mathbb{A}_1 \precneqq_{\mathcal{M}} \mathbb{A}_2$) if $\mathbb{A}_1 \prec_{\mathcal{L}} \mathbb{A}_2$ and not $\mathbb{A}_2 \prec_{\mathcal{L}} \mathbb{A}_1$ (resp. $\mathbb{A}_1 \prec_{\mathcal{M}} \mathbb{A}_2$ and not $\mathbb{A}_2 \prec_{\mathcal{M}} \mathbb{A}_1$).

Observe that by definition $\mathbb{A}_1 \prec_{\mathcal{L}} \mathbb{A}_2$ implies $\mathbb{A}_1 \prec_{\mathcal{M}} \mathbb{A}_2$. We now establish that autonomous *resetting* transitions extend the expressive power of DTA w.r.t. Markov chains ($\mathbb{A}^{na} \precneqq_{\mathcal{M}} \mathbb{A}$). The weaker result w.r.t. language ($\mathbb{A}^{na} \precneqq_{\mathcal{L}} \mathbb{A}$) is shown in [11].

Theorem 1. *There exists* $\mathcal{A} \in \mathbb{A}$ *such that for all* $\mathcal{A}' \in \mathbb{A}^{na}$ *there exists a Markov chain* $\mathcal{M}$ *with* $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}') \neq \mathbf{Pr}_{\mathcal{M}}(\mathcal{A})$.

Before proving this theorem, we prove some intermediate properties. We first establish a kind of 0-1 law for DTA in $\mathbb{A}^{na}$ and Markov chains. In order to obtain this intermediate result, we introduce some objects. *Simple chains* are Markov chains with a single action, no atomic proposition (or equivalently with the same valuation for all states) and such that each state s has a single successor state $sc(s)$ reached with rate λ_s. W.r.t. the acceptance probability of simple chains, we can consider DTAs without actions and atomic propositions. Moreover we add to each DTA an additional garbage location and we split the transitions, so that, w.l.o.g. one can assume that for each location ℓ of a DTA in $\mathbb{A}^{na}$, there are $C+1$ outgoing transitions: $\{\ell \xrightarrow{i-1 \leqslant x < i, r_i} sc_i(\ell) \mid 1 \leqslant i \leqslant C\} \cup \{\ell \xrightarrow{x \geqslant C, r_{C+1}} sc_{C+1}(\ell)\}$ where C is the maximal constant occurring in the DTA. The shape of the guards is not a restriction in the context of Markov chains. For all clock valuations $\bar{x}$, the clock valuation $sc(\ell, \bar{x})$ is defined by:

- Let $i = \min(j \mid j \in \{1, \ldots, C\} \wedge \bar{x} < j)$ with $\min(\emptyset) = C+1$;
- If $r_i =\downarrow$ then $sc(\ell, \bar{x}) = 0$ else $sc(\ell, \bar{x}) = \bar{x}$.

Observe the difference between sc_i, defined at the syntactical level, which maps a location to its i^{th} successor and sc, defined at the semantical level, which maps a pair consisting in a location and a clock valuation to the new clock valuation obtained by firing the single transition enabled w.r.t. the clock valuation.

We also define the region (multi-)graph $G_{\mathcal{A}} = (V, E)$ of such a DTA $\mathcal{A}$ as follows.

- V, the set of vertices, is defined by $V = \{(\ell, i) \mid \ell \in L \wedge 0 \leqslant i \leqslant C+1\}$;
- Let (ℓ, i) be a vertex, then for all j s.t. $\max(i, 1) \leqslant j \leqslant C+1$, there is a transition from (ℓ, i) to $(sc_j(\ell), j')$ labelled by j with $j' = 0$ if $r_j =\downarrow$ and $j' = j$ otherwise.

One interprets $G_{\mathcal{A}}$ as follows. The vertex $(\ell, 0)$ corresponds to the region defined by location ℓ with clock valuation 0. The vertex $(\ell, 1)$ corresponds to the region defined by location ℓ with clock valuation in $]0, 1[$. The vertex (ℓ, i) for $1 < i \leqslant C$ corresponds to the region defined by location ℓ with clock valuation in $[i-1, i[$. The vertex $(\ell, C+1)$ corresponds to the region defined by location ℓ with clock valuation in $[C, \infty[$. The transition outgoing from (ℓ, i) labelled by j corresponds to the combination of time elapsing to enter the region (ℓ, j) followed by an action of the Markov chain, leading to either (ℓ', j) or to $(\ell', 0)$, in case of reset.

Given s a state of a Markov chain, ℓ a location of DTA, and $\bar{x}$ a clock valuation, $p(s, \ell, \bar{x})$ denotes the probability of acceptance when the Markov chain starts in s and the DTA starts in ℓ with clock valuation $\bar{x}$. In particular for a DTA $\mathcal{A}$ applied to a Markov chain $\mathcal{M}$, $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}) = p(s_0, \ell_0, 0)$ where s_0 is the initial state of $\mathcal{M}$ and ℓ_0 is the initial location of $\mathcal{A}$ such that $lab(s_0) \models \Lambda(\ell_0)$.

Lemma 1. *Let s be a state of a simple Markov chain $\mathcal{M}$ and ℓ be a location of a DTA in $\mathbb{A}^{na}$. Then the function that maps t to $p(s, \ell, t)$ is continuous and for $i - 1 \leqslant t \leqslant i \leqslant C$ it is equal to:*

$$\int_t^i \lambda_s e^{-\lambda_s(\tau - t)} p(sc(s), sc_i(\ell), sc(\ell, \tau))d\tau + \int_C^\infty \lambda_s e^{-\lambda_s(\tau - t)} p(sc(s), sc_{C+1}(\ell), sc(\ell, \tau))d\tau + \sum_{i<j\leqslant C} \int_{j-1}^j \lambda_s e^{-\lambda_s(\tau - t)} p(sc(s), sc_j(\ell), sc(\ell, \tau))d\tau \qquad (1)$$

The above formula represents the probability of acceptance when the Markov chain starts in s and the DTA starts in ℓ with clock valuation t, with $i-1 \leqslant t \leqslant i \leqslant C$, therefore within the region (l, i). This probability is computed in terms of the probability of having the next CTMC transition within the region (l, i) itself, or any later region (l, j), multiplied by the probability of acceptance from the state reached by accepting the CTMC transition.

Proof. Define $p_n(s, \ell, t)$ as the probability that the run associated with a random timed path of $\mathcal{M}$ starting in s when the DTA starts in ℓ with clock valuation t reaches location ℓ_f after performing n actions. Then for $\ell \neq \ell_f$, $p_0(s, \ell, t) = 0$ and $p_0(s, \ell_f, t) = 1$. Assume that $p_n(s, \ell, t)$ is continuous (and so measurable) for all s and ℓ. Then the following equation holds for $i - 1 \leqslant t \leqslant i \leqslant C$:

$$\begin{aligned} p_{n+1}(s, \ell, t) = & \int_t^i \lambda_q e^{-\lambda_s(\tau - t)} p_n(sc(s), sc_i(\ell), sc(\ell, \tau))d\tau \\ & + \sum_{i<j\leqslant C} \int_{j-1}^j \lambda_s e^{-\lambda_q(\tau - t)} p_n(sc(s), sc_j(\ell), sc(\ell, \tau))d\tau \\ & + \int_C^\infty \lambda_s e^{-\lambda_s(\tau - t)} p_n(sc(s), sc_{C+1}(\ell), sc(\ell, \tau))d\tau \end{aligned}$$

Observe that for $\tau > C$, $p_n(sc(s), sc_{C+1}(\ell), sc(\ell,\tau))$ is constant since if there is a reset then $sc(\ell,\tau) = 0$ and if there is no reset then $sc(\ell,\tau) = \tau > C$ and so the valuation of the clock is irrelevant. Thus the equation can be rewritten as follows.

$$\begin{aligned} p_{n+1}(s,\ell,t) = & \int_t^i \lambda_s e^{-\lambda_s(\tau-t)} p_n(sc(s), sc_i(\ell), sc(\ell,\tau))d\tau \\ & + \sum_{i<j\leqslant C} \int_{j-1}^{j} \lambda_s e^{-\lambda_s(\tau-t)} p_n(sc(s), sc_j(\ell), sc(\ell,\tau))d\tau \\ & + e^{-\lambda_s(C-t)} p_n(sc(s), sc_{C+1}(\ell), sc(\ell, C+1)) \end{aligned}$$

Observe that $\max(1,\lambda_s)e^{-\lambda_s\tau}$ is uniformly continuous. So pick $\eta' > 0$ such that for all $\tau < \tau' \leqslant \tau + \eta'$ $\max(1,\lambda_s)|e^{-\lambda_s\tau} - e^{-\lambda_s\tau'}| \leqslant \frac{\varepsilon}{3C}$. Let $\eta = \min(\eta', \frac{\varepsilon}{3\lambda_s})$. Then for all $t < t' \leqslant t + \eta$, one bounds $|p_{n+1}(s,\ell,t) - p_{n+1}(s,\ell,t')|$ by the sum of three terms using the above equation to establish that $|p_{n+1}(s,\ell,t) - p_{n+1}(s,\ell,t')| \leqslant \varepsilon$. Thus $p_{n+1}(s,\ell,t)$ is continuous. When $t > C$, $p_{n+1}(s,\ell,t)$ is constant and so continuous.
Observe that $p(s,\ell,t) = \lim_{n\to\infty} p_n(s,\ell,t)$. So the mapping $p(s,\ell,t)$ is measurable as a limit of continuous mappings. Thus Eq. 1 holds for $i - 1 \leqslant t \leqslant i \leqslant C$: Repeating the same argument as the one for the inductive case yields the result. When $t > C$, $p(s,\ell,t)$ is constant and so continuous.

Proposition 1. *Let $\mathcal{A} \in \mathbb{A}^{na}$ and $z \in [0,1]$ such that for all Markov chains $\mathcal{M}$, $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}) = z$, then $z \in \{0,1\}$.*

Proof. We will even prove this result when restricting the quantification to Markov chains with a single action and a single valuation of propositions for all states and a single successor for all states. Thus we can omit propositions and actions in the DTA and only consider simple chains.

Let $\mathcal{A}$ be an automaton that satisfies the hypothesis. We want to establish that *for all configurations (ℓ,t) in some region of $G_{\mathcal{A}}$ reachable from $(\ell_0,0)$, and for all states s of a simple Markov chain, $p(s,\ell,t) = z$.* We do this by induction on the distance from the initial region in the region graph and then we prove that z is either 0 or 1. The basis case of the induction corresponds to the assumption $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}) = z$, for all $\mathcal{M}$.
For the inductive step we assume that for a given (ℓ,t), and for all states s of a simple chain, $p(s,\ell,t) = z$ and we prove that the $p(s',\ell',t') = z$, for all (s',ℓ',t') reachable in one step from (s,ℓ,t).
Let $\mathcal{M}$ be an arbitrary simple chain and define $\mathcal{M}_\lambda$ as the simple chain with a single transition outgoing from its initial state to the initial state of $\mathcal{M}$ whose rate is λ. Let s be the initial state of $\mathcal{M}_\lambda$.
By assumption, $p(s,\ell,t) = z$. Define $f(\tau)$ by $p(sc(s), sc_j(\ell), sc(\ell, t+\tau))$ when $j - 1 < t + \tau \leqslant j \leqslant C$ and by $p(sc(s), sc_{C+1}(\ell), sc(\ell, t+\tau))$ when $t + \tau > C$. Equation 1 can be rewitten as $p(s,\ell,t) = \int_{\tau\geqslant 0} \lambda e^{-\lambda\tau} f(\tau)d\tau$. Since for all λ,

$\mathbf{Pr}_{\mathcal{M}_\lambda}(\mathcal{A}) = z$, the Laplace transform of $f(\tau)$ is equal to $\frac{z}{\lambda}$, i.e. the Laplace transform of the constant function z. By the theorem of unicity of Laplace transforms, this entails that $f(\tau) = z$ except for a set of null measure. However, consider a successor region (ℓ', i) of location ℓ with clock valuation t'.

- Either $i = 0$ (meaning that there has been a reset) and the region has a single point reached with non null probability. So $p(sc(s), \ell', 0) = z$.
- Or $i > 0$, so by Lemma 1, $p(sc(s), \ell', t')$ is continuous inside the region w.r.t. t' and thus everywhere equal to z.

So the induction is established. So if a region of ℓ_f is reachable in the region graph, then $z = 1$. Otherwise ℓ_f is not reachable implying that no run is accepting, and thus $z = 0$.

We can now prove Theorem 1 ($\mathbb{A}^{na} \not\preceq_{\mathcal{M}} \mathbb{A}$).

Proof of Theorem 1. The DTA $\mathcal{A}$ in Fig. 1 (lower right) has an action set reduced to a singleton $\{a\}$ (omitted in the figure) and an empty set of propositions. The language of $\mathcal{A}$ is the set of timed paths whose first action occurs at time $\tau \in [2i, 2i+1[$ for some $i \in \mathbb{N}$. Assume by contradiction that there exists $\mathcal{A}' \in \mathbb{A}^{na}$ such that for all Markov chain $\mathcal{M}$, $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}') = \mathbf{Pr}_{\mathcal{M}}(\mathcal{A})$.

Pick an arbitrary Markov chain $\mathcal{M}$ and define $\mathcal{M}_\lambda$ as the Markov chain which has a single transition from its initial state to the initial state of $\mathcal{M}$ with rate λ. It is routine to check that $\mathbf{Pr}_{\mathcal{M}_\lambda}(\mathcal{A}) = \frac{1-e^{-\lambda}}{1-e^{-2\lambda}}$ (as only the first transition of $\mathcal{M}_\lambda$ is relevant) and, consequently, $\lim_{\lambda\to 0} \mathbf{Pr}_{\mathcal{M}_\lambda}(\mathcal{A}) = \frac{1}{2}$ and, given the hypothesis, also $\lim_{\lambda\to 0} \mathbf{Pr}_{\mathcal{M}_\lambda}(\mathcal{A}') = \frac{1}{2}$.

$\mathbf{Pr}_{\mathcal{M}_\lambda}(\mathcal{A}')$ can be decomposed as $p_{1,\lambda} + p_{2,\lambda}$ where $p_{1,\lambda}$ is the probability to accept the random timed path and that the first action takes place at most at time C and $p_{2,\lambda}$ is the probability to accept the random timed path and that the first action takes place after C, where C is the maximal constant of $\mathcal{A}'$. But $\lim_{\lambda\to 0} p_{1,\lambda} = 0$ and therefore $\lim_{\lambda\to 0} p_{2,\lambda} = \frac{1}{2}$.

On the other hand, let ℓ_1 be the location of $\mathcal{A}'$ reached from its initial location when the value of the clock is greater than C, its maximal constant. There must be one, if not $\lim_{\lambda\to 0} p_{2,\lambda} = 0$, which contradicts what derived above. We want to design an automaton $\mathcal{A}''$ equivalent to $\mathcal{A}'$ when reaching ℓ_1 with clock value greater than C: any timed path is accepted by $\mathcal{A}''$ iff it is accepted by $\mathcal{A}'$ when starting in ℓ_1 with clock valuation greater than C. For the construction we duplicate the automaton and merge the final location, the initial location is location ℓ_1 of the first copy, and in the first copy we add to the guard of all transitions the formula $x > C$ and redirect the transitions that reset the clock to the corresponding location of the second copy.

But then $\lim_{\lambda\to 0} p_{2,\lambda} = \mathbf{Pr}_{\mathcal{M}}(\mathcal{A}'')$. Since $\lim_{\lambda\to 0} p_{2,\lambda} = \frac{1}{2}$ and $\mathcal{M}$ is arbitrary, this contradicts Proposition 1 applied to $\mathcal{A}''$.

The DTA in Fig. 1 (upper right) shows that the above counter-example is of practical interest. Consider a periodic system that cycles over phases of duration 2, each split in two sub-phases of duration 1 (for example a running and a reset phase) and that can experience good (G), bad (B), and neutral (N) actions, generated from a CTMC of arbitrary complexity. The depicted DTA allows one to compute the probability of the CTMC behaviours characterized by a good action in the running sub-phase, given that in the preceding phases no bad action has happened in the running phase. Any action is instead allowed during the reset phase.

4 Autonomous Transitions and Conciseness

We have established that there exists DTAs that cannot be translated into DTAs without autonomous transitions ($\mathbb{A}^{na} \not\precsim_{\mathcal{M}} \mathbb{A}$). We now investigate whether restricted forms of use of autonomous transitions are as expressive as $\mathbb{A}^{na}$. To this goal we identify two additional subclasses of $\mathbb{A}$, namely $\mathbb{A}^{nra}$ and $\mathbb{A}^{rc}$, characterized by a limited presence of autonomous transitions and that are in the following subset relationship: $\mathbb{A}^{na} \subseteq \mathbb{A}^{nra} \subseteq \mathbb{A}^{rc} \subseteq \mathbb{A}$.

Restricted cycles. $\mathbb{A}^{rc}$ is the subclass of automata $\mathcal{A} \in \mathbb{A}$ in which all cycles of $\mathcal{A}$ including an autonomous transition with a reset also include a synchronizing transition $(\ell, \gamma, B, r, \ell')$ with $r = \downarrow$ or $\gamma = (x > C)$.

No reset on autonomous transitions. $\mathbb{A}^{nra}$ is the subclass of automata $\mathcal{A} \in \mathbb{A}^{rc}$ in which there is no autonomous transition that resets the clock: $\mathbb{A}^{nra} = \{\mathcal{A} \in \mathbb{A} \mid (\ell, \gamma, \sharp, r, \ell') \in Aut(\mathcal{A}) \Rightarrow r = \varnothing\}$.

The DTA on the left of Fig. 1 belongs to $\mathbb{A}^{rc} \backslash \mathbb{A}^{nra}$: indeed there is an autonomous transition with reset (from ℓ_1 to ℓ_0), therefore it is not in $\mathbb{A}^{nra}$, but although the transition is part of a cycle, that cycle also includes a synchronizing transition with reset (from ℓ_0 to ℓ_1). Any DTA with no reset on autonomous transitions is an example of $\mathbb{A}^{nra}$. The family $\mathbb{A}^{rc}$ has been introduced to provide an accurate syntactical characterization of DTA for which the autonomous transitions do not add expressive power. In some sense, the DTA of Theorem 1 emphasizes the interest of $\mathbb{A}^{rc}$ since the cycle performed by the autonomous resetting transition points out what increases the expressive power. $\mathbb{A}^{nra}$, which forbids clock resets on autonomous transitions, removes from CSL$^{\text{TA}}$ the capacity of combining time constants depending on the time elapsed during (a portion of) an execution. As observed in [12](section 4), clock resets on autonomous transitions are what makes CSL$^{\text{TA}}$ more expressive than asCSL [6].
The following frame summarizes the results for $\mathbb{A}$ subclasses.

$$\mathbb{A}^{na} \sim_{\mathcal{L}} \mathbb{A}^{nra} \sim_{\mathcal{L}} \mathbb{A}^{rc} \not\precsim_{\mathcal{M}} \mathbb{A}$$

with $\mathbb{A}^{rc}$ ($\mathbb{A}^{nra}$) exponentially more concise than $\mathbb{A}^{nra}$ ($\mathbb{A}^{na}$, respectively)

We first establish that in $\mathbb{A}^{rc}$ the autonomous resetting transitions can be mimicked in $\mathbb{A}^{nra}$ using additional finite memory, but with exponential cost.

Proposition 2. *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}^{rc}$ and outputs $\mathcal{A}' \in \mathbb{A}^{nra}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Sketch of Proof. The construction (1) duplicates locations by memorizing in the location an integer value, (2) take into account this value for modifying the guard and the destination of the outgoing transitions, and (3) deletes the reset of autonomous transitions. This value corresponds to the accumulated value of constants in the guards of resetting autonomous transitions since the last visit of a synchronizing transition with a reset or a guard $x > C$. The restriction over $\mathbb{A}^{rc}$ ensures that this value is bounded by some finite integer K. However K may be exponential in the size of $\mathcal{A}$ and thus this transformation is exponential.

The exponential blowup due to the duplication of locations is unavoidable:

Proposition 3. *There exists a family $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ in $\mathbb{A}^{rc}$ such that the size of $\mathcal{A}_n$ is $O(n^2)$ and for all $\mathcal{A} \in \mathbb{A}^{nra}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$, $(|Aut| + 1)|Synch| \geqslant 2^n$.*

We now prove that autonomous transitions in $\mathbb{A}^{nra}$ can be eliminated, also at an exponential cost.

Proposition 4. *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}^{nra}$ and outputs $\mathcal{A}' \in \mathbb{A}^{na}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Sketch of Proof. The construction proceeds in two steps: at first, cycles of autonomous transitions are eliminated, then all (linear) paths of autonomous transitions are eliminated. The first construction is quadratic, as we duplicate each location to store in the location the information on the number of autonomous transitions visited since the last visit of a synchronized transition. The idea of this construction is that if a path exceeds the number of autonomous transitions it must visit twice the same autonomous transition without visiting a synchronized transition and so diverges. In words: in the resulting DTA, divergence has been transformed into deadlock. This finite memory has a linear size w.r.t. the size of the original DTA.

The second step consists in eliminating autonomous transitions when there are no such cycles. The key point is to select a location ℓ which is the source of the last autonomous transition of a maximal path of such transitions. Thus every autonomous transition outgoing from ℓ reaches some location ℓ_u where only synchronized transitions are possible. Roughly speaking, the construction builds a synchronized transition corresponding to a sequence of an autonomous transition followed by a synchronized transition. However the construction is more involved since ℓ has to be duplicated in order to check which autonomous transition can be triggered (or if no autonomous transition can be triggered). This duplication also has an impact on the incoming transitions of ℓ. Repeating (at most $|L|$ times) this transformation eliminates all autonomous transitions. The exponential blowup due to the repetition of duplication of locations is unavoidable:

Proposition 5. *There exists a family of automata $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ in $\mathbb{A}^{nra}$ such that the size of $\mathcal{A}_n$ belongs to $O(n\log(n))$ and for all $\mathcal{A} \in \mathbb{A}^{na}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$ the number of its locations is at least 2^n.*

5 Conclusion and Future Work

We have established that autonomous transitions do enhance expressiveness of single clock DTAs, and more precisely for the less discriminating case of the probability of the random paths of a CTMC accepted by the DTA. This is the most relevant one for comparing some variations of (1-clock) CSL^{TA} defined in the literature. This enhanced expressiveness is due to the possibility of associating clock resets with autonomous transitions that occur in a cycle. The small counterexample of Proposition 1 can be seen as the basic construct to study systems with periodic behaviours or periodic phases, with clear practical implications. Even in DTA subclasses for which the autonomous transitions do not enhance expressiveness, they do play a role in defining concise DTAs: removing autonomous transitions may lead to an exponential blow up of the DTA.

We plan to investigate whether the precise identification of the characteristics that enhance expressiveness and conciseness can help the identification of the best algorithms for CSL^{TA} model-checking, in particular for the component-based method [4]. Moreover, following the suggestion by an anonymous reviewer, we intend to investigate further consequences of Proposition 1, for example to study systems that include probabilistic choices of autonomous transitions.

References

1. Ajmone-Marsan, M., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: Modelling with Generalized Stochastic Petri Nets. Wiley, Hoboken (1995)
2. Amparore, E.G., Ballarini, P., Beccuti, M., Donatelli, S., Franceschinis, G.: Expressing and computing passage time measures of GSPN models with HASL. In: Colom, J.-M., Desel, J. (eds.) PETRI NETS 2013. LNCS, vol. 7927, pp. 110–129. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38697-8_7
3. Amparore, E.G., Donatelli, S.: MC4CSL^{TA}: an efficient model checking tool for CSL^{TA}. In: QEST 2010, pp. 153–154. IEEE Computer Society (2010)
4. Amparore, E.G., Donatelli, S.: Efficient model checking of the stochastic logic CSLTA. Perform. Eval. **123–124**, 1–34 (2018)
5. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model-checking continuous-time Markov chains. ACM Trans. Comput. Log. **1**(1), 162–170 (2000)
6. Baier, C., Cloth, L., Haverkort, B.R., Kuntz, M., Siegle, M.: Model checking Markov chains with actions and state labels. IEEE TSE **33**, 209–224 (2007)
7. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: On the logical characterisation of performability properties. In: Montanari, U., Rolim, J.D.P., Welzl, E. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 780–792. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45022-X_65
8. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. IEEE TSE **29**(6), 524–541 (2003)
9. Ballarini, P., Barbot, B., Duflot, M., Haddad, S., Pekergin, N.: HASL: a new approach for performance evaluation and model checking from concepts to experimentation. Perform. Eval. **90**, 53–77 (2015)
10. TChen, T., Han, T., Katoen, J.-P., Mereacre, A.: Model checking of continuous-time Markov chains against timed automata specifications. Log. Methods Comput. Sci. **7**(1:12), 1–34 (2011)

11. Donatelli, S., Haddad, S.: Autonomous Transitions Enhance CSA$^{\mathrm{TA}}$ Expressiveness and Conciseness. Research report, Inria Saclay Ile de France, LSV, ENS Cachan, CNRS, INRIA, Université Paris-Saclay, Cachan, France, Universita degli Studi di Torino, October 2019. https://hal.inria.fr/hal-02306021
12. Donatelli, S., Haddad, S., Sproston, J.: Model checking timed and stochastic properties with CSL$^{\mathrm{TA}}$. IEEE TSE **35**(2), 224–240 (2009)
13. Feng, Y., Katoen, J.-P., Li, H., Xia, B., Zhan, N.: Monitoring CTMCs by multiclock timed automata. In: Chockler, H., Weissenbacher, G. (eds.) CAV 2018. LNCS, vol. 10981, pp. 507–526. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96145-3_27
14. Kuntz, M., Haverkort, B.R.: GCSRL-a logic for stochastic reward models with timed and untimed behaviour. In: 8th PMCCS, pp. 50–56 (2007)
15. Meyer, J.F., Movaghar, A., Sanders, W.H.: Stochastic activity networks: structure, behavior, and application. In: International Workshop on Timed Petri Nets, pp. 106–115. IEEE CS (1985)
16. Obal II, W.D., Sanders, W.H.: State-space support for path-based reward variables. Perform. Eval. **35**, 233–251 (1999)