Guarded Autonomous Transitions Increase Conciseness and Expressiveness of Timed Automata

S. Donatelli[1] and S. Haddad[2]

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

Abstract. Timed Automata (TA) are an appropriate model for specifying timed requirements for Continuous Time Markov Chain (CTMC). However in order to keep tractable the model checking of TA over a CTMC, temporal logics based on TA, like CSL^{TA}, restrict TA to have a single clock and to be deterministic (DTA). Different variants of DTA have been proposed to address the issue of their expressiveness and conciseness. Here we study the effect of two possible features: (1) autonomous transitions which are triggered by time elapsing in addition to synchronized transitions and (2) transitions guarded by propositional formulas instead of propositional formulas guarding locations. We first show that autonomous guarded transitions increase the expressiveness of DTA (as already shown for guarded locations). Then we identify a hierarchy of DTA subclasses all equivalent to DTA without autonomous transitions and we analyze their respective conciseness. In particular we show that eliminating reset in autonomous transitions implies an exponential blow-up while eliminating autonomous transitions without reset can be performed in polynomial time with the help of decision diagrams. Finally we compare TA with guarded transitions to TA with guarded locations showing that the former model is exponentially more concise than the latter one.

1 Introduction

Model checking CTMC. Defining some modal logic for specifying properties of a CTMC is a natural goal, since a CTMC is a (probabilistic) transition system. In fact the first main logic that has been proposed, CSL [5], is a variant of CTL where (1) the 'for all paths' and 'there exists a path' operators have been replaced by the operator expressing 'the probability that a random path is greater (or smaller) than some threshold' and (2) the 'until' operator is equipped with a time interval. The core of the associated model checking procedure consists in building some CTMCs and to analyze their transient behavior. CSL has been extended in several directions [6, 7] and tailored for dealing with CTMCs generated by generalized stochastic Petri nets [13]. Another approach consists in specifying the formula by a timed automaton (or even an hybrid automaton)

such as done in [16, 3, 10]. However without restriction, the model checking procedure can (1) either be based on simulation which only provides an estimation of the probability to be computed or (2) numerically solve multiple integrals which do not scale at all. When the timed automaton has a single clock and is deterministic (DTA), there is an efficient model checking procedure related to some particular Markov regenerative process. In addition, the logic CSL^{TA} [12] which follows such an approach has be proven to extend CSL and most of its variants.

Classes of DTA. The basic family of DTA only includes *synchronized transitions*: such a transition has a temporal guard and a subset of actions. The synchronized product of the CTMC and the DTA evolves by triggering a transition of the CTMC which can be matched by a transition of a DTA (i.e. the guard should be satisfied and the action labelling the transition of the CTMC is included in the subset labelling the transition of the DTA). In order to increase the expressive power of this family one introduces *autonomous transitions*: such a transition has a temporal threshold and when the clock value reaches this threshold, in the synchronized product only the DTA evolves. Moreover since a state of CTMC can be labelled by a valuation of atomic propositions, (1) either one can label the locations of the DTA by a propositional formula that should be satisfied by the valuation of the matching state, (2) or one can label the transitions of the DTA by a precondition that should be satisfied by the valuation of the matching source state and a precondition that should be satisfied by the valuation of the matching target state. The former family is denoted $\mathbb{A}_s$ while the latter is denoted $\mathbb{A}_g$, the main topic of our work.

Expressiveness and conciseness. In order to compare *expressiveness* of families of DTA, the usual qualitative notion is related to their timed languages: a family $\mathbb{A}$ is at least as expressive as a family $\mathbb{A}'$ if for any DTA in $\mathbb{A}$ there is a DTA in $\mathbb{A}'$ with the same language. Since the DTAs we study are used for defining the acceptance probability of a CTMC, we also introduce a quantitative notion: a family $\mathbb{A}$ is at least as expressive as a family $\mathbb{A}'$ if for any DTA in $\mathbb{A}$ there is a DTA in $\mathbb{A}'$ such that for any CTMC their acceptance probabilities are equal. When a family $\mathbb{A}$ is at least as expressive as a family $\mathbb{A}'$, it raises other issues: (1) *effectiveness*, i.e. does there exist an algorithm for producing an equivalent DTA? (2) *cost*, i.e. what is the complexity of this algorithm? and (3) *conciseness* what is the size of the equivalent DTA w.r.t. the size of the original one?

Our contributions. We first show that in $\mathbb{A}_g$ (as in $\mathbb{A}_s$ [11]) autonomous transitions strictly increase expressiveness even w.r.t. the quantitative notion. Then we characterize a large subclass of $\mathbb{A}_g$, denoted $\mathbb{A}_g^{rc}$ for which autonomous transitions do not increase expressiveness even w.r.t. the qualitative notion. This class $\mathbb{A}_g^{rc}$ includes the class of DTAs with not reset on autonomous transitions, denoted $\mathbb{A}_g^{nra}$, but we prove that $\mathbb{A}_g^{rc}$ is exponentially more concise than $\mathbb{A}_g^{nra}$. On the contrary, we establish that one can transform a DTA in $\mathbb{A}_g^{nra}$ into an equivalent DTA with no autonomous transition in polynomial time. This reduction

is tricky and requires to specify propositional formulas by decision diagrams. Finally we compare $\mathbb{A}_g$ and $\mathbb{A}_s$ showing that the former family is exponentially more concise than the latter one.

Organization. In Section 2, we introduce the syntax and semantics of DTA with autonomous guarded transitions and we define qualitative and quantitative notions of expressiveness. In Section 3 we present a hierarchy of subclasses of DTA with autonomous guarded transitions and establish a full classification w.r.t. expressiveness and conciseness. Section 4 establishes that using guarded transitions yields an exponentially more concise model than the one with guarded locations. Finally we conclude and give some perspectives to this work in Section 5. The appendix includes some proofs.

2 Preliminaries.

DTA are interpreted as a way to define timed paths of a CTMC, that can be accepted of rejected by the DTA itself. The CTMCs we consider are CTMCs with actions from a set Act and a valuation of a set of propositions AP associated to the CTMC states. Assuming indices start at 0, we identify the $(i+1)$-th state of a timed path with v_i, the boolean evaluation of the atomic propositions in that state, δ_i, the delay before action a_i or equivalently the sojourn time in state i, and τ_i to indicate 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 $Act = \{a, b\}$ a timed path $(\{p, q\}, 0.3) \xrightarrow{a} (\{p\}, 0.2) \xrightarrow{b} (\{q\}, 1) \xrightarrow{a} \cdots$, is interpreted as (1) the system staying in a state fulfilling $p \wedge q$ in the time interval $[0, 0.3[$, where at time 0.3 action a takes place, (2) the system moves to a state fulfilling $p \wedge \neg q$, stays there for 0.2 time units and then action b takes place, and (3) the system moves to a state fulfilling $\neg p \wedge q$, stays there for 1 time units and then action a takes place (at the global time $\tau = 1.5$).

DTA definition includes a clock x and two types of timed constraints associated with transitions: 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. Transitions also have an input and an output guard (indicated with φ^- and φ^+ respectively).

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 CTMC) that can be represented by a timed path,

the current location, say ℓ, is matched with the current state of the system, say $s = (v_i, \delta_i)$. This matching evolves in three ways depending on the delay $d \leqslant \delta_i$ (initially equals to δ_0), elapsed until the next transition $(v_i, \delta_i) \xrightarrow{a_i} (v_{i+1}, \delta_{i+1})$ of the system.

- Either after some delay $\delta \leqslant d$, there is an outgoing *autonomous transition* from ℓ which is *enabled*, meaning that (1) its boundary condition (say $x = \alpha$) is satisfied and (2) v_i fulfills φ^-. Then after delay δ, ℓ' is matched with s and d is decreased by δ.
- Else if there is a *synchronizing transition* outgoing from ℓ after time d has elapsed is *enabled* meaning that (1) v_i satisfies φ^-, (2) its inner condition (say $\alpha \bowtie x \bowtie' \beta$) is satisfied, (3) the action a belongs to the subset of actions associated with the synchronizing transition, and (4) v_{i+1} satisfies φ^+. Then after delay δ_i, ℓ' is matched with $s' = (v_{i+1}, \delta_{i+1})$ and d is set to δ_{i+1}.
- Otherwise there is no possible matching and the timed path is rejected by the DTA.

When a transition of the DTA is fired, clock x may keep its current value or may be reset. 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 by the existence of the unique (looping) synchronizing transition from ℓ_f with no boolean guards, no timing and no action conditions. Observe that the synchronization may go on 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) synchronizing transitions outgoing from the same location are never simultaneously enabled, (2) autonomous transitions outgoing from the same location are never simultaneously enabled, and (3) autonomous transitions have priority over synchronizing transitions.

Definition 2 (DTA). *A DTA is defined by a tuple* $\mathcal{A} = \langle L, \ell_0, \ell_f, AP, Synch, Aut \rangle$ *where* L *is a finite set of* locations, $\ell_0 \in L$ *is the* initial location, $\ell_f \in L$ *is the* final location, $Synch \subseteq L \times \mathcal{B}_{AP} \times \mathsf{InC} \times 2^{Act} \times \{\varnothing, \downarrow\} \times \mathcal{B}_{AP} \times L$ *is the set of* synchronizing transitions, *and* $Aut \subseteq L \times \mathcal{B}_{AP} \times \mathsf{BoundC} \times \sharp \times \{\varnothing, \downarrow\} \times L$ *is the set of* autonomous transitions.

$\ell \xrightarrow{\varphi^-, \gamma, B, r, \varphi^+} \ell'$ *denotes the synchronized transition* $(\ell, \varphi^-, \gamma, B, r, \varphi^+, \ell')$.

and $\ell \xrightarrow{\varphi^-, \gamma, \sharp, r(, \varphi^-)} \ell'$ *(repeating sometimes the* φ^- *formula for unifying the notation) denotes the autonomous transition* $(\ell, \varphi^-, \gamma, \sharp, r, \ell')$.

Furthermore $\mathcal{A}$ *fulfills the following conditions.*

- **Determinism on actions.** $\forall B, B' \subseteq Act \; s.t. \; B \cap B' \neq \varnothing, \forall \ell, \ell', \ell'' \in L$,
  if $\ell \xrightarrow{\varphi^-, \gamma, B, r, \varphi^+} \ell' \wedge \ell \xrightarrow{\varphi'^-, \gamma', B', r', \varphi'^+} \ell''$ *then*
  $\varphi^- \wedge \varphi'^- \Leftrightarrow \bot$ *or* $\varphi^+ \wedge \varphi'^+ \Leftrightarrow \bot$ *or* $\gamma \wedge \gamma' \Leftrightarrow \bot$.
- **Determinism on autonomous transitions.** $\forall \ell, \ell', \ell'' \in L$,
  if $\ell \xrightarrow{\varphi^-, x = \alpha, \sharp, r} \ell'$ *and* $\ell \xrightarrow{\varphi'^-, x = \alpha', \sharp, r'} \ell''$ *then* $\varphi^- \wedge \varphi'^- \Leftrightarrow \bot$ *or* $\alpha \neq \alpha'$.
- **Condition on the final location.** $\ell_f \xrightarrow{\top, \top, Act, \varnothing, \top} \ell_f \in Synch$.

$\mathbb{A}_g$ denotes the whole family of automata of Definition 2. We informally write "a transition with reset" or "a transition without reset" to indicate the condition $r = \downarrow$ and $r = \varnothing$ respectively.

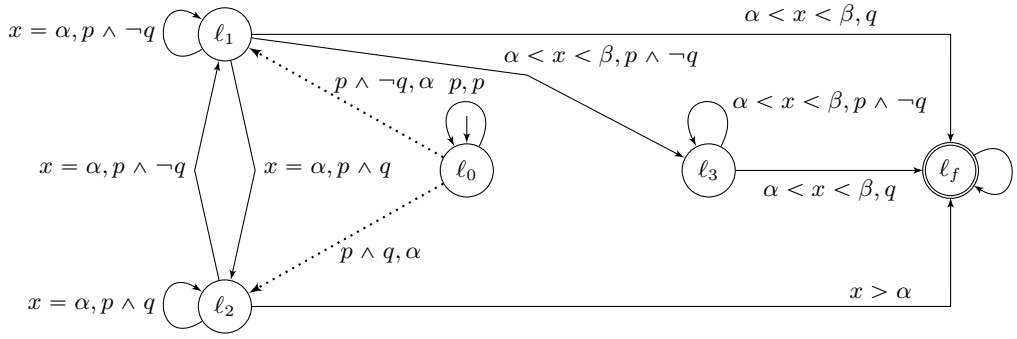


Fig. 1. A DTA specification of $p\mathbf{U}^{]\alpha,\beta[}q$ with $\alpha > 0$.

*Example 2 (**DTA example**).* Figure 1SD: out of margins
shows a DTA with locations ℓ_0, ℓ_1, ℓ_2, ℓ_3 and ℓ_f. The single initial location is ℓ_0. Autonomous transitions are depicted as dotted arcs, while synchronizing are depicted as solid arcs. For readability and conciseness we omit in the drawings: 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 clock guards (like $x \geqslant 0$) and input or output guards; 5) the name x of the clock in $x = \alpha$ guards of autonomous transitions. As a result an autonomous transition is depicted as either $l \overset{\varphi^-,\alpha}{\dashrightarrow} l'$, or as $l \overset{\varphi^-,\alpha,\downarrow}{\dashrightarrow} l'$,if there is a clock reset. SD: not quite the same arrow as in Fig. 1
The two dotted arcs out of ℓ_0 correspond to autonomous transitions, mutually exclusive due to their guards. Note that the self loop on ℓ_0 represents a synchronizing transitions, so it is mutually exclusive with the other two transitions out of ℓ_0 because of priority. Note that guards of the form $x = \alpha$, typically associated with autonomous transitions, can also be associated to synchronizing transitions. Figure 1 illustrates how to specify the temporal formula $p\mathbf{U}^{]\alpha,\beta[}q$ with a DTA $\mathcal{A} \in \mathbb{A}_g$. Since the clock counts time elapsed, no reset occurs. Observe that in the interval $[0, \alpha[$, the current location can only be ℓ_0 with the additional requirement that if an action occurs then p has to be fullfilled inside the whole interval. At time α, the current location can only be ℓ_1 or ℓ_2 depending on the truth value of q and with the guarantee that p holds in the interval $[0, \alpha]$. Considering the first action that occurs after α, there are three possible cases: (1) $p \wedge q$ was satisfied and the formula is satisfied (by taking transition

from ℓ_2 to ℓ_f), (2) $p \wedge \neg q$ was satisfied, q is now satisfied and the action occurs before β and so the formula is satisfied (by taking transition from ℓ_1 to ℓ_f) (3) or $p \wedge \neg q$ was satisfied and it is still satisfied and the action occurs before β, and so (by taking transition from ℓ_1 to ℓ_3) there is the same possibility to satisfy the formula represented by location ℓ_3.

Definition 3 (Run of a DTA).

A run *of* $\mathcal{A} \in \mathbb{A}_g$ *is a sequence:* $\rho = (\ell_0, v_0, \bar{x}_0, \delta_0) \xrightarrow{\varphi_0^-, \gamma_0, B_0, r_0, \varphi_0^+} (\ell_1, v_1, \bar{x}_1, \delta_1) \xrightarrow{\varphi_1^-, \gamma_1, B_1, r_1, \varphi_1^+} \cdots (\ell_i, v_i, \bar{x}_i, \delta_i) \xrightarrow{\varphi_i^-, \gamma_i, B_i, r_i, \varphi_i^+} \cdots$ *such that for all* $i \in \mathbb{N}$,

$\ell_i \in L$, $v_i \in \{\bot, \top\}^{AP}$, $\delta_i \in \mathbb{R}_{\geqslant 0}$, $\ell_i \xrightarrow{\varphi_i^-, \gamma_i, B_i, r_i, \varphi_i^+} \ell_{i+1} \in E \ = \mathit{Synch} \cup \mathit{Aut}$,

$v_i \models \varphi_i^-$, $v_{i+1} \models \varphi_i^+$, $\bar{x}_i + \delta_i \models \gamma_i$, $\bar{x}_{i+1} = \begin{cases} 0 & \text{if } r_i = \downarrow \\ \bar{x}_i + \delta_i & \text{otherwise} \end{cases}$

Let $\bar{x}_\sharp = min\{\alpha \mid \exists \ell_i \xrightarrow{\varphi, x=\alpha, \sharp, r} \ell' \in E \wedge \bar{x}_i \leqslant \alpha \wedge v_i \models \varphi\}$.
If $B_i = \sharp$ *then* $\bar{x}_i + \delta_i = \bar{x}_\sharp$ *and* $v_{i+1} = v_i$ *else* $\bar{x}_i + \delta_i < x_\sharp$.

*Example 3 (**DTA run**).* In the run we, again, describe v in terms of the subset of AP that evaluate to $\top$. Let us describe a possible run of the DTA of Figure 1, assuming $\alpha = 1$ and $\beta = 4$. The run starts with $v_0 = \{p\}$; at time 0.4, it goes from ℓ_0 to ℓ_0 by performing the synchronizing transition of the self-loop over ℓ_0. Then at time 1.0, it autonomously goes to location ℓ_2. If the next action happens at time 6.0 then it goes to ℓ_f. Note that this is a case in which the formula is satisfied already at time β, since $\beta = 4$, but the run reaches the final location only at time $5.0 > \beta$, when the first synchronizing transition takes place, and stays in the final location forever. The run described above corresponds, in more formal terms, to:

$(\ell_0, \{p\}, \bar{x}_0 = 0.0, \delta_0 = 0.4) \xrightarrow{p, x \geqslant 0, Act, \varnothing, p} (\ell_0, \{p, q\}, 0.4, 0.6) \xrightarrow{p \wedge q, x=1, \sharp, \varnothing}$
$(\ell_f, \{p, q\}, 1.0, 5.0) \xrightarrow{\top, x>1, Act, \varnothing, \top} (\ell_f, \{p\}, 6.0, 2.4) \xrightarrow{\top, x \geqslant 0, Act, \varnothing, \top} (\ell_f, \varnothing, 8.4, 0.7) \cdots$

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 "couple" 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.

Definition 4. *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* $\rho = (\ell_0, v'_0, \bar{x}_0, \delta'_0) \xrightarrow{\varphi_0^-, \gamma_0, B_0, r_0, \varphi_0^+} \cdots (\ell_i, v'_i, \bar{x}_i, \delta'_i) \xrightarrow{\varphi_i^-, \gamma_i, B_i, r_i, \varphi_i^+} \cdots$ *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 *that visits* ℓ_f.
The language $\mathcal{L}(\mathcal{A})$ *of* $\mathcal{A}$ *is the set of the timed paths accepted by* $\mathcal{A}$.

Note that, due to determinism, if such a run exists, it is unique.

*Example 4 (**Timed path recognized by a DTA run**).* The timed path $\sigma = (\{p\}, 0.4) \xrightarrow{a} (\{p,q\}, 5.6) \xrightarrow{b} (\{p\}, 2.4) \xrightarrow{c} (\varnothing, 0.7) \cdots$ is accepted by the DTA of Figure 1 using the run of Example 3, with the mapping κ. where $\kappa(0) = 0$ and for all $i > 0$ $\kappa(i) = i + 1$.
The (Zeno) timed path $\sigma = (\{p\}, 0) \xrightarrow{a} (\{p\}, 0) \xrightarrow{a} (\{p\}, 0) \cdots$ is recognized (but not accepted) by the DTA of Figure 1 using the run $(\ell_0, \{p\}, 0, 0) \xrightarrow{p, x \geqslant 0, Act, \varnothing, p} (\ell_0, \{p\}, 0, 0) \xrightarrow{p, x \geqslant 0, Act, \varnothing, p} (\ell_0, \{p\}, 0, 0) \cdots$, with mapping κ being the identity.

When a DTA is used for model checking a CTMC, it can be considered as a way to select the subset of timed paths of a CTMC accepted by the DTA. The CTMC we consider are CTMC with actions from a set Act and a valuation of a set of propositions AP associated to the CTMC states, as in the following definition.

Definition 5 (CTMC). *A* continuous time Markov chain $\mathcal{M}$ *with state and action labels is defined by* $\mathcal{M} = \langle S, s_0, Act, AP, lab, R \rangle$*, 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,* $R \subseteq S \times Act \times S \to \mathbb{R}_{\geqslant 0}$ *is a* rate function*. If* $R(s, a, s') > 0$*, we write* $s \xrightarrow{a, R(s,a,s')} s'$*.*

We assume that each state has at least one successor: for all $s \in S$, there exists $a \in Act$, $s' \in S$ such that $R(s, a, s') > 0$. 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 defined in [8]).

The paper objective is to compare different classes of DTAs in qualitative terms (i.e., w.r.t. timed path languages) and in probabilistic terms (i.e., w.r.t. accepting probabilities of the accepted path in a CTMC). These notions are independent of the type of DTA, and they have been already defined in [11].

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. CTMCs, 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 CTMC* $\mathcal{M}$*,* $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}_2) = \mathbf{Pr}_{\mathcal{M}}(\mathcal{A}_1)$*.*

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. CTMCs), denoted $\mathbb{A}_1 \precnsim_{\mathcal{L}} \mathbb{A}_2$ (resp. $\mathbb{A}_1 \precnsim_{\mathcal{M}} \mathbb{A}_2$) 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 $\mathbb{A}_1 \prec_{\mathcal{M}} \mathbb{A}_2$.

3 Eliminating autonomous transitions in $\mathbb{A}_g$

This section studies the role of autonomous transitions in $\mathbb{A}_g$. The role of autonomous transitions for DTAs in which conditions are associated with locations (as in [12]) has been investigated in [11], where it was shown that there are indeed certain subclasses of DTAs for which autonomous transitions can be removed, but that in general this is not the case. The work in [11] also provides a construction to eliminate such autonomous transitions, when possible, together with an analysis of its time and memory cost. In this section we investigate when, and at which cost, it is possible to eliminate autonomous transitions in DTA in $\mathbb{A}_g$. We propose the following hierarchy of subclasses $\mathbb{A}_g^{na} \subseteq \mathbb{A}_g^{nc} \subseteq \mathbb{A}_g^{nra} \subseteq \mathbb{A}_g^{rc} \subseteq \mathbb{A}_g$ where:

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

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

No reset and no cycle of autonomous transitions. $\mathbb{A}_g^{nc}$ is the subclass of DTA $\mathcal{A} \in \mathbb{A}_g^{nra}$ in which there is no cycle of autonomous transitions.

No autonomous transitions. $\mathbb{A}_g^{na}$ the subclass of DTA $\mathcal{A} \in \mathbb{A}_g^{nc}$ with no autonomous transitions.

The DTA of Figure 1 belongs to $\mathbb{A}_g^{nc} \backslash \mathbb{A}_g^{na}$ and the DTA of Proposition 1 presented below belongs to $\mathbb{A}_g \backslash \mathbb{A}_g^{rc}$,

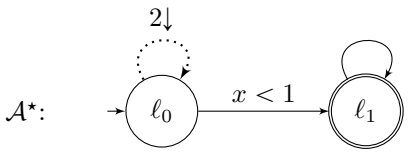
Let us explain why we introduce the intermediate subclasses between $\mathbb{A}_g$ and $\mathbb{A}_g^{na}$. $\mathbb{A}_g^{rc}$ points out which syntactical restrictions must be satisfied by automata in $\mathbb{A}_g$ in order not to extend the expressive power of $\mathbb{A}_g^{na}$. $\mathbb{A}_g^{nra}$ which forbids the clock reset by automous transitions disables the capacity to combine time constants depending on the execution. $\mathbb{A}_g^{nc}$ which in addition forbids loops of autonomous transitions is mainly introduced for simplifying the translations as we will show that it is equivalent to $\mathbb{A}_g^{nra}$ w.r.t. conciseness.

Our results are summarized in the frame below. Let us emphasize the main result: the elimination of autonomous transitions without reset can be performed in polynomial time. This is particularly interesting considering that the same elimination for DTA with guarded locations, according to [11], requires exponential time. When referring to the size of a DTA_g we consider the number of locations and transitions and the size of the formulas associated with transitions.

$$\mathbb{A}_g^{na} \sim_{\mathcal{L}} \mathbb{A}_g^{nc} \sim_{\mathcal{L}} \mathbb{A}_g^{nra} \sim_{\mathcal{L}} \mathbb{A}_g^{rc} \precnsim_{\mathcal{M}} \mathbb{A}_g$$

with $\mathbb{A}_g^{rc}$ exponentially more concise than $\mathbb{A}_g^{nra}$ and a quadratic translation from $\mathbb{A}_g^{nra}$ to $\mathbb{A}_g^{nc}$ and a polynomial translation from $\mathbb{A}_g^{nc}$ to $\mathbb{A}_g^{na}$.

In the framework of DTA with guarded locations, the main result (Theorem 1 in [11]) is that autonomous transitions strictly increase the expressiveness w.r.t. $\prec_{\mathcal{M}}$, and therefore also w.r.t. $\prec_{\mathcal{L}}$. The adaptation to $\mathbb{A}_g$ is immediate, since the automaton $\mathcal{A}^\star$ prsented below, that is used as a counterexample in [11], does not include any boolean expression over atomic proposition.



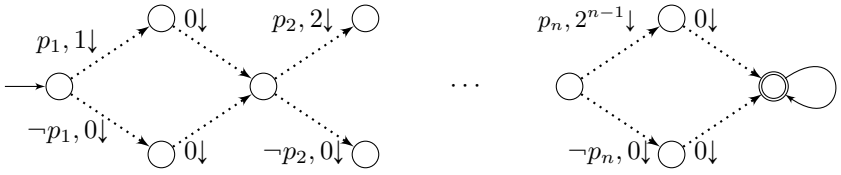
It then trivially follows that:

Proposition 1. *There exists $\mathcal{A}^\star \in \mathbb{A}_g$ such that for all $\mathcal{A} \in \mathbb{A}_g^{na}$ there exists a CTMC $\mathcal{M}$ with $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}) \neq \mathbf{Pr}_{\mathcal{M}}(\mathcal{A}^\star)$. Therefore $\mathbb{A}_g^{na} \not\preceq_{\mathcal{M}} \mathbb{A}_g$.*

• **From $\mathbb{A}_g^{rc}$ to $\mathbb{A}_g^{nra}$.** Observe that due to the loop around location ℓ_0, $\mathcal{A}^\star$ does not belong to $\mathbb{A}_g^{rc}$. The remaining results of the section simultaneously establish that: (1) $\mathbb{A}_g^{rc}$ characterize the DTAs for which autonomous transitions can be eliminated and (2) characterize the cost of this elimination in terms of time complexity and size of produced automatas. First we establish that eliminating autonomous transitions with reset induces an unavoidable exponential blowup.

Proposition 2. *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}_g^{rc}$ and outputs $\mathcal{A}' \in \mathbb{A}_g^{nra}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*
There exists a family $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ in $\mathbb{A}_g^{rc}$ such that the size of $\mathcal{A}_n$ belongs to $O(n^2)$ and for all $\mathcal{A} \in \mathbb{A}_g^{nra}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$, $(|Aut| + 1)|Synch| \geqslant 2^n$.

Proof. (Sketch) The proof is given in appendix. Here we exhibit the family $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ emphasizing that depending on the initial valuation there are 2^n different delays before a sequence of autonomous transitions reaches the final location.



• **From $\mathbb{A}_g^{nra}$ to $\mathbb{A}_g^{nc}$.** Observe that when autonomous transitions do not reset the clock, if a run visits twice the same autonomous transition without visiting synchronized transitions, then no time has elapsed and it will diverge infinitely repeating a cycle of autonomous transitions. The idea of the transformation corresponding to the next proposition consists in duplicating locations by associating a counter to them. This counter represents the number of autonomous transitions visited since the last visit of a synchronized transition (or the beginning of the run). When the counter exceeds the number of autonomous transitions of the DTA, then a cycle has been detected and the run ends up in a deadlock location. The proof of this proposition can be found in appendix

Proposition 3. *There exists an algorithm operating in quadratic time that takes as input $\mathcal{A} \in \mathbb{A}_g^{nra}$ and outputs $\mathcal{A}' \in \mathbb{A}_g^{nc}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

• **From $\mathbb{A}_g^{nc}$ to $\mathbb{A}_g^{na}$.** An interesting feature of specifying propositional formulas on transitions is that the final transformation can be performed in polynomial

time. To this aim we introduce a particular case of decision diagram (DD) for representating formulas as follows. Let DG be a directed acyclic graph rooted in u_0 including a final vertex u_f such that all vertices are reachable from u_0 and can reach u_f as depicted in Figure 2. Every transition is labelled by a formula and the formulas labeling outgoing transitions from a vertex are mutually exclusive (for each variable valuation at most one formula is true). Given a valuation v, $v \models DG$ if there is a path from u_0 to u_f such that $v \models \varphi$ for all φ labeling the transitions of the path. Observe that there is at most one such path. Thus deciding whether $v \models DG$ can be performed in linear time (assuming that the satisfaction of a formula labeling a transition by a valuation can be performed in linear time which is the case for standard representation of formulas).

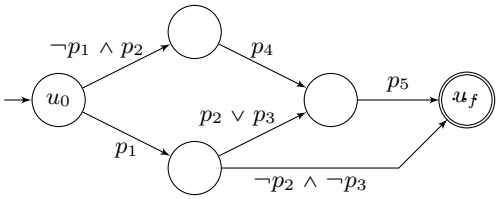


Fig. 2. A DD for formula $(\neg p_1 \wedge p_2 \wedge p_4 \wedge p_5) \vee (p_1 \wedge (p_2 \vee p_3) \wedge p_5) \vee (p_1 \wedge \neg p_2 \wedge \neg p_3)$.

The following proposition eliminates autonomous transitions when they do not reset the clock and there is no cycle made only of autonomous transitions. The associated transformation which is polynomial makes use of DDs for the formulas of the transitions.

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

Proof. The transformation proceeds in three stages

• The first stage consists in duplicating the locations w.r.t. *time regions*. Let $0 = \alpha_0 < \ldots < \alpha_m = C$ be the time constants occurring in $\mathcal{A}$ (adding 0 if necessary). The set of time regions is $\{\alpha_0\},]\alpha_0, \alpha_1[, \{\alpha_1\}, \ldots, \{\alpha_m\},]\alpha_m, \infty[$. For all location ℓ and all region rg, one creates a location $\langle \ell, rg \rangle$. The initial location is $\langle \ell_0, \{\alpha_0\} \rangle$ with ℓ_0 the initial location of $\mathcal{A}$.

For all synchronized transition $\ell \xrightarrow{\varphi^-,\gamma,B,r,\varphi^+} \ell'$ and all regions rg and rg' one creates a transition $\langle \ell, rg \rangle \xrightarrow{\varphi^-,\gamma \wedge x \in rg',B,r,\varphi^+} \langle \ell', rg' \rangle$. For all autonomous transition $\ell \xrightarrow{\varphi,x=i,\sharp,\varnothing} \ell'$ and all region rg, one creates a transition $\langle \ell, rg \rangle \xrightarrow{\varphi,x=i,\sharp,\varnothing} \langle \ell', \{i\} \rangle$. This step is obviously polynomial.

• Let $\mathcal{A}_1$ be the DTA produced by the first stage, the second stage produces a DTA $\mathcal{A}_2$ where the priority of the autonomous transitions is made explicit by restricting the temporal formulas of outgoing transitions. Let $\langle \ell, rg \rangle$ be a

location and $\{t_k = \langle \ell, rg\rangle \xrightarrow{\varphi_k, x=\alpha_k, \sharp, \varnothing} \langle \ell_k, \{\alpha_k\}\rangle\}_{k\leqslant K}$ be the autonomous transitions outgoing from $\langle \ell, rg\rangle$ with $rg \leqslant \alpha_1 \leqslant \cdots \leqslant \alpha_K$ (the other autonomous transitions are useless and are assumed to be deleted).

For all k, one creates an autonomous transition $\langle \ell, rg\rangle \xrightarrow{\varphi_k \wedge \bigwedge_{k'<k} \neg\varphi_{k'}, x=\alpha_k, \sharp, \varnothing} \langle \ell_k, \{\alpha_k\}\rangle$.

For all synchronized transition $\langle \ell, rg\rangle \xrightarrow{\varphi^-, \gamma, B, r, \varphi^+} \langle \ell', rg'\rangle$, one creates a transition
$\langle \ell, rg\rangle \xrightarrow{\varphi^- \wedge \bigwedge_{\alpha_k \leqslant rg'} \neg\varphi_k, \gamma, B, r, \varphi^+} \langle \ell', rg'\rangle$.
Also this step is obviously polynomial. Note that in $\mathcal{A}_2$ the priority of autonomous transitions becomes irrelevant, so $\mathcal{A}_2$ can be treated as DTA with no priority for autonomous transitions. This allows, in the next step, to remove autonomous transitions by aggregating in a single synchronized transition a path of autonomous transitions followed by a synchronized one.

• The final stage that produces $\mathcal{A}'$ from $\mathcal{A}_2$ consists in deleting the autonomous transitions and adding new synchronized transitions as follows. For all $\langle \ell, rg\rangle$ and $\langle \ell', \{i\}\rangle$ such that there is a path of autonomous transitions from $\langle \ell, rg\rangle$ to $\langle \ell', \{i\}\rangle$, and a synchronized transition out of $\langle \ell', \{i\}\rangle$, one specifies the formula $\varphi_{\ell,rg}^{\ell',i}$ by a DD whose vertices are locations both reachable from $\langle \ell, rg\rangle$ by autonomous transitions and can reach $\langle \ell', \{i\}\rangle$ by autonomous transitions. The edges of the DD are the autonomous transitions between such vertices, and the edges are labeled by the formulas of the autonomous transitions (remember that autonomous transitions only have input guards). Clearly the size of this DD is the size of a subgraph of $\mathcal{A}_2$ that includes all paths of autonomous transitions from $\langle \ell, rg\rangle$ to $\langle \ell', \{i\}\rangle$. Then for all synchronized transition $\langle \ell', \{i\}\rangle \xrightarrow{\varphi^-, \gamma, B, r, \varphi^+}$ $\langle \ell'', rg''\rangle$, one creates a transition $\langle \ell, rg\rangle \xrightarrow{\varphi_{\ell,rg}^{\ell',i} \wedge \varphi^-, x\geqslant i \wedge \gamma, B, r, \varphi^+} \langle \ell'', rg''\rangle$.

A new location ℓ'_f is added to $\mathcal{A}'$, ℓ'_F being the unique final location (with its loop), and, for all $\langle \ell_f, rg\rangle$ the transition $\langle \ell_f, rg\rangle \xrightarrow{true, x\geqslant o, Act, \varnothing, true} \ell'_f$. is added to $\mathcal{A}'$.

Illustration of Proposition 4. Figure 1 illustrates how to specify the temporal formula $p\mathbf{U}^{]\alpha,\beta[}q$ with a DTA$_g$ $\mathcal{A} \in \mathbb{A}_g^{nra}$ (and therefore also $\in \mathbb{A}_g^{nc}$) Observe that in the interval $[0, \alpha[$, the current location can only be ℓ_0 with the additional requirement that if an action occurs then p has to be fullfilled inside the whole interval. At time α, the current location can only be ℓ_1 or ℓ_2 depending on the truth value of q and with the guarantee that p holds in the interval $[0, \alpha]$. Considering the first action that occurs after α, there are three possible cases: (1) $p \wedge q$ was satisfied and the formula is satisfied, (2) $p \wedge \neg q$ was satisfied, q is now satisfied and the action occurs before β and so the formula is satisfied (3) $p \wedge \neg q$ was satisfied and it is still satisfied and the action occurs before β and so there is the same possibility to satisfy the formula represented by location ℓ_3.

Figure 3 depicts the DTA $\mathcal{A}' \in \mathbb{A}_g^{na}$ obtained by applying the transformation of Proposition 4 to the DTA $\mathcal{A} \in \mathbb{A}_g^{nc}$ depicted in Figure 1. W.r.t. the defined

transformation, we have done some simplifications. Since ℓ_1 and ℓ_2 can only be entered at time α there is no need to duplicate them. Since ℓ_3 can only be entered in interval $]\alpha, \beta[$ there is no need to duplicate it. In addition we have merged $\langle \ell_0, 0 \rangle$ and $\langle \ell_0,]0, \alpha[\rangle$ since their outgoing transitions are identical (up to the merging). We have also omitted locations that cannot reach the final location. Finally, no DD is necessary since there are no path of two autonomous transitions in the original DTA.

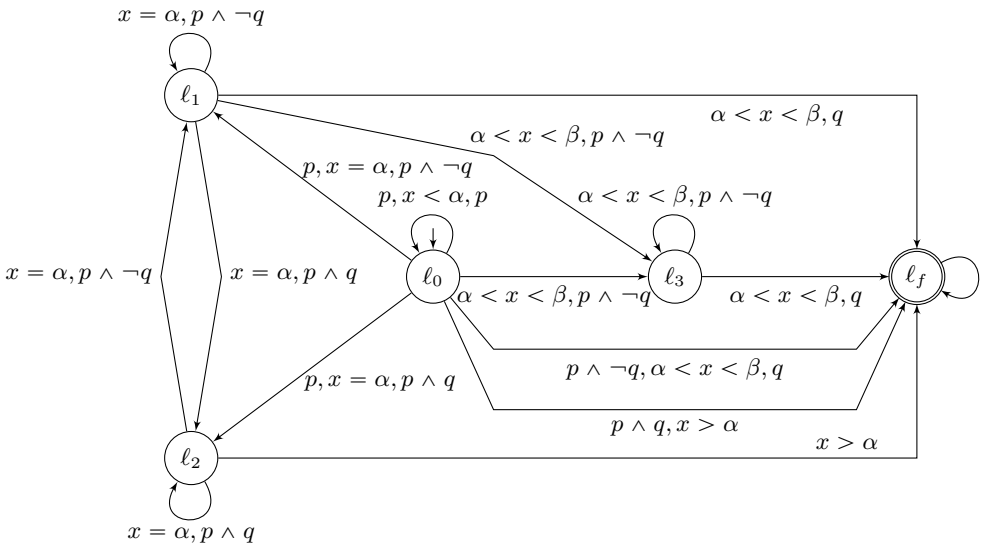


Fig. 3. Another DTA specification of $p\mathbf{U}^{]\alpha,\beta[}q$ with $\alpha > 0$.

4 DTA$_g$ versus DTA$_s$

This section compares the conciseness of guarded transitions versus guarded locations, i.e. comparing $\mathbb{A}_g$ with $\mathbb{A}_s$. We show that a DTA in $\mathbb{A}_s$ can be converted into a DTA in $\mathbb{A}_g$ in a quadratic time, while it takes an exponential time to convert a DTA in $\mathbb{A}_g$ into a DTA in $\mathbb{A}_s$, due to an (unavoidable) exponential growth of locations. A DTA in $\mathbb{A}_s$ has conditions associated only with locations. A transition $(v_i, \delta_i) \xrightarrow{a_i} (v_{i+1}, \delta_{i+1})$ of a timed path is recognized by a transition from location ℓ to ℓ' of a such a DTA only if, given that all the time and action requirements are satisfied (as for $\mathbb{A}_g$) only if $v_{i+1} \models \Lambda(\ell')$, where $\Lambda(\ell')$ is the boolean condition associated with location ℓ'. We briefly recall here the definition of a DTA in $\mathbb{A}_s$, its runs, acceptance of timed path by a run. More explanations and examples can be found in [11].

Definition 7 (DTA). $\mathcal{A} \in \mathbb{A}_s$ *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, $\Lambda : L \to \mathcal{B}_{AP}$ *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* $(\ell_f, \top, Act, \varnothing, \ell_f) \in Synch$.

Definition 8 (Run of $\mathcal{A}$**).** *A* run *of* $\mathcal{A} \in \mathbb{A}_s$ *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* $i \in \mathbb{N}$*:* $\ell_i \in L, l_0 \in L_0, v_i \in \{\top, \bot\}^{AP}, \delta_i \in \mathbb{R}_{\geqslant 0}$*:*

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

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* $\bar{x}_i + \delta_i < x_\sharp$.

Definition 9 (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*$_s$ $\mathcal{A}$*, according to definition 8. 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* $\mathcal{A}$.

We first consider the translation from $\mathbb{A}_s$ to $\mathbb{A}_g$, which mainly consists in shifting the formula of a location to its incoming transitions with a particular handling of the initial locations.

Proposition 5. *There exists an algorithm operating in quadratic time that takes as input* $\mathcal{A}_s \in \mathbb{A}_s$ *and outputs* $\mathcal{A}_g \in \mathbb{A}_g$ *with* $\mathcal{L}(\mathcal{A}_s) = \mathcal{L}(\mathcal{A}_g)$.

Proof. $\mathcal{A}_g$ has the same structure as $\mathcal{A}_s$ except that it has an additional location ℓ_0 which is taken as the initial one.

For all synchronized transition $\ell \xrightarrow{\gamma,B,r} \ell'$ in $\mathcal{A}_s$, $\mathcal{A}_g$ includes the synchronized transition $\ell \xrightarrow{\top,\gamma,B,r,\Lambda(\ell')} \ell'$ and if $\ell \in L_0$ then $\mathcal{A}_g$ includes the synchronized transition $\ell_0 \xrightarrow{\Lambda(\ell),\gamma,B,r,\Lambda(\ell')} \ell'$.
For all autonomous transition $\ell \xrightarrow{x=\alpha,\sharp,r} \ell'$ in $\mathcal{A}_s$, $\mathcal{A}_g$ includes the autonomous transition $\ell \xrightarrow{\Lambda(\ell'),x=\alpha,\sharp,r} \ell'$ and if $\ell \in L_0$ then $\mathcal{A}_g$ includes the autonomous transition: $\ell_0 \xrightarrow{\Lambda(\ell)\wedge\Lambda(\ell'),x=\alpha,\sharp,r} \ell'$.
The quadratic factor is due to the substitution of the $|L|$ formulas of $\mathcal{A}_s$ by at least $|E|$ formulas in $\mathcal{A}_g$.

The reverse translation is more costly and consists in duplicating a location w.r.t. the guards of the incoming and outgoing transitions.

Proposition 6. *There exists an algorithm operating in exponential time that takes as input $\mathcal{A}_g \in \mathbb{A}_g$ and outputs $\mathcal{A}_s \in \mathbb{A}_s$ with $\mathcal{L}(\mathcal{A}_g) = \mathcal{L}(\mathcal{A}_s)$.*

Proof. Given $\ell \in L$, let $\varphi^\ell_1, \dots \varphi^\ell_{n_\ell}$ be the formulas of entering guards of transitions incoming ℓ and exiting guards of transitions outgoing ℓ. Then $L_s = \{\langle \ell, I\rangle \mid \ell \in L \wedge I \subseteq \{1, \dots, n_\ell\}\} \cup \{\ell^*_f\}$ where ℓ^*_f is the final state (fulfilling the requirements of a DTA in $\mathbb{A}_s$) and for all $\langle \ell, I\rangle$, $\Lambda(\langle \ell, I\rangle) = \bigwedge_{i\in I} \varphi^\ell_i \wedge \bigwedge_{i\notin I} \neg\varphi^\ell_i$.

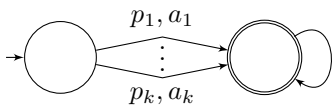
For all synchronized transition $\ell \xrightarrow{\varphi^\ell_i,\gamma,B,r,\varphi^{\ell'}_{i'}} \ell'$ in $\mathcal{A}_g$ and all I, I' such that $i \in I$ and $i' \in I'$, $\mathcal{A}_s$ includes the synchronized transition: $\langle \ell, I\rangle \xrightarrow{\gamma,B,r} \langle \ell', I'\rangle$.
For all autonomous transition $\ell \xrightarrow{\varphi^\ell_i,x=\alpha,\sharp,r} \ell'$ in $\mathcal{A}_g$ and all I, I' such that $i \in I$ and $i' \in I'$ with $\varphi^\ell_i = \varphi^{\ell'}_{i'}$, $\mathcal{A}_s$ includes the autonomous transition: $\langle \ell, I\rangle \xrightarrow{x=\alpha,\sharp,r} \langle \ell', I'\rangle$.
For all $\langle \ell_f, I\rangle$, there is a transition $\langle \ell_f, I\rangle \xrightarrow{\top,Act,\varnothing} \ell^*_f$.

Proposition 5 and 6 above can be trivially extended to sub-classes of $\mathbb{A}_s$ and $\mathbb{A}_g$, because the proofs are general and do not involve creation of autonomous transitions.

The exponential blowup due to the duplication of locations is unavoidable even without timing considerations, as shown by the next proposition.

Proposition 7. *There exists a family of automata $\{\mathcal{A}^k_g\}_{k\in\mathbb{N}}$ in $\mathbb{A}^{na}_g$ such that the size of $\mathcal{A}^k_g$ belongs to $O(k\log(k))$ and for all $\mathcal{A}_s \in \mathbb{A}_s$ with $\mathcal{L}(\mathcal{A}_s) = \mathcal{L}(\mathcal{A}^k_g)$ the number of its locations is at least $2^k - 1$.*

Proof. Consider the automaton $\mathcal{A}^k_g$ described below.



This automaton accepts timed paths whose first action may be a_i only if the initial state fullfills p_i. Consider in $\mathcal{A}_s$ the locations reached at time 0 by the runs before the first action is performed. At least $2^k - 1$ initial valuations must reach such a location. Assume that $\mathcal{A}_s$ has less than $2^k - 1$ locations. Then two initial valuations reach the same location. Let p_i be some proposition on which they differ. Thus there exists a initial valuation v with $v(p_i) = \bot$ such that a timed path starting with a_i is accepted which yields a contradiction.

5 Conclusion and future work

The results of this paper, together with those of a companion paper [11] allows to build a better understanding of DTAs and of the various CSL^{TA} definitions.

We have established that DTAs with autonomous transitions are more expressive that DTAs without autonomous transitions when there are cycles made only of autonomous transitions on which there is at least a reset, irrespectively of whether guards are associated to locations or to transitions.

Secondly, even when autonomous transitions do not enhance expressiveness, they improve conciseness: if feasible, removing autonomous transitions may lead to an exponential blow up of the DTA.

Finally, removing autonomous transitions from a DTA in $\mathbb{A}_g$ is less expensive than doing it for a DTA in $\mathbb{A}_g$. In particular to remove autonomous transitions from a DTA with no reset on autonomous transitions (i.e. belonging to $\mathbb{A}_g^{nra}$) is polynomial if decision diagrams are used to represent propositional formulas, while the analogous operations for a DTA belonging to $\mathbb{A}_g^{nra}$ is exponential. This result has motivated a throughout comparison of DTA_s and DTA_g, that has shown that guards on transitions may lead to more concise DTAs: indeed the translation from $\mathbb{A}_g$ to $\mathbb{A}_s$ is exponential, while the opposite translation is quadratic.

Various types of DTAs have been used for the definition of the stochastic logic CSL^{TA}. We can now assert that CSL^{TA} definitions that include autonomous transitions are more expressive than CSL^{TA} that do not. The counter-example of the proof of Proposition 1 has a clear interpretation in terms of periodic behaviour, showing that CSL^{TA} without autonomous transitions are not adequate to express certain periodicity properties. We can also state that CSL^{TA} specifications that include guarded transitions can be more concise than CSL^{TA} that considers guarded locations. Since the number of locations may significantly affect the complexity of model-checking CSL^{TA}, it is future work to investigate how the component-based model-checking of CSL^{TA} [2] can take advantage of the results of this paper to lower the cost of model-checking.

References

1. M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley & Sons, 1995.
2. E.G. Amparore and S. Donatelli. Efficient model checking of the stochastic logic cslta. *Performance Evaluation*, 123-124:1–34, 2018.
3. Elvio Gilberto Amparore, Paolo Ballarini, Marco Beccuti, Susanna Donatelli, and Giuliana Franceschinis. Expressing and computing passage time measures of GSPN models with HASL. In *Proceedings of PETRI NETS 2013, Milan, Italy*, volume 7927 of *LNCS*, pages 110–129. Springer, 2013.
4. Elvio Gilberto Amparore and Susanna Donatelli. MC4CSL$^{\text{TA}}$: an efficient model checking tool for CSL$^{\text{TA}}$. In *QEST 2010*, pages 153–154. IEEE Computer Society.
5. Adnan Aziz, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.
6. Christel Baier, Lucia Cloth, Boudewijn R. Haverkort, Matthias Kuntz, and Markus Siegle. Model Checking Markov Chains with Actions and State Labels. *IEEE TSE*, 33:209–224, 2007.
7. Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. *On the Logical Characterisation of Performability Properties*, pages 780–792. Springer Berlin Heidelberg, 2000.
8. Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-Checking Algorithms for Continuous-Time Markov Chains. *IEEE TSE*, 29(6):524–541, 2003.
9. Paolo Ballarini, Benoît Barbot, Marie Duflot, Serge Haddad, and Nihal Pekergin. HASL: A new approach for performance evaluation and model checking from concepts to experimentation. *Perform. Eval.*, 90:53–77, 2015.
10. Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. Model checking of continuous-time Markov chains against timed automata specifications. *Logical Methods in Comp. Science*, 7(1), 2011.
11. Susanna Donatelli and Serge Haddad. Expressiveness and conciseness of timed automata for the verification of stochastic models. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, *Proceedings of the 14th International Conference on Language and Automata Theory and Applications (LATA'20)*, volume 12038 of *Lecture Notes in Computer Science*, pages 170–183, Milan, Italy, March 2020. Springer.
12. Susanna Donatelli, Serge Haddad, and Jeremy Sproston. Model checking timed and stochastic properties with CSL$^{\text{TA}}$. *IEEE TSE*, 35(2):224–240, 2009.
13. Matthias Kuntz and Boudewijn R Haverkort. GCSRL-a logic for stochastic reward models with timed and untimed behaviour. In *8th PMCCS*, pages 50–56, 2007.
14. John F. Meyer, A. Movaghar, and William H. Sanders. Stochastic activity networks: Structure, behavior, and application. In *Int. Workshop on Timed Petri Nets*, pages 106–115. IEEE CS, 1985.
15. Linar Mikeev, Martin R. Neuhäußer, David Spieler, and Verena Wolf. On-the-fly verification and optimization of DTA-properties for large Markov chains. *Formal Methods in System Design*, 43(2), 2013.
16. W.Douglas Obal II and William Sanders. State-space support for path-based reward variables. *Performance Evaluation*, 35:233–251, 05 1999.

6 Appendix

Proposition 2. *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}_g^{rc}$ and outputs $\mathcal{A}' \in \mathbb{A}_g^{nra}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*
There exists a family $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ in $\mathbb{A}_g^{rc}$ such that the size of $\mathcal{A}_n$ belongs to $O(n^2)$ and for all $\mathcal{A} \in \mathbb{A}_g^{nra}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$, $(|Aut| + 1)|Synch| \geqslant 2^n$.

Proof. Consider an elementary path of $\mathcal{A}$ not including synchronized transitions with reset or with guard $x > C$. Define its *delay* to be the sum of constants occurring in its autonomous transitions with reset. Let K be the maximal delay of such paths.
The set of locations of $\mathcal{A}'$ is $L' = \{\langle \ell, i\rangle \mid 0 \leqslant i \leqslant K \wedge \ell \in L\backslash\{\ell_f\}\} \cup \{\ell_f\}$ with its initial location $\langle \ell_0, 0\rangle$ and final location ℓ_f (with its single looping transition).
Let γ be a guard. Define $\gamma + i$ the guard where all constants of γ are increased by i.

For all synchronized transition $\ell \xrightarrow{\varphi^-,\gamma,B,r,\varphi^+} \ell'$ $(\ell \neq \ell_f)$ of $\mathcal{A}$ and $i \leqslant K$:

- if $\ell' = \ell_f$ then there is a synchronized transition $\langle \ell, i\rangle \xrightarrow{\varphi^-,\gamma+i,B,r,\varphi^+} \ell_f$;
- else if $r =\downarrow$ or $\gamma = x > C$ then there is a synchronized transition $\langle \ell, i\rangle \xrightarrow{\varphi^-,\gamma+i,B,r,\varphi^+} \langle \ell', 0\rangle$;
- otherwise there is a synchronized transition $\langle \ell, i\rangle \xrightarrow{\varphi^-,\gamma+i,B,r,\varphi^+} \langle \ell', i\rangle$.

For all autonomous transition $\ell \xrightarrow{\varphi^-,x=c,\sharp,r} \ell'$ of $\mathcal{A}$ and $i \leqslant K$:

- if $\ell' = \ell_f$ then there is an autonomous transition $\langle \ell, i\rangle \xrightarrow{\varphi^-,x=c+i,\sharp,\varnothing} \ell_f$;
- else if $r = \varnothing$ there is an autonomous transition $\langle \ell, i\rangle \xrightarrow{\varphi^-,x=c+i,\sharp,\varnothing} \langle \ell', i\rangle$;
- else if $r =\downarrow$ and $c + i \leqslant K$ then there is an autonomous transition $\langle \ell, i\rangle \xrightarrow{\varphi^-,x=c+i,\sharp,\varnothing} \langle \ell', c + i\rangle$.

Observe that $\mathcal{A}'$ has no more autonomous transitions with reset.
For sake of simplicity, let $\langle \ell_f, i\rangle$ denote ℓ_f for all i. SD: where do the locations $\langle \ell_f, i\rangle$ come from? By construction I think there is only ℓ_f

Let us establish the correctness of this transformation.
• Let $\rho = (\ell_0, v_0, \bar{x}_0, \delta_0) \xrightarrow{\varphi_0^-,\gamma_0,B_0,r_0,\varphi_0^+} (\ell_1, v_1, \bar{x}_1, \delta_1) \cdots \xrightarrow{\varphi_{n-1}^-,\gamma_{n-1},B_{n-1},r_{n-1},\varphi_{n-1}^+} (\ell_n, v_n, \bar{x}_n, \delta_n) \cdots$ be a run of $\mathcal{A}$.
For all k define r'_k by $r'_k = \varnothing$ if $B_k = \sharp$ and $r'_k = r_k$ otherwise.
For all k define c_k, d_k by $c_0 = d_0 = 0$ and for $k > 0$:

- if $r_k = \varnothing$ then $c_k = c_{k-1}$;
- if $B_k \neq \sharp$, $r_k = \varnothing$ and $\gamma_k \neq x > C$ then $d_k = d_{k-1}$;
- if $B_k \neq \sharp$, $r_k = \varnothing$ and $\gamma_k = x > C$ then $d_k = 0$;
- if $B_k \neq \sharp$ and $r_k =\downarrow$ then $c_k = d_k = 0$;
- if $B_k = \sharp$ and $r_k = \varnothing$ then $d_k = d_{k-1}$;
- if $B_k = \sharp$, $r_k =\downarrow$ and $\gamma_k = x = c$ then $c_k = d_k = d_{k-1} + c$

The integer c_k represents the difference between the values of the clock in the k^{th} states of the original and simulating runs while d_k represents how c_k has to be taken into account in the k^{th} state of the simulating run. In general, d_k is

equal to c_k except when the original run goes through a guard $x > C$ in which case the difference between the clock values is irrelevant.

Then $\rho' = (\langle \ell_0, 0\rangle, v_0, \bar{x}_0, \delta_0) \xrightarrow{\varphi_0^-, \gamma_0 + d_0, B_0, r'_0, \varphi_0^+} (\langle \ell_1, d_1\rangle, v_1, \bar{x}_1 + c_1, \delta_1) \cdots \xrightarrow{\varphi_{n-1}^-, \gamma_{n-1} + d_{n-1}, B_{n-1}, r'_{n-1}, \varphi_{n-1}^+} (\langle \ell_n, d_n\rangle, v_n, \bar{x}_n + c_n)$ is a run of $\mathcal{A}'$. The main point is that all transitions in ρ can be mimicked due to the choice of K.

• Conversely let $\rho' = (\langle \ell_0, 0\rangle, v_0, \bar{x}'_0, \delta_0) \xrightarrow{\varphi_0^-, \gamma_0, B_0, r'_0, \varphi_0^+} (\langle \ell_1, d_1\rangle, v_1, \bar{x}'_1, \delta_1) \cdots \xrightarrow{\varphi_{n-1}^-, \gamma_{n-1} + d_{n-1}, B_{n-1}, r'_{n-1}, \varphi_{n-1}^+} (\langle \ell_n, d_n\rangle, v_n, \bar{x}'_n, \delta_n) \cdots$ be a run of $\mathcal{A}'$.

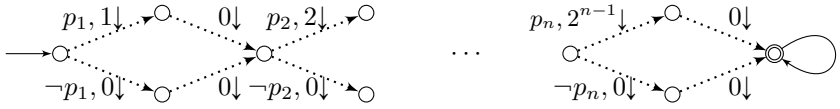
For all k define r_k as the original reset that associated with the creation of the transition labelled by $\gamma_k + d_{k-1}, A_n, r'_k$. SD: Shouldn't it be B_k? I think we are also missing φ_k^- and φ_k^+.

For all k define c_k by $c_0 = 0$ and for $k > 0$:

- if $r_k = \varnothing$ then $c_k = c_{k-1}$;
- if $r_k = r'_k = \downarrow$ then $c_k = 0$;
- if $B_k = \sharp$, $r_k = \downarrow$ and $\gamma_k = x = c$ then $c_k = c_{k-1} + c$.

Then $\rho = (\ell_0, v_0, \bar{x}'_0 - c_0, \delta_0) \xrightarrow{\varphi_0^-, \gamma_0, B_0, r_0, \varphi_0^+} (\ell_1, v_1, \bar{x}'_1 - c_1) \cdots \xrightarrow{\varphi_{n-1}^-, \gamma_{n-1}, B_{n-1}, r_{n-1}, \varphi_{n-1}^+} (\ell_n, v_n, \bar{x}'_n - c_n, \delta_n) \cdots$ is a run of $\mathcal{A}$.

We now establish the second assertion of the proposition. Consider the automaton $\mathcal{A}_n$ described below. Here Act is a singleton and so we omit the labels of the actions. SD: we should take care, since the compact notation used below over edges has not been used in the examples of Fig. 1 and 2. Here Act is a singleton and we compactly represent edge inscription by $(p_i, k, \downarrow)$ for $(p_i, x = k, Act, \downarrow, true$. Note that all transitions are autonomous and have a reset



Given a valuation v of the atomic propositions, let us define the integer $z(v) \in [0, 2^n[$ by: $z(v) = \sum_{i \leqslant n} 2^{i-1} \mathbf{1}_{v(p_i)=\top}$. SD: I think before we wrote $v \models p_i$ instead of $v(p_i) = \top$.

Observe that z is a one-to-one mapping. Then $\mathcal{A}_n$ accepts the timed paths starting with some initial valuation v such that the first action occurs not earlier than $z(v)$.

Assume by contradiction that there exists $\mathcal{A} \in \mathbb{A}_g^{nra}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$ such that $(|Aut|+1)|Synch| < 2^n$. Consider 2^n accepted timed paths σ_v starting with the 2^n possible valuations v such that the first action occurs at time $z(v)$. Let $t_v = \ell_v \xrightarrow{\varphi_v^-, \gamma_v, B_v, r_v, \varphi_v^+} \ell'_v$ be the synchronized transition in $\mathcal{A}$ corresponding to this action and, $\bar{x}_v$ be the clock valuation when entering ℓ_v. Observe that this clock valuation is either 0 or a time constant occurring in an autonomous transition. Due to the assumption, there are two different valuations v and v' with $t_v = t_{v'}$ and $\bar{x}_v = \bar{x}_{v'}$. W.l.o.g. let $z(v) < z(v')$. Consider a timed path starting with valuation v' and whose first action occurs at time $z(v)$. Such a path is accepted by $\mathcal{A}$ using the accepting run for $\sigma_{v'}$ up to $\ell_{v'} = \ell_v$ with clock

valuation $\bar{x}_{v'} = \bar{x}_v$ and then going on with the suffix of the accepting run for σ_v. However since $z(v) < z(v')$ such a path does not belong to $\mathcal{L}(\mathcal{A}_n)$.

□

Proposition 3. *There exists an algorithm operating in quadratic time that takes as input $\mathcal{A} \in \mathbb{A}_g^{nra}$ and outputs $\mathcal{A}' \in \mathbb{A}_g^{nc}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof. Let $\mathcal{A} \in \mathbb{A}_g^{nra}$. Observe that if a timed path visits twice the same autonomous transition without in the meantime visiting a synchronized transition, then it will infinitely cycles visiting only autonomous transitions (i.e. diverging). Let K be the number of autonomous transitions. $\mathcal{A}'$ is built as follows.
The set of locations of $\mathcal{A}'$ is $L' = \{(\ell, i) \mid 0 \leqslant i \leqslant K \wedge \ell \in L \backslash \{\ell_f\}\} \cup \{\ell_f, \ell_\perp\}$ with initial location $(\ell_0, 0)$ and final location ℓ_f(with its single looping transition).

For all synchronized transition $\ell \xrightarrow{\varphi^-, \gamma, B, r, \varphi^+} \ell'$ of $\mathcal{A}$ and $i \leqslant K$:

- if $\ell' = \ell_f$ then there is a synchronized transition $(\ell, i) \xrightarrow{\varphi^-, \gamma, B, r, \varphi^+} \ell_f$;
- otherwise there is a synchronized transition $(\ell, i) \xrightarrow{\varphi^-, \gamma, B, r, \varphi^+} (\ell', 0)$.

For all autonomous transition $\ell \xrightarrow{\varphi^-, x=c, \sharp, \varnothing} \ell'$ of $\mathcal{A}$ and $i \leqslant K$:

- if $i = K$ then there is an autonomous transition $(\ell, i) \xrightarrow{\varphi^-, x=c, \sharp, \varnothing} \ell_\perp$;
- else if $\ell' = \ell_f$ then there is an autonomous transition $(\ell, i) \xrightarrow{\varphi^-, x=c, \sharp, \varnothing} \ell_f$;
- otherwise there is an autonomous transition $(\ell, i) \xrightarrow{\varphi^-, x=c, \sharp, \varnothing} (\ell', i+1)$.

By construction, there is no cycle of autonomous transitions in $\mathcal{A}'$.
A timed path of $\mathcal{A}'$ that does not visit $\ell_\perp$ provides a timed path of $\mathcal{A}$ by omitting the second component of the locations while the only timed paths of $\mathcal{A}$ that cannot be mimicked by $\mathcal{A}'$ are the diverging ones. Furthermore the "partial" simulation of such paths by $\mathcal{A}'$ leads to a deadlock location.

□