Autonomous Transitions Enhance CSLTA Expressiveness and Conciseness

Susanna Donatelli, Serge Haddad

Autonomous Transitions Enhance CSL^TA Expressiveness and Conciseness

Susanna Donatelli
Dipartimento di Informatica, Università di Torino, Torino, Italy, donatelli@di.unito.it

Serge Haddad
LSV, ENS Paris-Saclay, CNRS, Inria, Université Paris-Saclay, France, haddad@lsv.fr

—— **Abstract** ——

CSL^TA is a stochastic temporal logic for continuous-time Markov chains (CTMC) where formulas similarly to those of CTL* are inductively defined by nesting of timed path formulas and state formulas. In particular a timed path formula of CSL^TA is specified by a single-clock Deterministic Timed Automaton (DTA). Such a DTA features two kinds of transitions: *synchronizing* transitions triggered by CTMC transitions and *autonomous* transitions triggered by time elapsing that change the location of the DTA when the clock reaches a given threshold. It has already been shown that CSL^TA strictly includes stochastic logics like CSL and asCSL. An interesting variant of CSL^TA consists in equipping transitions rather than locations by boolean formulas. Here we answer the following question: do autonomous transitions and/or boolean guards on transitions enhance expressiveness and/or conciseness of DTAs? We show that this is indeed the case. In establishing our main results we also identify an accurate syntactical characterization of DTAs for which the autonomous transitions do not add expressive power but lead to exponentially more concise DTAs.

1 Introduction

Stochastic logics like CSL [5] allow 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 that 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 [12] 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 [15] for Stochastic Activity Networks [13], while hybrid automata have been used to define rather complex forms of passage time [3] 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 logic CSL^TA [11] 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. Autonomous transitions have priority over synchronizing ones. The determinism requirement is introduced so that the synchronized product of the DTA and the CTMC is still a stochastic process as all sources

of non-determinism are eliminated. Furthermore the single clock restriction implies that this stochastic process is a Markov Regenerative Process whose performance indice related to the satisfaction of the formula can be efficiently computed [2, 4]. It has been shown [11] that CSL^{TA} strictly includes stochastic logics like CSL and asCSL. Various extensions of CSL^{TA} have been presented in the literature. DTA with multiple clocks has been introduced [10]. In that paper a different definition of DTA is used: the state labels of the CTMC act as input symbols of the DTA and autonomous transitions are not considered. The work in [14] addresses the model checking of single-clock CSL^{TA} in the context of infinite state Markov chains. The definition of DTA differs from the ones in [11] and in [10]. Autonomous transitions and actions are not considered and each DTA transition has an associated pair of boolean guards (*pre* and *post*). The transition can then be taken only when the CTMC moves from a state that satisfies *pre* to a state that satisfies *post*.

Our contributions. This paper addresses two research questions aimed at investigating whether there are substantial differences in the various definitions of CSL^{TA}. The first one is *whether the presence of autonomous transitions enhances the conciseness and 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 with reset do enhance expressiveness, and provide an hierarchy of subclasses of DTA that do not extend expressiveness. We show that w.r.t. conciseness some subclasses are equivalent, while eliminating autonomous transitions from the larger subclass leads to an exponential increase in DTA size. The increase of expressive power has been proved in the less discriminating situation of quantitative comparison (it is indeed possible that two DTAs with different timed languages have the same probability of accepted paths for all Markov chains). This result uses Laplace transforms (often applied to simplify computations) which, to our knowledge, have never been considered for proofs of expressiveness.

The second question is *whether the use of boolean guards may change the above set of results*. We show that, whatever the considered subclass, DTA with boolean guards are exponentially more concise than DTA with location formulas. Then we prove that all expressiveness results are preserved but one of the transformation with exponential blow-up can now be performed with a polynomial blow-up. This last point requires the use of decision diagrams to represent formulas in the transformed DTA.

The paper is organized as follows: Section 2 introduces the necessary background and the required definitions. Section 3 answers the first research question and Section 4 the second one. Section 5 summarizes the results and outlines practical implications and possible investigations. All proofs not in the text are collected in the appendix.

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 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 2** (**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, 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 3, 5 and 7), 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, the current location, say ℓ, is matched with the current state of the system, say (v_i, δ_i). The mapping Λ 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_i, 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 $\delta \leqslant \delta_i$, 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 after delay δ, ℓ' is matched with s and delay δ_i is decreased by δ.
- Else if there is a *synchronizing transition* outgoing from ℓ such that (1) after time δ_i has elapsed its inner condition (say $\alpha \bowtie x \bowtie' \beta$) is satisfied, (2) the action a 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 after delay δ_i, ℓ' is matched with s'.
- 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 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 must 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 3** (**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, $\Lambda : L \to \mathcal{B}_{AP}$ 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 action.** $\forall B, B' \subseteq Act\, s.t.\, B \cap B' \neq \varnothing, \forall \ell, \ell', \ell'' \in L,$
  $if\ \ell \xrightarrow{\gamma, B, r} \ell'\ and\ \ell \xrightarrow{\gamma', B', r'} \ell''\ then\ \ \Lambda(\ell') \wedge \Lambda(\ell'') \Leftrightarrow \bot\ or\ \gamma \wedge \gamma' \Leftrightarrow \bot.$
- **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 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.
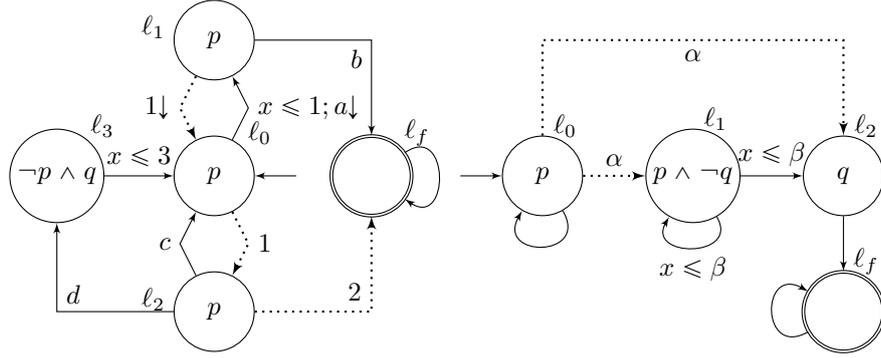
Figure 1 Two examples of DTA.

▸ **Example 4** (**DTA examples**). 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 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 guards (like $x \geq 0$) 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.

Figure 1 also shows, on the right, the DTA for the CSL timed-bounded Until $pU^{[\alpha, \beta]}q$.

Readers familiar with the definition of DTAs in [11] may have recognized that the above definition has been slightly changed. Apart from some syntactical changes, the only difference is that we allow cycles made only of autonomous transitions. This restriction was imposed in [11] to ease the model-checking procedure implementation.

▸ **Definition 5** (**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* $i \in \mathbb{N}$*:* $\ell_i \in L, \ell_0 \in L_0, v_i \in \{\top, \bot\}^{AP}, \delta_i \in \mathbb{R}_{\geq 0}$,

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

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

▸ **Example 6** (**DTA run**). In the run we, again, describe v in terms of the subset of AP that evaluate to $\top$. For the left DTA of Figure 1 the following is a possible run:
$(\ell_0, \{p\}, \bar{x}_0 = 0.0, \delta_0 = 0.2) \xrightarrow{x \leq 1, \{a\}, \downarrow} (\ell_1, \{p, q\}, 0.0, 0.7) \xrightarrow{x \geq 0, \{b\}, \varnothing} (\ell_f, \{q\}, 0.7, 1.0)$
$\xrightarrow{x \geq 0, Act, \varnothing} (\ell_f, \{p\}, 1.7, 3.3) \xrightarrow{x \geq 0, Act, \varnothing} (\ell_f, \{p\}, 5.0, 2.2) \cdots$
Another possible run is: $(\ell_0, p, 0.0, 0.2) \xrightarrow{x \leq 1, \{a\}, \downarrow} (\ell_1, \{p, q\}, 0.0, 1.0)$
$\xrightarrow{x = 1, \sharp, \downarrow} (\ell_0, \{p, q\}, 0.0, 1.0) \xrightarrow{x = 1, \sharp, \varnothing} (\ell_2, \{p, q\}, 1.0, 0.0) \xrightarrow{x \geq 0, \{c\}, \varnothing} (\ell_0, \{p, q\}, 1.0, 0.0) \cdots$

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 7** (**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* $\delta_i = \sum_{\kappa(i-1) < h \leq \kappa(i)} \delta'_h$
- $\forall h, \kappa(i-1) < h \leq \kappa(i) \Rightarrow v'_h = v_i$ *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* $\mathcal{A}$.

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

▸ **Example 8** (**Timed path recognized by a DTA run**). A timed path $\sigma = (p, 0.2) \xrightarrow{a} (\{p, q\}, 0.7) \xrightarrow{b} (q, 1.0) \xrightarrow{c} \cdots$ is recognized in the DTA on the left of Figure 1 by the first run of Example 6, with a one-to-one mapping κ. Since ℓ_f is visited by this run, $\sigma \in \mathcal{L}(\mathcal{A})$.
A timed path $\sigma = 0{:}(\{p, q\}, 0.5) \xrightarrow{a} 1{:}(p, 0.6) \xrightarrow{b} 2{:}(\{p, q\}, 4.0) \xrightarrow{c} 3 {:}(p, \delta) \cdots$ is recognized by the DTA on the right of Figure 1, assuming $\alpha = 2$ and $\beta = 5$, through the run $\rho = 0{:}(\ell_0, \{p, q\}, \bar{x}_0 = 0.0, \delta_0 = 0.5) \xrightarrow{Act} 1{:}(\ell_0, p, 0.5, 0.6) \xrightarrow{Act} 2{:}(\ell_0, \{p, q\}, 1.1, 0.9) \xrightarrow{x = 2, \sharp, \varnothing} 3{:}(\ell_2, \{p, q\}, 2.0, 3.1) \xrightarrow{Act} 4{:}(\ell_f, p, 5.1, \delta) \cdots$
The mapping κ fulfills $\kappa(0) = 0, \kappa(1) = 1, \kappa(2) = 3, \kappa(3) = 4, \ldots$
Since ℓ_f is visited by this run, the path is accepted.

DTA have been used to identify sets of behaviours of Continuous Time Markov Chains (CTMCs) that satisfy path formulas of CSL$^{\text{TA}}$, as recalled in the introduction. We shall consider CTMCs in which states are associated with a valuation of a set of propositions AP and a change of states is associated with some actions in Act. This leads to the following representation for CTMCs.

▸ **Definition 9** (**CTMC representation**). *A continuous time Markov chain with state and action labels is represented by a tuple* $\mathcal{M} = \langle S, s_0, Act, AP, lab, R \rangle$*, 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* $\lambda = R(s, a, s') \wedge \lambda > 0$, *we write* $s \xrightarrow{a,\lambda} s'$.

We assume that each state has at least one successor: for all $s \in S$, 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]).

$\mathbb{A}$ denotes the whole family of automata of Definition 3. We identify several subclasses of $\mathbb{A}$, characterized by either the absence of autonomous transitions ($\mathbb{A}^{na}$) or to a limited presence of such transitions ($\mathbb{A}^{nc} \subseteq \mathbb{A}^{nar} \subseteq \mathbb{A}^{rc}$), as follows:

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 includes a synchronizing transition $(\ell, \gamma, B, r, \ell')$ with $r = \downarrow$ or $\gamma = (x > C)$.

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

No reset and no cyle of autonomous transitions $\mathbb{A}^{nc}$ is the subclass of the automata $\mathcal{A} \in \mathbb{A}^{nar}$ in which there is no cycle made only of autonomous transitions.

No autonomous transitions $\mathbb{A}^{na}$ is the subclass of the automata $\mathcal{A} \in \mathbb{A}^{nc}$ with no autonomous transitions: $\mathbb{A}^{na} = \{\mathcal{A} \in \mathbb{A} \mid Aut(\mathcal{A}) = \varnothing\}$

The DTA on the left of Figure 1 belongs to $\mathbb{A}^{rc} \backslash \mathbb{A}^{nar}$, while the one on the right belongs to $\mathbb{A}^{nc} \backslash \mathbb{A}^{na}$. Any DTA with no reset on autonomous transitions but with a cycle of autonomous transitions is an example of $\mathbb{A}^{nar} \backslash \mathbb{A}^{nc}$.

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

3 The role of autonomous transitions

We aim at comparing the expressive power of the families of DTA introduced above. We adopt both qualitative and quantitative points of view. From a qualitative point of view, we consider the timed path languages that they generate. From a quantitative point of view, we consider the accepting probabilities associated with the Markov chains.

▸ **Definition 10.** *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* $\mathbb{A}_1 <_{\mathcal{L}} \mathbb{A}_2$, *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* $\mathbb{A}_1 <_{\mathcal{M}} \mathbb{A}_2$, *if for all* $\mathcal{A}_1 \in \mathbb{A}_1$ *there exists* $\mathcal{A}_2 \in \mathbb{A}_2$ *such that for all Markov chain* $\mathcal{M}$, $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}_2) = \mathbf{Pr}_{\mathcal{M}}(\mathcal{A}_1)$.

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 $\mathbb{A}_1 <_{\mathcal{L}} \mathbb{A}_2$ and $\mathbb{A}_2 <_{\mathcal{L}} \mathbb{A}_1$ (resp. $\mathbb{A}_1 <_{\mathcal{M}} \mathbb{A}_2$ and $\mathbb{A}_2 <_{\mathcal{M}} \mathbb{A}_1$). $\mathbb{A}_2$ is *strictly more expressive than* $\mathbb{A}_1$ w.r.t. language (resp. Markov chains), denoted $\mathbb{A}_1 \lesssim_{\mathcal{L}} \mathbb{A}_2$ (resp. $\mathbb{A}_1 \lesssim_{\mathcal{M}} \mathbb{A}_2$) if $\mathbb{A}_1 <_{\mathcal{L}} \mathbb{A}_2$

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 $\mathbb{A}_1 \prec_{\mathcal{M}} \mathbb{A}_2$.

The following frame summarizes the results of this section. The expressiveness results are the strongest possible as the equivalences are proved for the qualitative relation while the strict relation is established for the quantitative relation.

$$\mathbb{A}^{na} \sim_{\mathcal{L}} \mathbb{A}^{nc} \sim_{\mathcal{L}} \mathbb{A}^{nar} \sim_{\mathcal{L}} \mathbb{A}^{rc} \precsim_{\mathcal{M}} \mathbb{A}$$
with $\mathbb{A}^{rc}$ (resp. $\mathbb{A}^{nc}$) exponentially more concise than $\mathbb{A}^{nar}$ (resp. $\mathbb{A}^{na}$)
and a quadratic translation from $\mathbb{A}^{nar}$ to $\mathbb{A}^{nc}$.

- **Relation between $\mathbb{A}$ and $\mathbb{A}^{na}$.**

First we establish that autonomous *resetting* transitions extend the expressive power of DTA w.r.t. Markov chains. The weaker result w.r.t. language is shown in the appendix since the associated proof is considerably simpler.

To this aim, we 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 valuation $\bar{x}$, the clock valuation $sc(\ell, \bar{x})$ is defined by:

- Let $i = \min(j \mid j \in \{1, \dots, C\} \land \bar{x} < j)$ with $\min(\varnothing) = C+1$;
- 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 ith 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 $G = (V, E)$ of such DTA as follows.

- V, the set of vertices, is defined by $V = \{(\ell, i) \mid \ell \in L \land 0 \leqslant i \leqslant C+1\}$;
- 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 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 11.** *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:*

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

▸ **Proposition 12.** *Let $\mathcal{A} \in \mathbb{A}^{na}$ and $z \in [0, 1]$ such that for all Markov chain $\mathcal{M}$, $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}) = z$ 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 configuration (ℓ, t) in some region of $\mathcal{A}$ 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 transition relation of the region graph and then we prove that z is either 0 or 1. The base 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 $(s'\ell', t')$ 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 11, $p(sc(s), \ell', t')$ 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 $z = 1$. Otherwise ℓ_f is not reachable implying that no run is accepting and thus $z = 0$.

◂

We now establish that $\mathbb{A}^{na} \precsim_{\mathcal{M}} \mathbb{A}$.

▸ **Proposition 13.** *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 $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}') \neq \mathbf{Pr}_{\mathcal{M}}(\mathcal{A})$.*

Proof. $\mathcal{A}$ described below is associated with 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 $\mathcal{M}$, $\mathbf{Pr}_{\mathcal{M}}(\mathcal{A}') = \mathbf{Pr}_{\mathcal{M}}(\mathcal{A})$.
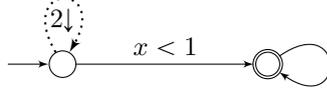
Pick an arbitrary Markov chain $\mathcal{M}$ and define $\mathcal{M}_\lambda$ as the Markov chain which has a single transition from its initial state to the initial state of $\mathcal{M}$ with rate λ. It is routine to check that $\mathbf{Pr}_{\mathcal{M}_\lambda}(\mathcal{A}) = \frac{1-e^{-\lambda}}{1-e^{-2\lambda}}$ (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 12 applied to $\mathcal{A}''$.

◀

• From $\mathbb{A}^{rc}$ to $\mathbb{A}^{na}$.

We introduced the family $\mathbb{A}^{rc}$ to provide an accurate syntactical characterization of DTA for which the autonomous transitions do not add expressive power. In some sense, the DTA of Proposition 13 emphasizes the interest of $\mathbb{A}^{rc}$ since the cycle performed by the autonomous resetting transition points out what increases the expressive power. As a first step in the other direction, we establish that in $\mathbb{A}^{rc}$ the autonomous resetting transitions can be mimicked using additional finite memory.
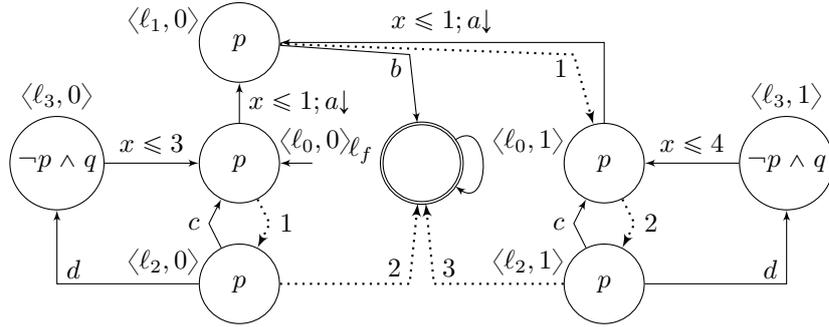
▸ **Proposition 14.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}^{rc}$ and outputs $\mathcal{A}' \in \mathbb{A}^{nar}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

We illustrate this transformation on the DTA presented on the left of Figure 1 producing the DTA of Figure 2. Here it is sufficient to duplicate once every location. The resetting autonomous transition from ℓ_1 to ℓ_0 does not reset anymore and goes from $\langle \ell_1, 0\rangle$ to $\langle \ell_0, 1\rangle$. The synchronized resetting transition from ℓ_0 to ℓ_1 yields two transitions from $\langle \ell_0, i\rangle$ to $\langle \ell_1, 0\rangle$ for $i \in \{0, 1\}$. The timed constants occurring in the transitions from some $\langle \ell, 1\rangle$ are increased by 1. The location $\langle \ell_1, 1\rangle$ is not present since unreachable from $\langle \ell_0, 0\rangle$.

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

▸ **Proposition 15.** *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 $\mathcal{A} \in \mathbb{A}^{nar}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$, $(|Aut| + 1)|Synch| \geqslant 2^n$.*

In order to eliminate cycles of autonomous transitions of a DTA in $\mathbb{A}^{nar}$, we also add to locations a finite memory representing the number of autonomous transitions visited since the

Figure 2 Elimination of resetting autonomous transitions.

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.

▸ **Proposition 16.** *There exists an algorithm operating in quadratic time that takes as input $\mathcal{A} \in \mathbb{A}^{nar}$ and outputs $\mathcal{A}' \in \mathbb{A}^{nc}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

We show how to eliminate autonomous transitions without reset (that do not form a cycle). 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 has also an impact on the incoming transitions of ℓ. Repeating (at most $|L|$ times) this transformation eliminates all autonomous transitions. In appendix, we illustrate this transformation on the DTA presented on the right of Figure 1.

▸ **Proposition 17.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}^{nc}$ and outputs $\mathcal{A}' \in \mathbb{A}^{na}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

The exponential blowup due to the repetition of duplication of locations is unavoidable as shown by the next proposition.

▸ **Proposition 18.** *There exists a family of automata $\{\mathcal{A}_n\}_{n \in \mathbb{N}}$ in $\mathbb{A}^{nar}$ 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.*

4 DTA with boolean guards

An alternative version of DTA has been proposed in [14]: locations have no associated formula, synchronized transitions have two formulas, one evaluated on the source state and one evaluated on the destination state. Autonomous transitions have a formula evaluated in the current state. Our aim is twofold: first relate the two versions w.r.t. expressiveness and

conciseness and then examine whether the relations established for the previous case still hold. Let us formalize this model by adapting the definitions of section 2.

▸ **Definition 19** (**DTA**). *A DTA* with boolean guards *is defined by a tuple* $\mathcal{A} = \langle L, L_0, \ell_F, AP,$ *Synch, Aut*⟩ *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.

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

and $\ell \xrightarrow{\varphi^-, \gamma, \sharp, r(,\varphi^-)} \ell'$ *(repeating sometimes the formula for unifying the notations) 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.** $\ell_f \xrightarrow{\top, \top, Act, \varnothing, \top} \ell_f \in Synch.$

One denotes the class of such DTA by $\mathbb{A}_g$ with subclasses by $\mathbb{A}_g^{na}$, $\mathbb{A}_g^{nc}$, $\mathbb{A}_g^{nar}$ and $\mathbb{A}_g^{rc}$.

▸ **Definition 20** (**Run of a DTA**).

A run *of a DTA* $\mathcal{A}$ *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* $i \in \mathbb{N},$

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

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

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

▸ **Definition 21.** *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, *and* ρ *visits* $\ell_f.$

Our results are summarized below. Let us emphasize the main items: (1) while being equivalent to the original version, this model is exponentially more concise and (2) the elimination of autonomous transitions without reset that was performed in exponential time in the original version is now performed in polynomial time.

$$\mathbb{A}_g^{na} \sim_{\mathcal{L}} \mathbb{A}_g^{nc} \sim_{\mathcal{L}} \mathbb{A}_g^{nar} \sim_{\mathcal{L}} \mathbb{A}_g^{rc} \precsim_{\mathcal{M}} \mathbb{A}_g \sim_{\mathcal{L}} \mathbb{A}$$
with $\mathbb{A}_g$ (resp. $\mathbb{A}_g^{rc}$) exponentially more concise than $\mathbb{A}$ (resp. $\mathbb{A}_g^{nar}$) and
a quadratic (resp. polynomial) translation from $\mathbb{A}_g^{nar}$ (resp. $\mathbb{A}_g^{nc}$) to $\mathbb{A}_g^{nc}$ (resp. $\mathbb{A}_g^{na}$).

● **From** $\mathbb{A}$ **to** $\mathbb{A}_g$ **and back.** The translation from $\mathbb{A}$ to $\mathbb{A}_g$ mainly consists in shifting the formula of the location to the incoming transitions with a particular handling of the initial locations.

▸ **Proposition 22.** *There exists an algorithm operating in quadratic time that takes as input $\mathcal{A} \in \mathbb{A}$ (resp. $\mathbb{A}^{na}$, $\mathbb{A}^{nc}$, $\mathbb{A}^{nar}$, $\mathbb{A}^{rc}$) and outputs $\mathcal{A}' \in \mathbb{A}_g$ (resp. $\mathbb{A}_g^{na}$, $\mathbb{A}_g^{nc}$, $\mathbb{A}_g^{nar}$, $\mathbb{A}_g^{rc}$) with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

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

▸ **Proposition 23.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}_g$ (resp. $\mathbb{A}_g^{na}$, $\mathbb{A}_g^{nc}$, $\mathbb{A}_g^{nar}$, $\mathbb{A}_g^{rc}$) and outputs $\mathcal{A}' \in \mathbb{A}$ (resp. $\mathbb{A}^{na}$, $\mathbb{A}^{nc}$, $\mathbb{A}^{nar}$, $\mathbb{A}^{rc}$) with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

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

▸ **Proposition 24.** *There exists a family of automata $\{\mathcal{A}_n\}_{n \in \mathbb{N}}$ in $\mathbb{A}_g^{na}$ such that the size of $\mathcal{A}_n$ belongs to $O(n \log(n))$ and for all $\mathcal{A} \in \mathbb{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$ the number of its locations is at least $2^n - 1$.*

• **From $\mathbb{A}_g^{rc}$ to $\mathbb{A}_g^{na}$.** Since the algorithm of Proposition 14 does not transform the propositional features of the DTA, it is also appropriate for $\mathbb{A}_g^{rc}$. Similarly the exponential lower bound still holds.

▸ **Proposition 25.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}_g^{rc}$ and outputs $\mathcal{A}' \in \mathbb{A}_g^{nar}$ 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 $\mathcal{A} \in \mathbb{A}_g^{nar}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$, $(|Aut| + 1)|Synch| \geqslant 2^n$.

Since the algorithm of Proposition 16 does not transform the propositional features of the DTA, it is also appropriate for $\mathbb{A}_g^{nar}$.

▸ **Proposition 26.** *There exists an algorithm operating in quadratic time that takes as input $\mathcal{A} \in \mathbb{A}_g^{nar}$ and outputs $\mathcal{A}' \in \mathbb{A}_g^{nc}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

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 representing 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. Every transition is labelled by a formula and the formulas labeling outgoing transitions from a vertex are mutually exclusive. 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 transition 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). So in the next proposition, one assumes that the formulas can be specified with such DD.

▸ **Proposition 27.** *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 (exemplified in the appendix)
• 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 $\langle \ell, rg \rangle \xrightarrow{\varphi,x=i,\sharp,\varnothing} \langle \ell', \{i\} \rangle$.

• 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 $\langle \ell, rg \rangle \xrightarrow{\varphi^- \wedge \bigwedge_{\alpha_k \leqslant rg'} \neg\varphi_k,\gamma,B,r,\varphi^+} \langle \ell', rg' \rangle$.

In $\mathcal{A}_2$ disregarding the priority of autonomous transitions does not modify the language.

• 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 $\langle \ell', \{i\} \rangle$, one specifies the formula φ 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 labeled by the formulas of the autonomous transitions.

Then for all synchronized transition $\langle \ell', \{i\} \rangle \xrightarrow{\varphi^-,\gamma,B,r,\varphi^+} \langle \ell'', rg'' \rangle$, one creates a transition $\langle \ell, rg \rangle \xrightarrow{\varphi \wedge \varphi^-,x \geqslant i \wedge \gamma,B,r,\varphi^+} \langle \ell'', rg'' \rangle$. Then, all locations $\langle \ell_f, rg \rangle$ are merged into a single one.

◀

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$^{\text{TA}}$ 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 13 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. Considering DTAs with boolean guards associated with transitions the effect of the presence of autonomous transitions does not change, but the use of guards may lead to more concise DTAs. 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$^{\text{TA}}$ model-checking, in particular for the recently proposed component-based method [2].

───── **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^{TA}: an efficient model checking tool for CSL^{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, Serge Haddad, and Jeremy Sproston. Model checking timed and stochastic properties with CSL^{TA}. *IEEE TSE*, 35(2):224–240, 2009.

12 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.

13 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.

14 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.

15 W.Douglas Obal II and William Sanders. State-space support for path-based reward variables. *Performance Evaluation*, 35:233–251, 05 1999.

6 Appendix

6.1 $\mathbb{A}$ versus $\mathbb{A}^{na}$

▸ **Proposition 28.** *Let $\mathcal{A}$ be the DTA presented in the proof of Proposition 13. There does not exist an automaton $\mathcal{A}' \in \mathbb{A}^{na}$ such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof. Assume by contradiction that there exists $\mathcal{A}' \in \mathbb{A}^{na}$ such that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.
Let C be the maximal constant occurring in $\mathcal{A}'$ that we assume w.l.o.g. to be odd.
The timed path $(C+1) \xrightarrow{a} (0) \xrightarrow{a} (0) \cdots \xrightarrow{a} (0) \cdots$ belongs to $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$ (here we omit the valuation since the set of propositions is empty).
Let $(\ell_0, C+1) \xrightarrow{\gamma_0, \{a\}, r_0} (\ell_1, 0) \cdots$ be the corresponding accepting run.
Then for all $\tau > 0$, the run $(\ell_0, C+\tau) \xrightarrow{\gamma_0, \{a\}, r_0} (\ell_1, 0) \cdots$ is also accepting which entails that the timed path $(C+2) \xrightarrow{a} (0) \xrightarrow{a} (0) \cdots \xrightarrow{a} (0) \cdots$ belongs to $\mathcal{L}(\mathcal{A}')$ in contradiction with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.

◂

▸ **Lemma 11.** *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:*

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

The above formula computes 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 $i - 1 \leqslant t \leqslant i \leqslant C$:

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

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.

$$p_{n+1}(s, \ell, t) = \int_t^i \lambda_s e^{-\lambda_s(\tau-t)} p_n(sc(s), sc_i(\ell), sc(\ell, \tau)) d\tau$$

$$+ \sum_{i < j \leqslant C} \int_{j-1}^j \lambda_s e^{-\lambda_s(\tau-t)} p_n(sc(s), sc_j(\ell), sc(\ell, \tau)) d\tau$$

$$+ e^{-\lambda_s(C-t)} p_n(sc(s), sc_{C+1}(\ell), sc(\ell, C+1))$$

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 Equation 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.

◀

6.2 $\quad \mathbb{A}^{rc} \sim_{\mathcal{L}} \mathbb{A}^{nar} \sim_{\mathcal{L}} \mathbb{A}^{nc} \sim_{\mathcal{L}} \mathbb{A}^{na}$

▶ **Proposition 14.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}^{rc}$ and outputs $\mathcal{A}' \in \mathbb{A}^{nar}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

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 set of initial locations $L'_0 = \{\langle \ell, 0 \rangle \in L' \mid \ell \in L_0\}$ and final location ℓ_f (with its single looping transition). For all $\langle \ell, i \rangle$, $\Lambda(\langle \ell, i \rangle) = \Lambda(\ell)$.

Let γ be a guard. Define $\gamma + i$ the guard where all constants of γ are increased by i.

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

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

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

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

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.

Let us establish the correctness of this transformation.

● Let $\rho = (\ell_0, v_0, \bar{x}_0, \delta_0) \xrightarrow{\gamma_0, B_0, r_0} (\ell_1, v_1, \bar{x}_1, \delta_1) \cdots \xrightarrow{\gamma_{n-1}, B_{n-1}, r_{n-1}} (\ell_n, v_n, \bar{x}_n, \delta_n) \cdots$ 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 kth states of the original and simulating runs while d_k represents how c_k has to be taken into account in the kth 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 $\rho' = (\langle\ell_0,0\rangle, v_0, \bar{x}_0, \delta_0) \xrightarrow{\gamma_0+d_0, B_0, r'_0} (\langle\ell_1, d_1\rangle, v_1\bar{x}_1 + c_1, \delta_1)\cdots \xrightarrow{\gamma_{n-1}+d_{n-1}, B_{n-1}, r'_{n-1}}$ $(\langle\ell_n, d_n\rangle, v_n, \bar{x}_n + c_n)$ 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 $\rho' = (\langle\ell_0,0\rangle, v_0, \bar{x}'_0, \delta_0) \xrightarrow{\gamma_0, B_0, r'_0} (\langle\ell_1, d_1\rangle, v_1, \bar{x}'_1, \delta_1)\cdots \xrightarrow{\gamma_{n-1}+d_{n-1}, B_{n-1}, r'_{n-1}}$ $(\langle\ell_n, d_n\rangle, v_n, \bar{x}'_n, \delta_n)\cdots$ 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 $\gamma_k + d_{k-1}, A_n, r'_k$. For all k define c_k by $c_0 = 0$ and for $k > 0$:
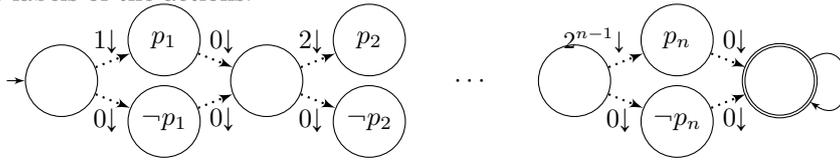
- if $r_k = \varnothing$ then $c_k = c_{k-1}$;
- if $r_k = r'_k =\downarrow$ then $c_k = 0$;
- if $B_k = \sharp$, $r_k =\downarrow$ and $\gamma_k = x = c$ then $c_k = c_{k-1} + c$.

Then $\rho = (\ell_0, v_0, \bar{x}'_0 - c_0, \delta_0) \xrightarrow{\gamma_1, B_1, r_1} (\ell_1, v_1, \bar{x}'_1 - c_1)\cdots \xrightarrow{\gamma_{n-1}, B_{n-1}, r_{n-1}} (\ell_n, v_n, \bar{x}'_n - c_n, \delta_n)\cdots$ is a run of $\mathcal{A}$.

◀

▶ **Proposition 15.** *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 $\mathcal{A} \in \mathbb{A}^{nar}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$, $(|Aut| + 1)|Synch| \geqslant 2^n$.*

Proof. Consider the automaton $\mathcal{A}_n$ described below. Here *Act* is a singleton and so we omit the labels of the actions.

Given a valuation v of the atomic propositions, let us define the integer $z(v) \in [0, 2^n[$ by: $z(v) = \sum_{i\leqslant n} 2^{i-1}\mathbf{1}_{v(p_i)=\top}$. 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 $\mathcal{A} \in \mathbb{A}^{nar}$ 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 $t_v = \ell_v \xrightarrow{\gamma_v, r_v} \ell'_v$ 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 16.** *There exists an algorithm operating in quadratic time that takes as input $\mathcal{A} \in \mathbb{A}^{nar}$ and outputs $\mathcal{A}' \in \mathbb{A}^{nc}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof. Let $\mathcal{A} \in \mathbb{A}^{nar}$. 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 set of initial locations $L'_0 = \{(\ell, 0) \in L' \mid \ell \in L_0\}$ and final location ℓ_f (with its single looping transition). For all $\langle \ell, i \rangle$, $\Lambda(\langle \ell, i \rangle) = \Lambda(\ell)$. $\Lambda(\ell_\perp) = \top$.

For all synchronized transition $\ell \xrightarrow{\gamma, B, r} \ell'$ of $\mathcal{A}$ and $i \leqslant K$:
- if $\ell' = \ell_f$ then there is a synchronized transition $(\ell, i) \xrightarrow{\gamma, B, r} \ell_f$;
- otherwise there is a synchronized transition $(\ell, i) \xrightarrow{\gamma, B, r} (\ell', 0)$.

For all autonomous transition $\ell \xrightarrow{x=c, \sharp, \varnothing} \ell'$ of $\mathcal{A}$ and $i \leqslant K$:
- if $i = K$ then there is an autonomous transition $(\ell, i) \xrightarrow{x=c, \sharp, \varnothing} \ell_\perp$;
- else if $\ell' = \ell_f$ then there is an autonomous transition $(\ell, i) \xrightarrow{x=c, \sharp, \varnothing} \ell_f$;
- otherwise there is an autonomous transition $(\ell, i) \xrightarrow{x=c, \sharp, \varnothing} (\ell', i+1)$.

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.

◂

▸ **Proposition 17.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}^{nc}$ and outputs $\mathcal{A}' \in \mathbb{A}^{na}$ with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof. The algorithm iteratively applies a transformation that decreases the number of locations that are sources of autonomous transitions. Let $\{t_u = \ell \xrightarrow{x=c_u, \sharp} \ell_u\}_{u \leqslant k}$ be the last transitions of elementary paths consisting of autonomous transitions with maximal length starting from some ℓ (there must be at least one such location). W.l.o.g assume that $c_1 \leqslant \cdots \leqslant c_k$. We denote this set of values (adding 0 if necessary) as $0 = \alpha_0 < \cdots < \alpha_K$ by eliminating repetitions. By convention, $\alpha_{-1} = -\infty$. One builds another $\mathcal{A}' \in \mathbb{A}^{nc}$ as follows.

One creates two sets of locations: $\{\langle \ell, i, u \rangle \mid 0 \leqslant i \leqslant K, 1 \leqslant u \leqslant k, \alpha_i \leqslant c_u\}$ and $\{\langle \ell, i \rangle \mid 0 \leqslant i \leqslant K+1\}$. The location $\langle \ell, i, u \rangle$ corresponds to location ℓ with additional requirements (1) to enter in it with clock valuation in $]\alpha_{i-1}, \alpha_i]$ and (2) such that if no synchronized transitions occurs then the autonomous transition t_u is followed. For $i \leqslant K$, the location $\langle \ell, i \rangle$ corresponds to location ℓ with additional requirements (1) to enter in it with clock valuation in $]\alpha_{i-1}, \alpha_i]$ and (2) such that no autonomous transition can occur. The location $\langle \ell, K+1 \rangle$ corresponds to location ℓ with additional requirement to enter in it with clock valuation in $]\alpha_K, \infty[$.
- For all i, u, $\Lambda(\langle \ell, i, u \rangle) = \Lambda(\ell) \wedge (\bigwedge_{\alpha_{i-1} < a_{u'} < a_u} \neg\Lambda(\ell_{u'})) \wedge \Lambda(\ell_u)$;
- For all $i \leqslant K$, $\Lambda(\langle \ell, i \rangle) = \Lambda(\ell) \wedge (\bigwedge_{\alpha_{i-1} < a_{u'}} \neg\Lambda(\ell_{u'}))$;
- $\Lambda(\langle \ell, K+1 \rangle) = \Lambda(\ell)$.

For all transition $\ell^- \xrightarrow{\gamma^-, B^-} \ell$ with $\ell^- \neq \ell$ and all i, u, one creates transitions: $\ell^- \xrightarrow{\gamma^- \wedge \alpha_{i-1} < x \leqslant \alpha_i, B^-}$ $\langle \ell, i, u \rangle$, when $i \leqslant K$, $\ell^- \xrightarrow{\gamma^- \wedge \alpha_{i-1} < x \leqslant \alpha_i, B^-} \langle \ell, i \rangle$ and $\ell^- \xrightarrow{\gamma^- \wedge \alpha_K < x, B^-} \langle \ell, K+1 \rangle$.

For all transition $\ell^- \xrightarrow{\gamma^-, B^-, \downarrow} \ell$ with $\ell^- \neq \ell$ and all u, one creates transitions: $\ell^- \xrightarrow{\gamma^-, B^-, \downarrow}$ $\langle \ell, 0, u \rangle$ and $\ell^- \xrightarrow{\gamma^-, B^-, \downarrow} \langle \ell, 0 \rangle$.

For all synchronized transition $\ell \xrightarrow{\gamma^+, B^+, r^+} \ell^+$ with $\ell^+ \neq \ell$ and all i, u, one creates transitions: $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x < c_u, B^-, r^+} \ell^+$ and $\langle \ell, i \rangle \xrightarrow{\gamma^+, B^-, r^+} \ell^+$.

For all synchronized transition $\ell \xrightarrow{\gamma^+, B^+} \ell$ and all i, i', u, u', one creates transitions: $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x < c_u \wedge \alpha_{i'-1} < x \leqslant \alpha_{i'}, B^-}$ $\langle \ell, i', u' \rangle$,
$\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x < c_u \wedge \alpha_{i-1} < x \leqslant \alpha_i, B^-} \langle \ell, i' \rangle$,
$\langle \ell, i \rangle \xrightarrow{\gamma^+ \wedge \alpha_{i'-1} < x \leqslant \alpha_{i'}, B^-} \langle \ell, i', u' \rangle$, $\langle \ell, i \rangle \xrightarrow{\gamma^+ \wedge \alpha_{i'-1} < x \leqslant \alpha_{i'}, B^-} \langle \ell, i' \rangle$
and $\langle \ell, i \rangle \xrightarrow{\gamma^+ \wedge \alpha_K < x, B^-} \langle \ell, K+1 \rangle$.

For all synchronized transition $\ell \xrightarrow{\gamma^+, B^+, \downarrow} \ell$ and all i, u, u', one creates transitions: $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x < c_u, B^-, \downarrow}$ $\langle \ell, 0, u' \rangle$, $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x < c_u, B^-, \downarrow} \langle \ell, 0 \rangle$,
$\langle \ell, i \rangle \xrightarrow{\gamma^+, B^-, \downarrow} \langle \ell, 0, u' \rangle$, and $\langle \ell, i \rangle \xrightarrow{\gamma^+, B^-, \downarrow} \langle \ell, 0 \rangle$.

For all synchronized transition $\ell_u \xrightarrow{\gamma^+, B^+, r^+} \ell^+$ with $\ell^+ \neq \ell$, one creates transitions: for all i, $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x \geqslant c_u, B^+, r^+} \ell^+$.

For all synchronized transition $\ell_u \xrightarrow{\gamma^+, B^+} \ell$ and all i, i', u' one creates transitions: $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x \geqslant c_u \wedge \alpha_{i'-1} < x \leqslant \alpha_{i'}, B^+}$ $\langle \ell, i', u' \rangle$, $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x \geqslant c_u \wedge \alpha_{i'-1} < x \leqslant \alpha_{i'}, B^+} \langle \ell, i' \rangle$ and $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x \geqslant c_u \wedge \alpha_K < x, B^+} \langle \ell, K+1 \rangle$.

For all synchronized transition $\ell_u \xrightarrow{\gamma^+, B^+, \downarrow} \ell$ and all i, u' one creates transitions: $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x \geqslant c_u, B^+, \downarrow}$ $\langle \ell, 0, u' \rangle$ and $\langle \ell, i, u \rangle \xrightarrow{\gamma^+ \wedge x \geqslant c_u, B^+, \downarrow} \langle \ell, 0 \rangle$.

Then we delete ℓ and all adjacent transitions. By construction, $\mathcal{A}$ and $\mathcal{A}'$ accept the same timed paths and there is one less location source of an autonomous transition. One can iterate at most $|L|$ times this transformation leading to an equivalent automaton in $\mathbb{A}^{na}$.

◀

We illustrate this transformation on the DTA presented on the right of Figure 1 producing the DTA of Figure 3. The only location from which autonomous transitions start is ℓ_0. The autonomous transitions are triggered when $x = \alpha$ and lead to locations with formulas q and $p \wedge \neg q$. Combining them with formula p of ℓ_0 entails that ℓ_0 has to be duplicated into three locations: (1) $\ell_{0,0}$ that must be entered with clock value at most α and with formula $p \wedge \neg q$, (2) $\ell_{0,1}$ that must be entered with clock value at most α and with formula $p \wedge q$, (3) and another location that must be entered with clock value greater than α and with formula p. However this location is not reachable and so omitted in the figure. The transition looping around ℓ_0 yields to four transitions depending on which formulas hold before and after the transition. The transition from $\ell_{0,0}$ to ℓ_1 corresponds to the sequence of transitions in the original automaton of the autonomous transition from ℓ_0 to ℓ_1 followed by the synchonized transition which loops around ℓ_1.

▶ **Proposition 18.** *There exists a family of automata $\{\mathcal{A}_n\}_{n \in \mathbb{N}}$ in $\mathbb{A}^{nar}$ 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.*

Proof. Consider the automaton $\mathcal{A}_n$ described below. Here Act is a singleton and so we omit the labels of the actions.
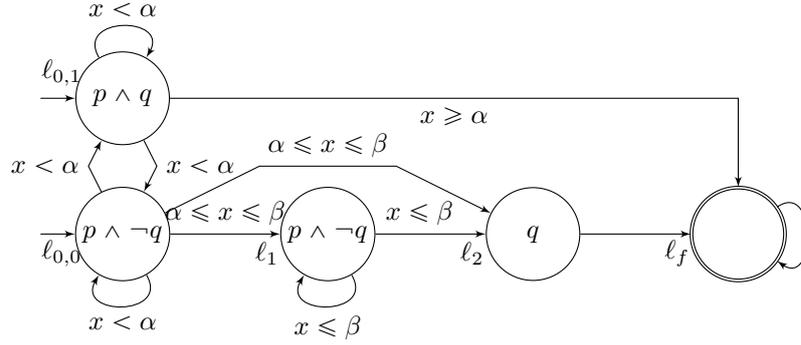
Figure 3 Elimination of autonomous transitions with no reset.
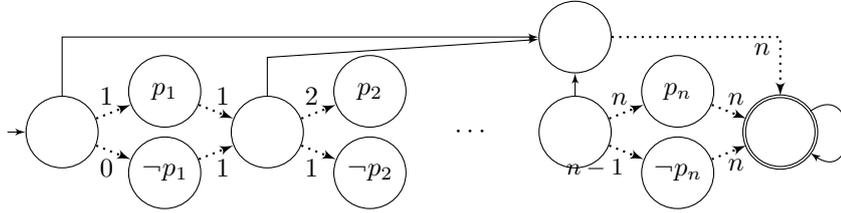
$\mathcal{A}_n$ accepts the timed paths where: either there is no action before time n or there is a single action before time n at some time τ with $i \leqslant \tau < i + 1 \leqslant n$ and p_i initially true.

Assume by contradiction that there exists $\mathcal{A} \in \mathbb{A}^{na}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$ and less than 2^n locations. In particular, it has less than 2^n initial locations. Since there are accepted timed paths for any initial valuation of the atomic propositions, two different valuations (say v and v') of the atomic propositions should satisfy $\Lambda(\ell_0)$ for some initial location ℓ_0. Let p_i be such that $v(p_i) = \top$ and $v'(p_i) = \bot$. Any timed path starting with initial valuation v and a single action occurring in $[0, n[$ at some time τ with $i \leqslant \tau < i + 1 \leqslant n$ is accepted. This implies that any timed path starting with initial valuation v' and a single action occurring in $[0, n[$ at some time τ with $i \leqslant \tau < i + 1 \leqslant n$ is also accepted. But such a timed path does not belong to $\mathcal{L}(\mathcal{A}_n)$.

◀

6.3 $\mathbb{A}$ versus $\mathbb{A}_g$

▸ **Proposition 22.** *There exists an algorithm operating in quadratic time that takes as input $\mathcal{A} \in \mathbb{A}$ (resp. $\mathbb{A}^{na}$, $\mathbb{A}^{nc}$, $\mathbb{A}^{nar}$, $\mathbb{A}^{rc}$) and outputs $\mathcal{A}' \in \mathbb{A}_g$ (resp. $\mathbb{A}_g^{na}$, $\mathbb{A}_g^{nc}$, $\mathbb{A}_g^{nar}$, $\mathbb{A}_g^{rc}$) with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

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

For all synchronized transition $\ell \xrightarrow{\gamma, B, r} \ell'$ in $\mathcal{A}$, $\mathcal{A}'$ includes the synchronized transition $\ell \xrightarrow{\top, \gamma, B, r, \Lambda(\ell')} \ell'$ and if $\ell \in L_0$ then $\mathcal{A}'$ 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 $\mathcal{A}$, $\mathcal{A}'$ includes the autonomous transition $\ell \xrightarrow{\Lambda(\ell'), x=\alpha, \sharp, r} \ell'$ and if $\ell \in L_0$ then $\mathcal{A}'$ includes the autonomous transition: $\ell_0 \xrightarrow{\Lambda(\ell) \wedge \Lambda(\ell'), x=\alpha, \sharp, r} \ell'$.

The quadratic factor is mainly due to the substitution of $|L|$ formulas by at least $|E|$ formulas.

◀

▶ **Proposition 23.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}_g$ (resp. $\mathbb{A}_g^{na}$, $\mathbb{A}_g^{nc}$, $\mathbb{A}_g^{nar}$, $\mathbb{A}_g^{rc}$) and outputs $\mathcal{A}' \in \mathbb{A}$ (resp. $\mathbb{A}^{na}$, $\mathbb{A}^{nc}$, $\mathbb{A}^{nar}$, $\mathbb{A}^{rc}$) with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.*

Proof. Given $\ell \in L$, let $\varphi_1^\ell, \dots \varphi_{n_\ell}^\ell$ be the formulas of entering guards of transitions incoming ℓ and exiting guards of transitions outgoing ℓ. Then $L' = \{\langle \ell, I \rangle \mid \ell \in L \wedge I \subseteq \{1, \dots, n_\ell\}\} \cup \{\ell_f^*\}$ where ℓ_f^* is the final state (fulfilling the requirements of a DTA in $\mathbb{A}$) and for all $\langle \ell, I \rangle$, $\Lambda(\langle \ell, I \rangle) = \bigwedge_{i \in I} \varphi_i^\ell \wedge \bigwedge_{i \notin I} \neg \varphi_i^\ell$.

For all synchronized transition $\ell \xrightarrow{\varphi_i^\ell, \gamma, B, r, \varphi_{i'}^{\ell'}} \ell'$ in $\mathcal{A}'$ and all I, I' such that $i \in I$ and $i' \in I'$, $\mathcal{A}'$ includes the synchronized transition: $\langle \ell, I \rangle \xrightarrow{\gamma, B, r} \langle \ell', I' \rangle$.

For all autonomous transition $\ell \xrightarrow{\varphi_i^\ell, x = \alpha, \sharp, r} \ell'$ in $\mathcal{A}$ and all I, I' such that $i \in I$ and $i' \in I'$ with $\varphi_i^\ell = \varphi_{i'}^{\ell'}$, $\mathcal{A}'$ 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 24.** *There exists a family of automata $\{\mathcal{A}_n\}_{n \in \mathbb{N}}$ in $\mathbb{A}_g^{na}$ such that the size of $\mathcal{A}_n$ belongs to $O(n \log(n))$ and for all $\mathcal{A} \in \mathbb{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$ the number of its locations is at least $2^n - 1$.*

**Proof.** Consider the automaton $\mathcal{A}_n$ described below.
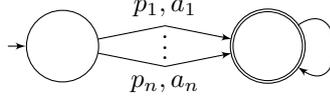
This automaton accepts timed paths whose first action may be a_i only if the initial state fullfills p_i. Consider in $\mathcal{A}$ the locations reached at time 0 by the runs before the first action is performed. At least $2^n - 1$ initial valuations must reach such a location. Assume that $\mathcal{A}$ has less than $2^n - 1$ 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.

◀

6.4 $\mathbb{A}_g^{rc} \sim_\mathcal{L} \mathbb{A}_g^{nar} \sim_\mathcal{L} \mathbb{A}_g^{nc} \sim_\mathcal{L} \mathbb{A}_g^{na}$

▶ **Proposition 25.** *There exists an algorithm operating in exponential time that takes as input $\mathcal{A} \in \mathbb{A}_g^{rc}$ and outputs $\mathcal{A}' \in \mathbb{A}_g^{nar}$ 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 $\mathcal{A} \in \mathbb{A}_g^{nar}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_n)$, $(|Aut| + 1)|Synch| \geqslant 2^n$.

Proof. Here we only exhibit the family $\{\mathcal{A}_n\}_{n \in \mathbb{N}}$. Given $\mathcal{A}_n$ described below, the proof is done as the one of Proposition 15.


◀
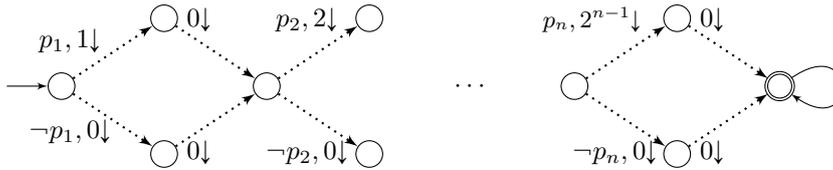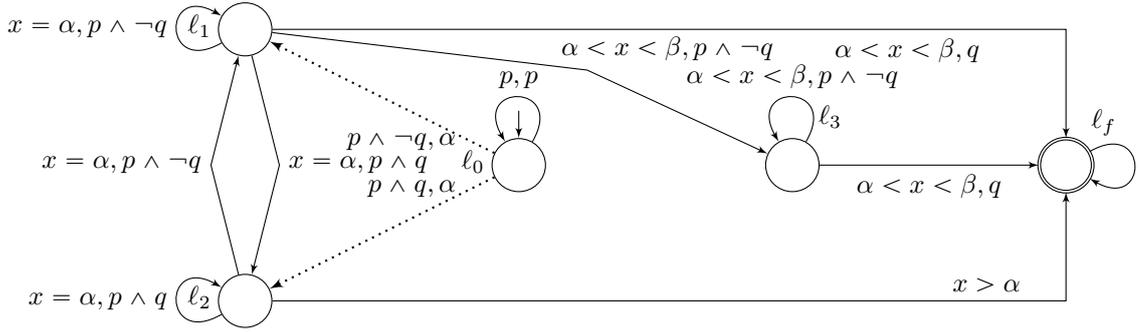
Figure 4 A DTA specification of $p\mathbf{U}^{]\alpha,\beta[}q$ with $\alpha > 0$.

Application of Proposition 27.

The DTA with boolean guards of Figure 4 illustrates how to specify the temporal formula $p\mathbf{U}^{]\alpha,\beta[}q$ with a DTA $\mathcal{A} \in \mathbb{A}_g^{nc}$. 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) either $p \wedge q$ was satisfied and the formula is satisfied, (2) either $p \wedge \neg q$ was satisfied, q is now satisfied and the action occurs before β and so the formula is satisfied (3) or $p \wedge \neg q$ was satisfied and 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 5 depicts the DTA $\mathcal{A}' \in \mathbb{A}_g^{na}$ obtained by applying the transformation of Proposition 27 to the DTA $\mathcal{A} \in \mathbb{A}_g^{nc}$ depicted in Figure 4. 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.
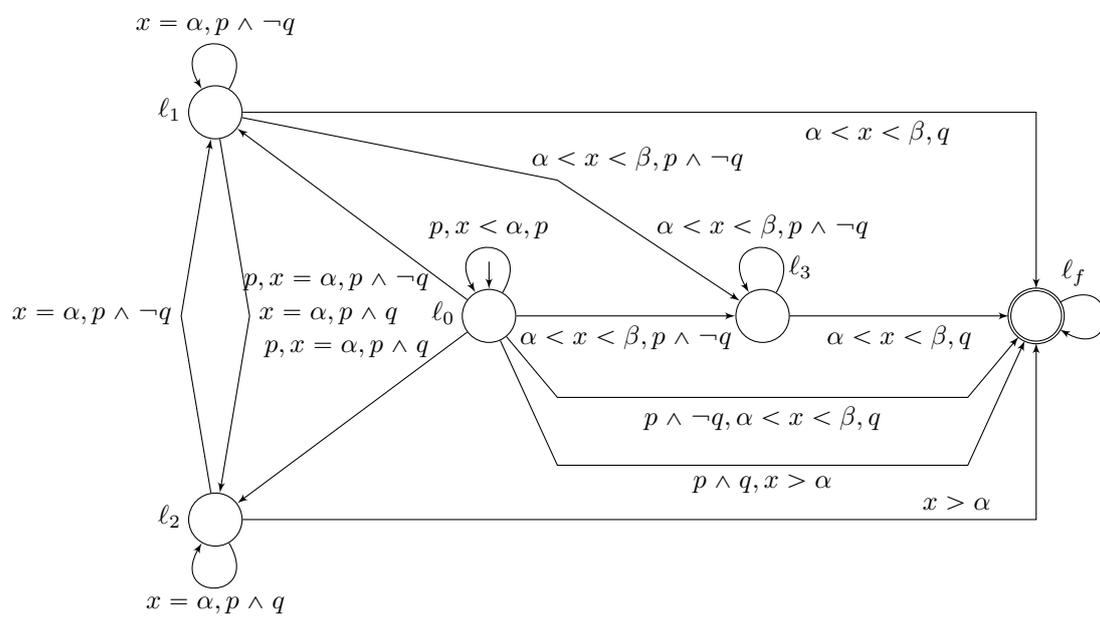
Figure 5 Another DTA specification of $p\mathbf{U}^{]\alpha,\beta[}q$ with $\alpha > 0$.