Susanna Donatelli
Serge Haddad
Jeremy Sproston

Model Checking Timed and Stochastic Properties with CSL-TA

Research Report LSV-08-30

October 2008

Laboratoire
Spécification
et
Vérification

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Model Checking Timed and Stochastic Properties with CSL^{TA}

Susanna Donatelli[1], Serge Haddad[2], Jeremy Sproston[1]
[1] Dipartimento di Informatica, Università di Torino, Italy
[2] LSV - ENS Cachan & CNRS, France

Abstract

Markov chains are a well-known stochastic process that provide a balance between being able to adequately model the system's behavior and being able to afford the cost of the model solution. Systems can be modelled directly as Markov chains, or with a higher-level formalism for which Markov chains represent the underlying semantics. Markov chains are widely used to study the performance of computer and telecommunication systems. The definition of stochastic temporal logics like Continuous Stochastic Logic (CSL) and its variant asCSL, and of their model-checking algorithms, allows a unified approach to the verification of systems, allowing the mix of performance evaluation and probabilistic verification.

In this paper we present the stochastic logic CSL^{TA}, which is more expressive than CSL and asCSL, and in which properties can be specified using automata (more precisely, timed automata with a single clock). The extension with respect to expressiveness allows the specification of properties referring to the probability of a finite sequence of timed events. A typical example is the responsiveness property "with probability at least 0.75, a message sent at time 0 by a system A will be received before time 5 by system B and the acknowledgment will be back at A before time 7", a property that cannot be expressed in either CSL or asCSL. Furthermore, the choice of using automata rather than the classical temporal operators Next and Until should help in enlarging the accessibility of model checking to a larger public. We also present a model-checking algorithm for CSL^{TA}.

1 Introduction

Software performance engineering [1] (SPE) is a discipline that advocates an integrated approach to system design and system analysis. SPE emphasizes the importance of obtaining performance measures early in the development process, when appropriate decisions can be taken at a lower cost, usually through modelling, given that the target system is not available yet. Over the years, SPE has evolved to encompass other non-functional requirements such as dependability and quality of service (QoS) aspects.

Part of the work of a software performance engineer consists of defining appropriate performance and dependability properties. In this paper we introduce a new stochastic logic that allows us to define and check non-functional properties over Continuous Time Markov Chains (CTMC).

CTMCs are a well-known stochastic process that balances the need to have a model that is representative enough of the real system being modelled, while still allowing affordable solution costs: there are standard and widespread solution methods for the computation of performance measures of a CTMC. The relevance of CTMCs for SPE is clearly evidenced by recent works on automatic translations from UML diagrams, annotated according to some standard or quasi-standard profile such as UML-SPT [2] and MARTE [3], to various performance evaluation formalisms like queueing networks [4], stochastic Petri nets [5] and stochastic process algebras [6], whose underlying stochastic process is in most cases a CTMC. A complete overview of the tools and algorithms for the translation from UML diagrams to performance models can be found in [7].

Historically, CTMCs have been analyzed using steady state and transient analysis, to compute the probability of finding the system in a given state assuming it has reached equilibrium, or the probability of finding a system in a given state at time t. From these basic methods, the use of state and/or transition rewards allows the computation of performance/dependability properties.

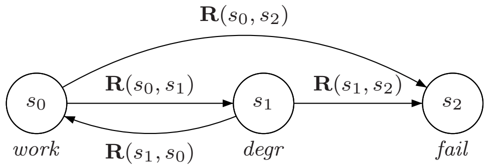

Figure 1. A simple CTMC

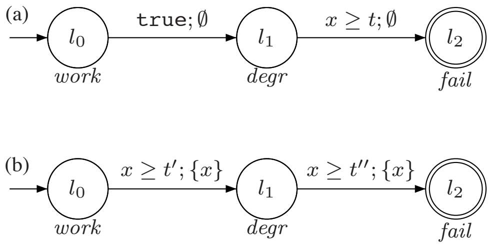

Figure 2. Automata for two QoS properties

More recently the definition of Continuous Stochastic Logic (CSL) [8, 9] and variants, such as asCSL [10], has introduced a new approach for the definition of the performance and dependability properties of a system. Temporal logic based approaches are particularly useful when the measure of interest depends on the execution path. Given a formal description of the system and its requirements, we can then execute a model-checking algorithm which establishes automatically whether the system model meets the requirements expressed in CSL or asCSL.

To illustrate the advantages of stochastic logics, consider a system whose stochastic behavior is described by a CTMC, whose states (of which there can be millions) are partitioned into “system is working properly” (*work*-states), “system is working in degraded mode” (*degr*-states), or “system is not working properly” (*fail*-states). The CTMC can move from *work* to *degr* states and to *fail* states (either directly or through *degr* states). A simple example of CTMC exhibiting this behaviour is shown in Figure 1. A classical dependability property requires the computation of the probability of failing within the time interval I, or later than a given threshold t: these probabilities can be easily computed using classical solution methods for CTMCs.

Instead, if we are interested in only those failures in which the system fails within the time interval I, without first entering the degraded mode, we have to compute the probability of reaching a *fail* state within I, while passing only through *work* states. The stochastic temporal logic CSL has temporal operators that allow a simple and semantically-clear description of such a property using the Until operator: $\mathcal{P}_{\leq\lambda}(\mathit{work}\ \mathcal{U}^I \mathit{fail})$. The property is satisfied in a state s if the set of timed paths of the CTMC that start in s and visit only work states before entering a fail state at a time in the interval I have a probability at most λ.

The logic asCSL permits the specification of paths in terms of state labels (such as work, degr and fail) and *action labels*. For example, $\mathcal{P}_{\leq\lambda}((\mathit{work}, \mathit{Act})^*; (\mathit{work}, \mathit{failure1}); (\mathit{fail}, \surd)^I)$, is similar to the CSL formula above, with the additional restriction that the change from work-states to fail-states is due to action $\mathit{failure1}$.

In this paper, we propose the new stochastic temporal logic CSL$^{\text{TA}}$ (Continuous Stochastic Logic with Timed Automata), which builds on CSL and asCSL by enriching the set of properties that can be defined and verified, and presents its associated model-checking algorithm. Let us first explain the main motivations for introducing a new logic. Modifications are along two lines: properties are specified using deterministic one-clock timed automata, and the defined logic is at least as expressive as CSL and asCSL (and strictly more expressive than CSL and asCSL without nested formulae). For the time being we shall be informal, at the risk of being slightly imprecise, to convey the main ideas. The rest of the paper will provide the required formal development.

The idea of using automata for specifying system behavior is familiar to computer scientists in general, and to software engineers in particular. The use of automata to specify temporal logic properties is not new: Vardi and Wolper [11] define a linear-time temporal logic with Büchi automata operators, while Clarke et al. [12] introduce the temporal logic ECTL, which uses Muller automata to specify linear and branching temporal properties, and develop an associated model-checking algorithm.

Timed automata [13] are a widespread formalism for the specification of timed systems, and are supported by tools such as UPPAAL [14]. Previous work has considered the use of timed automata to specify both the system and the properties that it should satisfy [13], or, more typically, the use of timed temporal logic to specify the properties of a timed automata system [15, 16]; observe that in [16], the model-checking algorithm involves the transformation of temporal logic properties into timed automata. In this paper we use timed automata to specify timed and probabilistic properties of the system, and not to specify the system itself.

We illustrate the limitations of CSL and asCSL with respect to CSL$^{\text{TA}}$ using again the CTMC of Figure 1. Assume that we are interested in the probability of the system exhibiting the following behavior: the system goes from *work* states to *degr* states and then from *degr* states to *fail* states in a time greater than or equal to t. This property can be expressed in CSL$^{\text{TA}}$ using the timed automaton of Figure 2(a), where x is a clock (a variable whose value increases at the same rate as time). This property can be expressed also in asCSL, using a formula similar to the one given above: $\mathcal{P}_{\leq\lambda}((work, Act)^*; (degr, Act)^*; (fail, \surd)^I)$, for the interval $I = [t, \infty)$. This property cannot be expressed in CSL, which can only express the probability of being in *work* or *degr* states until, at a time at least t, the system moves to *fail* states: this property is obviously not equivalent to the original one since it includes also paths that cycle between *work* and *degr* states.

Now assume that the QoS requirements imposed on the system are more stringent and detailed, requiring that, with probability at least λ, the system goes from *work* states to *degr* states in no less than t' time units, and then from *degr* states to *fail* states in no less than t'' time units. Paths are therefore characterized in terms of states and in terms of two time constraints. This property can be expressed in CSL$^{\text{TA}}$ using the timed automaton of Figure 2(b), where the edge label $\{x\}$ indicates that the clock x is reset to zero on traversal of the edge. Observe that the two properties are different, because (a) checks only the global time to get to *fail* states, while (b) also "looks" inside the composition of this duration: indeed the automaton of Figure 2(b) is not equivalent to the automaton of Figure 2(a) with $t = t' + t''$.

This QoS requirement cannot be expressed either in CSL or in asCSL, as will be explained in later sections. What can be indeed expressed in CSL and asCSL is that, with probability at most λ, the system moves not before t' from *work* states to the subset of *degr* states from which, with probability at most λ', the system moves to *fail* not before t''. Note that we have to utilize a "fake" probability, introducing an overspecification with respect to the original QoS requirement, because timed intervals cannot be nested directly in CSL and asCSL, while probabilistic operators can be nested. Note that taking $\lambda' = 1$ does not solve the problem.

In addition to introducing CSL$^{\text{TA}}$, we present its associated model-checking algorithm. Contrary to the previous approaches, which perform *ad hoc* transformations of the CTMC before a transient or steady-state analysis, this algorithm generates a Markov regenerative process and then computes a reachability probability on this process. Furthermore, we prove that CSL$^{\text{TA}}$ is at least as expressive as CSL and asCSL: it is possible to transform any CSL or asCSL formula into an equivalent CSL$^{\text{TA}}$ formula. We note that the CSL$^{\text{TA}}$ model-checking algorithm, when executed on CSL$^{\text{TA}}$ properties transformed from CSL properties, is no more expensive in terms of computational complexity than the CSL model-checking algorithm of [9]. Finally, we show that CSL$^{\text{TA}}$ is strictly more expressive than CSL: note that the proof technique used is different from those used in the non-stochastic context, for example in [17]. We also show that CSL$^{\text{TA}}$ is more expressive than asCSL, when restricting to the case of formulae without nesting.

With regard to related work, performance metrics that depend on paths have also been studied in [18, 19]. In particular, the work in [19] uses automata for the specification of the set of paths of interest of a CTMC: rewards, which are usually associated with states or transitions of the CTMC, are instead associated with locations and transitions of the automaton, thus providing a wide range of performance measures based on states and/or events of the CTMC. We also note that the logic CSL$^{\text{TA}}$ is similar to the logic $\mathrm{TECTL}^*_{\exists}$ [20] from the non-probabilistic model-checking literature. One-clock timed automata have been studied in, for example, [21, 22]. Finally, we recall that the original definition of CSL permitted the description of a sequence of timed Until formulae within a single probabilistic operator $\mathcal{P}_{\sim\lambda}$ [8], in contrast to the more established definition in which only one time Until formula can be included within a probabilistic operator: however, the decidability results of [8] are based on results from algebraic and trascendental number theory, whereas established performance evaluation techniques are used as the foundation of the algorithms for CSL in [9] and for CSL$^{\text{TA}}$ in this paper.

The rest of the paper is organized as follows: Section 2 defines the syntax and semantics of CSL$^{\text{TA}}$, illustrated with the help of small examples. Section 3 presents the model-checking algorithm for CSL$^{\text{TA}}$ and gives an example on a simple CTMC, while Section 4 compares the expressiveness of CSL$^{\text{TA}}$, CSL and asCSL. Section 5 summarizes the paper and discusses future work. A conference version of this paper appears as [23]. We extend that version in the following ways: the semantics of CSL$^{\text{TA}}$ is improved in order to make the modelling of properties more intuitive; Section 3 is expanded; and a more formal approach, including proofs of the main results, is taken in Section 4.

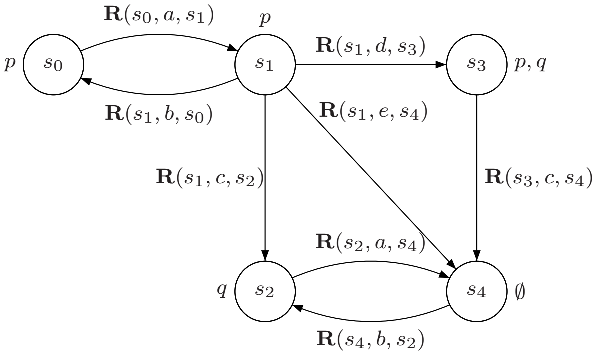
$\mathbf{R}(s_0, a, s_1)$
p
p
s_0
s_1
$\mathbf{R}(s_1, d, s_3)$
s_3
p, q
$\mathbf{R}(s_1, b, s_0)$
$\mathbf{R}(s_1, e, s_4)$
$\mathbf{R}(s_1, c, s_2)$
$\mathbf{R}(s_3, c, s_4)$
$\mathbf{R}(s_2, a, s_4)$
q
s_2
s_4
$\emptyset$
$\mathbf{R}(s_4, b, s_2)$

Figure 3. An ASMC

2 Syntax and Semantics of CSL$^{\text{TA}}$

In this section, we first introduce a number of preliminary concepts. After defining a class of labelled CTMCs, we recall the notion of execution path of a CTMC as a finite or infinite sequence of transitions from state to state. We then introduce a restricted class of timed automata, and consider the manner in which such automata can be used to express properties of CTMC execution paths. Finally we introduce the syntax of CSL$^{\text{TA}}$, which is similar to CSL but which uses timed automata to express properties of paths, and present its semantics.

2.1 Labelled Markov Chains

We first introduce continuous-time Markov chains labelled both by atomic propositions on states and by actions on transitions.

Atomic propositions can refer to basic properties which are observed when the system is in a state (such as $idle$ or $error$), whereas actions refer to basic properties which are observed when the system makes a transition from state to state (such as $activate$ or $send\_message$). Such labelled Markov chains can be used as the underlying semantic model of high-level formalisms such as stochastic Petri nets and stochastic process algebras. Let $\mathbb{R}_{\geq 0}$ ($\mathbb{R}_{>0}$) be the set of non-negative (positive) reals, and let $\mathbb{N}$ be the set of natural numbers.

Definition 2.1 (Action- and state-labelled Markov chain) *An* action- and state-labelled continuous-time Markov chain *(ASMC) is a tuple $\mathcal{M} = \langle S, Act, AP, lab, \mathbf{R} \rangle$, where S is a finite set of states, Act is a finite set of action labels, AP is a finite set of atomic propositions, $lab : S \rightarrow 2^{AP}$ is a state labelling function, and $\mathbf{R} : S \times Act \times S \rightarrow \mathbb{R}_{\geq 0}$ is a rate matrix. We require that for any state s there exists a pair $(a, s') \in Act \times S$ with $\mathbf{R}(s, a, s') > 0$.*

Intuitively, the rate matrix $\mathbf{R}$ describes the transitions that can be made between states of the ASMC, on which actions, and with which rate. A transition from state s to state s', performing action a, exists if $\mathbf{R}(s, a, s') > 0$. A transition from s to s' performing a and of duration $\tau \in \mathbb{R}_{>0}$ is denoted by $s \xrightarrow{a,\tau} s'$.

Definition 2.2 (Paths of $\mathcal{M}$) *A* finite path *of an ASMC $\mathcal{M}$ is a finite sequence of transitions $\sigma = s_0 \xrightarrow{a_0,\tau_0} s_1 \xrightarrow{a_1,\tau_1} \ldots s_{n-1} \xrightarrow{a_{n-1},\tau_{n-1}} s_n$ where $\mathbf{R}(s_i, a_i, s_{i+1}) > 0$ for $i = 0, \ldots, n-1$. An* infinite path *of $\mathcal{M}$ is an infinite sequence of transitions $\sigma = s_0 \xrightarrow{a_0,\tau_0} s_1 \xrightarrow{a_1,\tau_1} \ldots$ where $\mathbf{R}(s_i, a_i, s_{i+1}) > 0$ for all $i \geq 0$ and such that $\sum_{i \geq 0} \tau_i = \infty$.*

Notation. Given $s \in S$, let $Path^{\mathcal{M}}(s)$ be the set of infinite paths $s_0 \xrightarrow{a_0,\tau_0} s_1 \xrightarrow{a_1,\tau_1} \ldots$ such that $s_0 = s$. Let $\Pr_s^{\mathcal{M}}$ be the probability measure on $Path^{\mathcal{M}}(s)$ defined in the standard manner (for example, see [9, 10]). Let $\sigma = s_0 \xrightarrow{a_0,\tau_0} s_1 \xrightarrow{a_1,\tau_1} \ldots \xrightarrow{a_{n-1},\tau_{n-1}} s_n$ be a finite path. Then $|\sigma| = n$ denotes the length of σ, and $\tau(\sigma) = \sum_{i=0}^{n-1} \tau_i$ is the total duration of σ. By convention, $\tau_{-1} = 0$. For an infinite path σ, we let $|\sigma| = \infty$ and $\tau(\sigma) = \infty$.

As usual, we can describe an ASMC by a graph. An example of an ASMC is given in Figure 3. The vertices of this graph are its states whereas the edges represent its transitions. The atomic propositions (here p and q) that are satisfied in a state are indicated near the corresponding node. Finally, the rate of a transition labels the corresponding edge.

2.2 Timed Automata

We now present a restricted variant of timed automata [13], which are used in CSL$^{\mathrm{TA}}$ to describe properties of ASMC paths. More precisely, in our context, timed automata are used as acceptors of finite ASMC paths. The class of timed automata that we consider are deterministic (i.e., given a path σ of an ASMC, there is at most one path of the timed automaton which reads σ), and have a single clock. In the same manner as in classical analysis techniques for timed automata [13], we present our timed automata using natural-numbered constants (rational-numbered constants can also be considered through re-scaling) We proceed to define deterministic (one-clock) timed automata. We use the symbol $\sharp$ to denote a pseudo-action that is not included in the action set Act of any ASMC ($\sharp \notin Act$). Clock variables are real-valued variables whose value increases linearly with time. We consider a single *clock* variable x. A *valuation* $\bar{x} \in \mathbb{R}_{\geq 0}$ is interpreted as assigning a non-negative real value to x. A *constraint* is of the form $\alpha \prec x \prec \beta$ or $\alpha \prec x$ where $\alpha, \beta \in \mathbb{N}, \alpha \leq \beta$ and $\prec$ stands for either $<$ or $\leq$. An *inner constraint* is a constraint $\alpha \prec x \prec \beta$ such that $\alpha < \beta$. The set of inner constraints is denoted Inner. A *boundary constraint* is a constraint $\alpha \leq x \leq \beta$ such that $\alpha = \beta$; we generally write boundary constraints as $x = \alpha$. The set of boundary constraints is denoted $\mathsf{Boundary}$. Let γ be a constraint and $\bar{x}$ be a clock valuation. Then we write $\bar{x} \vDash \gamma$ if γ is satisfied when $\bar{x}$ is substituted for x in γ.

Locations of a timed automaton are labelled with *state propositions*. A state proposition is a proposition which either holds, or does not hold, in an ASMC state. For a set Σ of state propositions, let $\models_\Sigma$ be its associated satisfaction relation: hence we write $\mathcal{M}, s \models_\Sigma \Phi$ to denote that the state s of the ASMC $\mathcal{M}$ satisfies Φ. We omit $\mathcal{M}$ and write $s \models_\Sigma \Phi$ when clear from the context. We also consider Boolean expressions of state propositions: for example $s \models_\Sigma \Phi_1 \wedge \Phi_2$ denotes that s satisfies Φ_1 and Φ_2. Let $\mathcal{B}(\Sigma)$ be the set of Boolean expressions over state propositions of Σ. We will make precise later in the paper the set of state propositions Σ used in CSL$^{\mathrm{TA}}$. For the purposes of the current explanation, the reader can consider the case in which $\Sigma = AP$ with $s \models_{AP} p$ if and only if $p \in lab(s)$, for a state s and $p \in AP$.

Definition 2.3 (Deterministic Timed Automaton) *A* deterministic timed automaton *(DTA)* $\mathcal{A} = \langle \Sigma, Act, L, \Lambda, Init, Final, \rightarrow \rangle$ *comprises:*

- Σ*, a finite alphabet of state propositions;*
- Act*, a finite alphabet of actions;*
- L*, a finite set of locations;*
- $\Lambda : L \rightarrow \mathcal{B}(\Sigma)$*, a location labelling function;*
- $Init$*, a subset of* L *called the initial locations;*
- $Final$*, a subset of* L *called the final locations;*
- $\rightarrow \subseteq L \times ((\mathsf{Inner} \times 2^{Act}) \cup (\mathsf{Boundary} \times \{\sharp\})) \times \{\emptyset, x\} \times L$*, a set of edges, where* $l \xrightarrow{\gamma, A, r} l'$ *denotes that* $(l, \gamma, A, r, l') \in \rightarrow$.

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

- **Initial determinism:** $\forall l, l' \in Init, \Lambda(l) \wedge \Lambda(l') \Leftrightarrow \mathtt{false}$.
- **Determinism on actions:** $\forall A, A' \subseteq Act\, s.t.\, A \cap A' \neq \emptyset, \forall l, l', l'' \in L$*, if* $l'' \xrightarrow{\gamma, A, r} l \wedge l'' \xrightarrow{\gamma', A', r'} l'$ *then either* $\Lambda(l) \wedge \Lambda(l') \Leftrightarrow \mathtt{false}$ *or* $\gamma \wedge \gamma' \Leftrightarrow \mathtt{false}$.
- **Determinism on $\sharp$:** $\forall l, l', l'' \in L$*, if* $l'' \xrightarrow{\gamma, \sharp, r} l \wedge l'' \xrightarrow{\gamma', \sharp, r'} l'$ *then either* $\Lambda(l) \wedge \Lambda(l') \Leftrightarrow \mathtt{false}$ *or* $\gamma \wedge \gamma' \Leftrightarrow \mathtt{false}$.
- **No $\sharp$-labelled loops:** *For all sequences* $l_0 \xrightarrow{\gamma_0, A_0, r_0} l_1 \xrightarrow{\gamma_1, A_1, r_1} \cdots \xrightarrow{\gamma_{n-1}, A_{n-1}, r_{n-1}} l_n$ *such that* $l_0 = l_n$*, there exists* $i \leq n$ *such that* $A_i \neq \sharp$.

Let Φ_1 and Φ_2 be state propositions in the alphabet Σ. Figure 4 depicts a DTA, using the usual conventions for the graphical representation of timed automata (i.e., nodes represent locations, and edges represent edges labelled with their guards, actions sets, and the set of clocks to be reset to 0, respectively). Initial locations are denoted by an incoming arrow with no source, and final locations are denoted by a double border. Edges labelled by $\sharp$ are called *boundary edges* while the

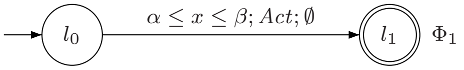

Figure 4. The DTA $\mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}\Phi_1}$

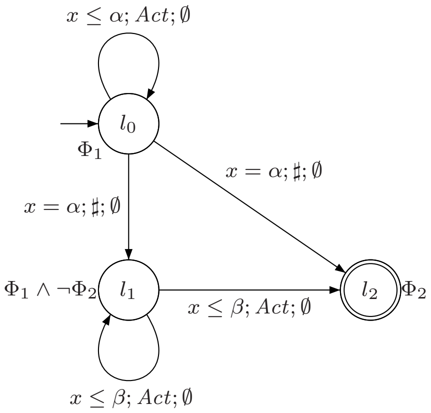
$x \leq \alpha; Act; \emptyset$
l_0
Φ_1
$x = \alpha; \sharp; \emptyset$
$x = \alpha; \sharp; \emptyset$
$\Phi_1 \wedge \neg\Phi_2$
l_1
l_2
Φ_2
$x \leq \beta; Act; \emptyset$
$x \leq \beta; Act; \emptyset$

Figure 5. The DTA $\mathcal{A}_{\Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2}$

other edges are called *inner edges*. For the DTA of Figure 4 the determinism is obvious, because there is no choice allowed. Figure 5 shows a more complex DTA, with inner edges (the self loops on l_0, l_1, and the arc from l_0 to l_1), and boundary edges (the arcs from l_0 to l_1 and from l_0 to l_2). The DTA respects the determinism constraints of the definition, because the two boundary edges out of location l_0 lead to two locations whose labelling cannot be both satisfied by any state of an ASMC (indeed $\Lambda(l_1) \wedge \Lambda(l_2) = (\Phi_1 \wedge \neg\Phi_2) \wedge \Phi_2 = \texttt{false}$).

The semantics of DTA, expressed in terms of paths, is standard [13], apart from the case of boundary edges, which are *urgent* and have *priority* over other edges. Urgency specifies that time cannot elapse if a boundary edge is enabled, and it is a feature of the variants of timed automata used in the tools UPPAAL [14] and KRONOS [24]. In our context, the notions of urgency and priority are not relevant when considering a DTA in isolation. They will be introduced later when we define the notion of a path of a DTA that reads an ASMC path, and the notion of path acceptance (Definition 2.7).

Examples of DTA: Next and Until. The DTA $\mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}\Phi_1}$ in Figure 4 specifies behaviors in which the first transition of $\mathcal{M}$ must be taken to a state satisfying Φ_1 after at least α time units, but not after β time units, and corresponds to the Next path formula $\mathcal{X}^{[\alpha,\beta]}\Phi_1$ of CSL [9]. The action of the transition is not important; this fact is represented by the action set Act on the edge of the DTA.

We can use the DTA $\mathcal{A}_{\Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2}$ of Figure 5 to represent the property of eventually reaching a state satisfying Φ_2 at some instant between α and β time units, remaining within states satisfying Φ_1 before that point (the timed Until path property $\Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2$ of CSL [9]). In contrast to the previous example, this DTA uses boundary edges which witness that the time interval $[\alpha, \beta]$ has been entered. In this way, we distinguish between the time interval $[0, \alpha)$, where the truth value of Φ_2 is irrelevant, and the time interval $[\alpha, \beta]$, where the truth value of Φ_2 becomes relevant.

Paths of a DTA. We now define a notion of path in a DTA, which represents a timed evolution of the automaton.

Definition 2.4 (Configurations of $\mathcal{A}$) *A* configuration *of a DTA $\mathcal{A}$ is a pair $(l, \bar{x})$, where $l \in L$ and $\bar{x}$ is a valuation.*

Given an edge $e = (l, \gamma, A, r, l') \in\rightarrow$, let $\mathsf{source}(e) = l$, $\mathsf{guard}(e) = \gamma$, $\mathsf{action}(e) = A$, $\mathsf{reset}(e) = r$, and $\mathsf{target}(e) = l'$. We let the valuation $\bar{x}[x := 0]$ be equal to 0 and let the valuation $\bar{x}[\emptyset := 0]$ be equal to $\bar{x}$.

Definition 2.5 (Step of $\mathcal{A}$) *A* step *of a DTA $\mathcal{A}$ from a configuration $(l, \bar{x})$ is $(l, \bar{x}) \xrightarrow{\delta, e} (l', \bar{x}')$ of $\mathcal{A}$ with $\delta \geq 0$, $\mathsf{source}(e) = l$, $\bar{x} + \delta \vDash \mathsf{guard}(e)$, $\mathsf{target}(e) = l'$, and $\bar{x}' = (\bar{x} + \delta)[\mathsf{reset}(e) := 0]$.*

A single step in the evolution of $\mathcal{A}$ is a transition in which we let time elapse and then an inner or a boundary edge is taken. Note that $\delta = 0$ is also allowed.

Definition 2.6 (Paths of $\mathcal{A}$) *A* finite path *of a DTA $\mathcal{A}$ is a finite sequence of steps* $(l_0, \bar{x}_0) \xrightarrow{\delta_0, e_0} (l_1, \bar{x}_1) \xrightarrow{\delta_1, e_1} \dots (l_{n-1}, \bar{x}_{n-1}) \xrightarrow{\delta_{n-1}, e_{n-1}} (l_n, \bar{x}_n)$.

An infinite path *of a DTA $\mathcal{A}$ is an infinite sequence of steps* $(l_0, \bar{x}_0) \xrightarrow{\delta_0, e_0} (l_1, \bar{x}_1) \xrightarrow{\delta_1, e_1} \dots$.

2.3 Acceptance of ASMC paths

We now give an intuitive explanation of how a path $\sigma = s_0 \xrightarrow{a_0, \tau_0} s_1 \xrightarrow{a_1, \tau_1} \dots$ of an ASMC $\mathcal{M}$ *can be accepted* by a DTA $\mathcal{A}$. The key idea is that $\mathcal{A}$ evolves according to the states and actions that it "reads" along σ. Recalling that the value of clocks in timed automata increase at the same rate as real-time, as time elapses in $\mathcal{M}$ the value of the clock x of $\mathcal{A}$ changes accordingly. Steps corresponding to inner edges of $\mathcal{A}$ *are triggered by transitions of* $\mathcal{M}$, whereas steps corresponding to boundary edges of $\mathcal{A}$ *are triggered by the elapse of time* (without a corresponding transition of $\mathcal{M}$).

The DTA $\mathcal{A}$ begins in a configuration $(l_0, 0)$ with location $l_0 \in \mathit{Init}$ such that the initial state s_0 of σ satisfies the expression $\Lambda(l_0)$ over state propositions (formally, $s_0 \models_\Sigma \Lambda(l_0)$). Note that, by initial determinism, there is at most one $l \in \mathit{Init}$ such that s_0 satisfies $\Lambda(l)$. If s_0 does not satisfy $\Lambda(l)$ for all $l \in \mathit{Init}$, then $\mathcal{A}$ rejects σ.

Given the existence of $l_0 \in \mathit{Init}$ such that s_0 satisfies $\Lambda(l_0)$, the DTA $\mathcal{A}$ then moves from $(l_0, 0)$ to another configuration depending on the first transition $s_0 \xrightarrow{a_0, \tau_0} s_1$ of σ. First we consider the case in which there are no outgoing boundary edges from l_0. If there exists a step $(l_0, 0) \xrightarrow{\tau_0, e_0} (l_1, \bar{x}_1)$ such that $a_0 \in \mathsf{action}(e_0)$ and s_1 satisfies $\Lambda(l_1)$, then this step is taken. Note that, by determinism on actions, there exists at most one step satisfying these conditions. If no such step exists, then $\mathcal{A}$ rejects σ.

Now we consider the case in which there exists at least one boundary edge from l_0. Consider the step $(l_0, 0) \xrightarrow{\delta'_0, e'_0} (l'_1, \bar{x}'_1)$, where $\mathsf{action}(e'_0) = \sharp$, which (by urgency of boundary edges) corresponds to the earliest boundary edge available by letting time elapse from $(l_0, 0)$. If $\delta'_0 > \tau_0$, then this step is available only *after* $\mathcal{M}$ has performed the transition $s_0 \xrightarrow{a_0, \tau_0} s_1$; hence, the DTA "reads" the ASMC transition *before* the boundary edge is available, and this case is similar to the case in which there are no boundary edges from l_0 in the previous paragraph. If, however, $\delta'_0 \leq \tau_0$, the DTA takes the step before "reading" the ASMC transition. Note that, in this case, the remaining time before the transition of $\mathcal{M}$ must be "read" by $\mathcal{A}$ is $\tau_0 - \delta'_0$, rather than τ. This has implications for deciding whether a boundary edge can be taken from $(l'_1, \bar{x}'_1)$, or whether the transition of $\mathcal{M}$ must be "read" before any boundary edge is enabled for choice.

Unless $\mathcal{A}$ has already rejected σ, the path of $\mathcal{A}$ generated by σ then continues from $(l_1, \bar{x}_1)$ or $(l'_1, \bar{x}'_1)$. Finally, if the path of $\mathcal{A}$ generated by σ reaches a configuration with a location in Final, then σ is accepted. If, however, the path of $\mathcal{A}$ generated by σ does not reach such a configuration, then σ is rejected. Hence, there are two ways in which $\mathcal{A}$ can reject σ: if there does not exist a step corresponding to the "reading" of a transition of σ, or if a final location is never reached.

We now describe formally the conditions for the acceptance of an ASMC path by a DTA.

Definition 2.7 *Let $\mathcal{M}$ be an ASMC, and let $\mathcal{A}$ be a DTA. The infinite path $\sigma_\mathcal{M} = s_0 \xrightarrow{a_0, \tau_0} s_1 \xrightarrow{a_1, \tau_1} \dots$ of $\mathcal{M}$ is* accepted *by $\mathcal{A}$ if there exists:*

- *a finite path $\sigma_\mathcal{A} = (l_0, \bar{x}_0) \xrightarrow{\delta_0, e_0} (l_1, \bar{x}_1) \xrightarrow{\delta_1, e_1} \dots \xrightarrow{\delta_{m-1}, e_{m-1}} (l_m, \bar{x}_m)$ of $\mathcal{A}$,*
- *an index $n \in \mathbb{N}$,*
- *a time $\tau \leq \tau_n$, and*
- *a function $\kappa : \{0, \dots, m\} \to \{0, \dots, n\}$ which maps indices of $\sigma_\mathcal{A}$ to indices of $\sigma_\mathcal{M}$,*

such that the following conditions are satisfied:

C1: $l_0 \in \mathit{Init}$, $\bar{x}_0 = 0$, $\kappa(0) = 0$ *and* $\forall 0 \leq i \leq m, l_i \in \mathit{Final} \Leftrightarrow i = m$*;*

C2: $\forall 0 \leq i \leq m$, $s_{\kappa(i)} \models_\Sigma \Lambda(l_i)$*;*

C3: $\forall 0 \leq i < m$, *if e_i is an inner edge then* $\kappa(i+1) = \kappa(i) + 1 \wedge a_{\kappa(i)} \in \mathsf{action}(e_i)$ *else* $\kappa(i+1) = \kappa(i)$*;*

C4: $\forall 0 \leq i < m$, *if e_i is an inner edge then*
Boundary edges are urgent: for all $0 \leq \delta' < \delta_i$, there does not exist an edge $e' = (l_i, \gamma, \sharp, r, l') \in \to$ such that $\bar{x} + \delta' \models \gamma$ and $s_{\kappa(i)} \models_\Sigma \Lambda(l')$, and

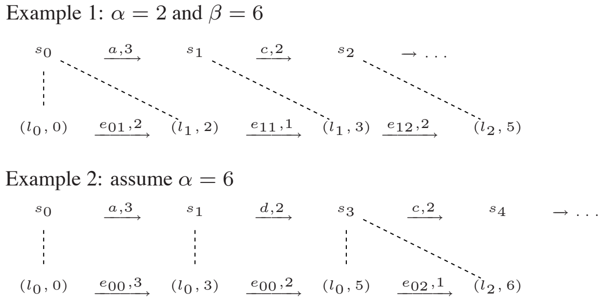

Figure 6. Examples of path acceptance

Boundary edges have priority: for all edges $e' \in \to$ such that $e \neq e'$, if $\mathsf{source}(e') = l$, $\bar{x} + \delta_i \models \mathsf{guard}(e')$, $\mathsf{target}(e') = l'$ *and* $s_{\kappa(i)} \models_\Sigma \Lambda(l')$ *then* $\mathsf{action}(e') \neq \sharp$;

C5: $\forall 0 \leq i < n$, $\sum_{j|\kappa(j)=i} \delta_j = \tau_{\kappa(i)}$;

C6: $\sum_{j|\kappa(j)=n} \delta_j = \tau$.

Condition *C1* specifies that $\sigma_\mathcal{A}$ must start from an initial location and end in a final location. *C2* requires that the state propositions satisfy the expressions labelling the corresponding locations in the sequence. *C3* specifies that κ can map consecutive indices of $\sigma_\mathcal{A}$ to the same index of $\sigma_\mathcal{M}$, provided that the DTA edges corresponding to these indices are boundary edges. It also requires that a transition of the ASMC in σ can be matched by a traversal of an inner edge provided that the action of the transition is included in the action set of the edge. *C4* limits the path of the DTA to paths whose steps respect the additional conditions on $\sharp$: urgency and priority of boundary edges. *C5* "align times", by requiring that the sum of durations in $\sigma_\mathcal{A}$ corresponding to a particular index i of $\sigma_\mathcal{M}$ is τ_i. *C6* applies the reasoning of *C5* to the case in which the path $\sigma_\mathcal{A}$ features boundary edges directly before reaching a final state (recall that $\tau \leq \tau_n$).

It should be clear that given an ASMC $\mathcal{M}$ and a DTA $\mathcal{A}$, due to our requirements for DTA and to the additional requirements of urgency and priority of boundary edges, there is at most one path of $\mathcal{A}$ that accepts a given path of $\mathcal{M}$. Accordingly, if s is a state of $\mathcal{M}$ we let $AccPath^\mathcal{M}(s, \mathcal{A})$ be the set of infinite paths of $\mathcal{M}$ starting from s that are accepted by $\mathcal{A}$.

Examples of path acceptance. In Figure 6, we present two examples of the way in which a path of the ASMC $\mathcal{M}$ of Figure 3 can be accepted by the DTA $\mathcal{A}_{p\mathcal{U}^{[\alpha,\beta]}q}$. We write e_{ij} to refer to the edge of $\mathcal{A}_{p\mathcal{U}^{[\alpha,\beta]}q}$ from location l_i to location l_j, and we use dotted lines to represent the κ function. Note that the presence of more than one dotted line from a state s_i means that the DTA traverses a boundary edge. Example 1 of Figure 6 has $\alpha = 2$ and $\beta = 6$, and depicts a case in which q does not hold at time α, but becomes true at time 5, which belongs to $[\alpha, \beta]$; therefore the DTA reaches l_2 through l_1. Example 2 of Figure 6 has $\alpha = 6$ and $\beta > 6$, and depicts a case in which q already holds before α; therefore the DTA reaches l_2 directly from l_0.

We now describe briefly some examples of paths of $\mathcal{M}$ which are rejected by $\mathcal{A}_{p\mathcal{U}^{[2,6]}q}$. If the first transition of $\mathcal{M}$ is $s_0 \xrightarrow{a,7} s_1$, then the associated path of $\mathcal{A}_{p\mathcal{U}^{[2,6]}q}$ will consist of the single step $(l_0, 0) \xrightarrow{e_{01},2} (l_1, 2)$: after the value of the clock x exceeds 6, it will not be possible to take further steps. If on the other hand the path of $\mathcal{M}$ is $s_0 \xrightarrow{a,1} s_1 \xrightarrow{c,0.5} s_2$, then the associated path of $\mathcal{A}_{p\mathcal{U}^{[2,6]}q}$ will consist of the single step $(l_0, 0) \xrightarrow{e_{00},1} (l_0, 1)$, after which it will not be possible to take any further steps: the state s_2 is not labelled by p, boundary edges are available only at time 2, and yet the transition $s_1 \xrightarrow{c,0.5} s_2$ occurs before time 2.

All the timed automata configurations considered in Figure 6 are configurations in which the DTA spends a non-zero amount of time: this is not always the case if boundary edges are involved. Indeed a path obtained from the first example by splitting the step $(l_0, 0) \xrightarrow{e_{01},2} (l_1, 2)$ into the two steps $(l_0, 0) \xrightarrow{e_{00},2} (l_0, 2)$ and $(l_0, 2) \xrightarrow{e_{01},0} (l_1, 2)$ is also an accepting path for the ASMC execution of the example. Another source of zero-time configurations in an accepting path is the presence in the path of more than one edge with the same clock constraints and no reset of clock in between.

2.4 CSL$^{\text{TA}}$

Given the definition of DTA, we can now present formally the syntax of CSL$^{\text{TA}}$. Note that the syntax of CSL$^{\text{TA}}$ is essentially identical to that of CSL or asCSL [8, 9, 10], apart from the fact that properties of paths are specified using DTA (instead of being specified by timed temporal logic operators as, for example, in CSL).

Definition 2.8 (Syntax of CSL$^{\text{TA}}$) *Let $\lambda \in [0, 1]$ be a real number, and let $\sim \in \{<, \leq, \geq, >\}$ be a comparison operator. The syntax of CSL^{TA} is defined by:*

$$\Phi ::= p \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathcal{S}_{\sim\lambda}(\Phi) \mid \mathcal{P}_{\sim\lambda}(\mathcal{A}(\Phi_1, \ldots, \Phi_n))$$

where $p \in AP$ and $\mathcal{A}(\Phi_1, \ldots, \Phi_n)$ is a DTA with a finite alphabet Σ of state propositions such that $\Sigma = \{\Phi_1, \ldots, \Phi_n\}$ and $\Phi_1, \ldots, \Phi_n$ are CSL^{TA} formulae.

Note that CSL$^{\text{TA}}$ is a CTL*-like language with nesting of path and state formulae [25]; in particular, the state propositions of a DTA are state formulae of CSL$^{\text{TA}}$. For example, we can write a CSL$^{\text{TA}}$ formula such as $\mathcal{P}_{\geq 0.99}(\mathcal{A}_{p\mathcal{U}^{[\alpha,\beta]}\mathcal{P}_{\geq 0.1}(\mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}q})})$ (which corresponds to the CSL formula $\mathcal{P}_{\geq 0.99}(p\mathcal{U}^{[\alpha,\beta]}\mathcal{P}_{\geq 0.1}(\mathcal{X}^{[\alpha,\beta]}q)))$.

Intuitively, state s satisfies the formula $\mathcal{S}_{\sim\lambda}(\Phi)$ if and only if $val \sim \lambda$, where val is the steady state probability, computed assuming the ASMC starts in s, of being in an ASMC state that satisfies Φ. State s satisfies instead the formula $\mathcal{P}_{\sim\lambda}(\mathcal{A})$ if and only if $val \sim \lambda$, where val is the probability of all ASMC paths starting in s and accepted by $\mathcal{A}$.

We proceed to define the semantics of CSL$^{\text{TA}}$ in terms of the satisfaction relation $\models$. For a given CSL$^{\text{TA}}$ formula Φ and state s of $\mathcal{M}$, we write $\mathcal{M}, s \models \Phi$ to denote that Φ is satisfied in state s. We write $\pi(s, \cdot)$ for the steady-state distribution of $\mathcal{M}$, computed starting from state s.

Definition 2.9 (Semantics of CSL$^{\text{TA}}$) *For $\mathcal{M} = \langle S, Act, AP, lab, \mathbf{R}\rangle$, and state $s \in S$, the satisfaction relation $\models$ for CSL^{TA} is defined as follows:*

$$\begin{array}{lcl}
\mathcal{M}, s \models p & \Leftrightarrow & p \in lab(s) \\
\mathcal{M}, s \models \neg\Phi & \Leftrightarrow & \mathcal{M}, s \not\models \Phi \\
\mathcal{M}, s \models \Phi_1 \wedge \Phi_2 & \Leftrightarrow & \mathcal{M}, s \models \Phi_1 \textit{ and } \mathcal{M}, s \models \Phi_2 \\
\mathcal{M}, s \models \mathcal{S}_{\sim\lambda}(\Phi) & \Leftrightarrow & \sum_{s' \in S \textit{ s.t. } \mathcal{M}, s' \models \Phi} \pi(s, s') \sim \lambda \\
\mathcal{M}, s \models \mathcal{P}_{\sim\lambda}(\mathcal{A}(\Phi_1, \ldots, \Phi_n)) & \Leftrightarrow & \mathrm{Pr}_s^{\mathcal{M}}(AccPath^{\mathcal{M}}(s, \mathcal{A}(\Phi_1, \ldots, \Phi_n))) \sim \lambda \, .
\end{array}$$

3 Model checking for CSL$^{\text{TA}}$

As usual with CTL*-like languages [25], in order to evaluate the satisfaction of a formula Φ over an ASMC $\mathcal{M}$, we proceed by a bottom-up evaluation of the subformulae occurring in Φ over all the states of $\mathcal{M}$, labelling accordingly the states with the subformulae that they satisfy. Let Φ' be such a subformula.

- If Φ' is either an atomic proposition p, $\neg\Psi$ or $\Psi \wedge \Psi'$, then the evaluation is performed by a straightforward application of Definition 2.9.
- If $\Phi' = \mathcal{S}_{\sim\lambda}(\Psi)$, then the steady-state of $\mathcal{M}$ with respect to every state s is computed first. Afterwards the steady-state probability of the subset of states that fulfill Ψ is computed and the value compared with λ in order to check whether s satisfies Φ'.
- Finally, if $\Phi' = \mathcal{P}_{\sim\lambda}(\mathcal{A})$, for each state s, we compute the probability of $AccPath^{\mathcal{M}}(s, \mathcal{A})$ (the set of paths of $\mathcal{M}$ accepted by $\mathcal{A}$), and we compare it to λ in order to check whether s satisfies Φ'. Observe that the state properties of $\mathcal{A}$ label the states of the ASMC due to the bottom-up evaluation. The computation of $\mathrm{Pr}_s^{\mathcal{M}}(AccPath^{\mathcal{M}}(s, \mathcal{A}))$ is the topic of the remainder of this section.

We use s_0 to denote the state for which we compute $\mathrm{Pr}_{s_0}^{\mathcal{M}}(AccPath^{\mathcal{M}}(s_0, \mathcal{A}))$, and let $l_0 \in Init$ be the location of $\mathcal{A}$ for which $s_0 \models_\Sigma \Lambda(l_0)$ (if no such location exists then $\mathrm{Pr}_{s_0}^{\mathcal{M}}(AccPath^{\mathcal{M}}(s_0, \mathcal{A})) = 0$).

3.1 The "synchronized" stochastic process $\mathcal{M} \times \mathcal{A}$

The computation of $\Pr^{\mathcal{M}}_{s_0}(AccPath^{\mathcal{M}}(s_0, \mathcal{A}))$ requires the definition of a stochastic process $\mathcal{M} \times \mathcal{A}$ that describes the joint evolution of $\mathcal{M}$ and $\mathcal{A}$. The stochastic process has been enriched with two absorbing states: $\top$ and $\bot$. At some instant of its execution, this process may be in one of three situations.

1. At some previous instant, $\mathcal{A}$ has not been able to mimic the execution of $\mathcal{M}$ and thus this process is in the absorbing state $\bot$ whatever are the subsequent transitions of $\mathcal{M}$.
2. At some previous instant, $\mathcal{A}$ has reached a final location by following the execution of $\mathcal{M}$ and thus this process is in the absorbing state $\top$ whatever are the subsequent transitions of $\mathcal{M}$.
3. Otherwise the process is in some state of $\mathcal{M}$ associated with a finite timed execution of $\mathcal{A}$ not ending in a final location.

States of $\mathcal{M} \times \mathcal{A}$. If at some instant the execution of $\mathcal{M}$ is neither rejected nor accepted, we observe that, for the future behavior of the process $\mathcal{M} \times \mathcal{A}$, only the current location of the path in the DTA and the value of clock x are relevant. This yields the following state description: $N(t) = (s(t), l(t), \bar{x}(t))$, where $s(t)$ is the state of $\mathcal{M}$ at time $t \in \mathbb{R}_{\geq 0}$, $l(t)$ is the location of $\mathcal{A}$ at time t, and $\bar{x}(t)$ is the value of the clock at time t. However, in $\mathcal{M} \times \mathcal{A}$ we consider only *tangible* states, i.e., states which do not trigger a boundary edge in zero time. Therefore we introduce the following definition (which allows us to skip non tangible states) which is sound because, by definition of DTA, there are no loops of $\sharp$ transitions in $\mathcal{A}$.

Definition 3.1 *Let $(s, l, \bar{x}) \in S \times L \times \mathbb{R}_{\geq 0}$. Then closure$(s, l, \bar{x})$ is defined as follows:*

- *if $l \in Final$ then closure$(s, l, \bar{x}) = \top$;*
- *if $l \notin Final$ and there is a boundary edge $l \xrightarrow{\gamma, \sharp, r} l'$ with $\bar{x} \models \gamma$ and $s \models_\Sigma \Lambda(l')$ then*

$$closure(s, l, \bar{x}) = closure(s, l', \bar{x}[r := 0]) \ ;$$

- *otherwise closure$(s, l, \bar{x}) = (s, l, \bar{x})$.*

The set of states of the process $\mathcal{M} \times \mathcal{A}$ is a subset of $\{\bot, \top\} \cup \{(s, l, \bar{x}) \mid closure(s, l, \bar{x}) = (s, l, \bar{x}),\ s \models_\Sigma \Lambda(l)\}$.

Behavior of $\mathcal{M} \times \mathcal{A}$. Let $C = \{c_0, ..., c_m\}$ be the set of constants used in the clock constraints of $\mathcal{A}$ enlarged with 0, ordered as follows: $0 = c_0 < c_1 < \cdots < c_m$. We define $\mathsf{next}(c_i) = c_{i+1}$ for all $i < m$ and $\mathsf{next}(c_m) = \infty$.

Let $(s, l, \bar{x})$ be a state such that $\bar{x} \in [c_i, \mathsf{next}(c_i))$ for some $i \leq m$. Process $\mathcal{M} \times \mathcal{A}$ can evolve from $(s, l, \bar{x})$ due to two reasons: (1) the ASMC $\mathcal{M}$ changes its state or (2) time elapses and the value of x reaches $\mathsf{next}(c_i)$.

In case (1), a transition $s \xrightarrow{a, \tau} s'$ is taking place in the ASMC $\mathcal{M}$ before the next timing constant $\mathsf{next}(c_i)$ is reached, i.e., $\bar{x} + \tau < \mathsf{next}(c_i)$, for some τ. If this transition cannot be read by $\mathcal{A}$ from its state $(l, \bar{x})$, then the stochastic process makes a transition to $\bot$. If it can be read through an edge (l, γ, A, r, l') of $\mathcal{A}$ (and by definition there is exactly either one or no such edge), then the process $\mathcal{M} \times \mathcal{A}$ moves to $closure(s', l', (v + \tau)[r := 0])$. Note that $closure$ is needed since boundary transitions may be then triggered in zero time and/or $\mathcal{A}$ may reach a final state, so that the process enters the $\top$ state.

Case (2) represents instead the situation in which the next clock barrier is reached by x before an ASMC transition takes place (an amount of time equal to $\mathsf{next}(c_i) - \bar{x}$ has elapsed). Then the process $\mathcal{M} \times \mathcal{A}$ evolves from $(s, l, \bar{x})$ to $closure(s, l, \mathsf{next}(c_i))$.

It is straightforward to show that each path of $\mathcal{M} \times \mathcal{A}$ leading to $\top$ corresponds to a (single) path in $\mathcal{M}$ accepted by $\mathcal{A}$, and vice versa. Furthermore, $\Pr^{\mathcal{M}}_{s_0}(AccPath^{\mathcal{M}}(s_0, \mathcal{A}))$ can be computed as the probability of reaching $\top$ in process $\mathcal{M} \times \mathcal{A}$ from $(s_0, l_0, 0)$. In the remainder of this section we explain how the latter probability can be computed.

$\mathcal{M} \times \mathcal{A}$ is a Markov Renewal Process. We can rewrite a state of $\mathcal{M} \times \mathcal{A}$ (different from $\bot, \top$) in terms of the last clock constant reached, as follows: $N(t) = (s(t), l(t), c(t), \bar{x}(t) - c(t))$ where $c(t)$ is the largest $c \in C$ such that $c \leq \bar{x}(t)$.

We now show that $\mathcal{M} \times \mathcal{A}$ is a Markov renewal process (MRP). For the definition of MRP and Markov renewal sequences, see, for example, [26]. Consider a sequence $\{T_k, k = 0, 1, 2, \ldots\}$ of strictly increasing timing instants in the evolution of $\mathcal{M} \times \mathcal{A}$, with $N(T_k) = (s_k, l_k, c(T_k), \bar{x}_k - c(T_k))$. The timing instants are defined as follows:

1. $T_0 = 0$,
2. if $\bar{x}_k < c_m$ then T_{k+1} is the next time at which the next constant in C is reached, the clock x is reset to 0, or the process reaches $\{\top, \bot\}$,
3. if $\bar{x}_k \geq c_m$ then T_{k+1} is the first time after T_k that the clock x is reset to 0 or the process reaches $\{\top, \bot\}$.

When for some T_k, $N(T_k) \in \{\bot, \top\}$, the sequence T_k is finite. Similarly when $\bar{x}_k \geq c_m$ it could happen that the probability to reach a regeneration point is strictly less than 1. These particular cases do not raise any problem with respect to our computation (see the discussion at the end of this section).

Let $Y_k = N(T_k^+)$ be the state directly after all of the events at time T_k.

Theorem 3.2 $(Y, T) = \{(Y_k, T_k), k = 0, 1, 2, \ldots\}$ *is a Markov renewal sequence and* $N(t)$ *is an MRP.*

The proof of Theorem 3.2 is straightforward given the definition of MRP (see [26]), because, due to the definition of T_k, we have that $Y_k = (s(T_k^+), l(T_k^+), c(T_k))$ where $c(T_k) \in C$ is the value of the clock at T_k, and the joint distribution of Y_{k+1} and $T_{k+1} - T_k$ only depends on Y_k. Therefore (Y, T) is a Markov renewal sequence. Moreover $N(t) = (s(t), l(t), c(t), \bar{x}(t) - c(t))$, which is equal to $(s_k, l_k, c(T_k), \delta)$ for some k and $0 \leq \delta \leq T_{k+1} - T_k$, is a MRP because $N(t)$, which is equal to $N(T_k + \delta)$, only depends on Y_k.

It is well-known that $Y = \{Y_k, k = 0, 1, 2, \ldots\}$ is a Discrete Time Markov Chain (DTMC), namely the embedded DTMC of the MRP. In general the solution of an MRP requires the definition of the global and local kernel matrices (see [26]). The computation of the probability of reaching the absorbing state $\top$ from the initial state can be performed on the DTMC $P_{i,j}$ which expresses the probability that, if i is the state at regeneration instant 0, then j is the state at the next regeneration instant T_1 (that is, $P_{ij} = Pr\{Y_1 = j | Y_0 = i\}$).

3.2 Tangible Reachability Graph of $\mathcal{M} \times \mathcal{A}$

We next define a data structure that supports the definition of the DTMC Y and the computation of its transition probabilities. This data structure is called Tangible Reachability Graph (TRG), and is inspired by the identically-named graph of Deterministic Stochastic Petri Nets [27], in which the elapsing of time between two consecutive timing constants c and $\mathsf{next}(c)$ is interpreted as a deterministic "transition" of duration $\mathsf{next}(c) - c$. Note that in our case a deterministic "transition" can only be preempted by a transition of $\mathcal{M} \times \mathcal{A}$ that includes a clock reset.

The nodes of the TRG take the form of elements of $(S \times L \times C) \cup \{\bot, \top\}$. For a constant $c \in C$ and a constraint γ, we write $(c, \mathsf{next}(c)) \vDash \gamma$ if, for all $\bar{x} \in (c, \mathsf{next}(c))$, we have $\bar{x} \vDash \gamma$. The arcs between nodes of the TRG are defined by the following four rules:

[M]: a simple Markovian move, in which the ASMC $\mathcal{M}$ moves "according to" the DTA $\mathcal{A}$ and there is no clock reset. Formally, there exists the arc $(s, l, c) \xrightarrow{\mathbf{M}(a,e)} (s', l', c)$ if (1) $\mathbf{R}(s, a, s') > 0$, (2) $e = (l, \gamma, A, \emptyset, l')$ is an inner edge of $\mathcal{A}$ such that $(c, \mathsf{next}(c)) \vDash \gamma$, $a \in A$ and $s' \models_\Sigma \Lambda(l')$, and (3) $l' \notin \mathit{Final}$. Furthermore, there exists the arc $(s, l, c) \xrightarrow{\mathbf{M}(a,e)} \top$ if the conditions (1) and (2) above are satisfied, and $l' \in \mathit{Final}$.

[M_res]: as for a simple Markovian move, but with a clock reset that can start an evolution of $\mathcal{A}$ over boundary transitions. Formally, there exists the arc $(s, l, c) \xrightarrow{\mathbf{M\_res}(a,e)} \mathit{closure}(s', l', 0)$ if (1) $\mathbf{R}(s, a, s') > 0$ and (2) $e = (l, \gamma, A, x, l')$ is an inner edge of $\mathcal{A}$ such that $(c, \mathsf{next}(c)) \vDash \gamma$, $a \in A$ and $s' \models_\Sigma \Lambda(l')$.

[M_KO]: a Markovian move that is not accepted by $\mathcal{A}$. Formally, there exists the arc $(s, l, c) \xrightarrow{\mathbf{M\_KO}(a)} \bot$ if there exists $s' \in S$ such that $\mathbf{R}(s, a, s') > 0$ and there does not exist an inner edge $e = (l, \gamma, A, r, l')$ of $\mathcal{A}$ such that $(c, \mathsf{next}(c)) \vDash \gamma$, $a \in A$ and $s' \models_\Sigma \Lambda(l')$.

[D]: let time elapse. Formally, there exists the arc $(s, l, c) \xrightarrow{\mathbf{D}} \mathit{closure}(s, l, \mathsf{next}(c))$ if $c < c_m$.

Note that there is a single arc from a node (s, l, c) due to a transition (s, a, s') in the ASMC, because of the assumption of determinism of $\mathcal{A}$, and that there is at most one **D** arc from a node. Observe also that we evaluate the guard of a transition with respect to the open interval $(c, \mathsf{next}(c))$ based on the straightforward result that, given a finite set of clock values (here C), the probability that an ASMC performs a transition when the value of the clock belongs to this set is null.

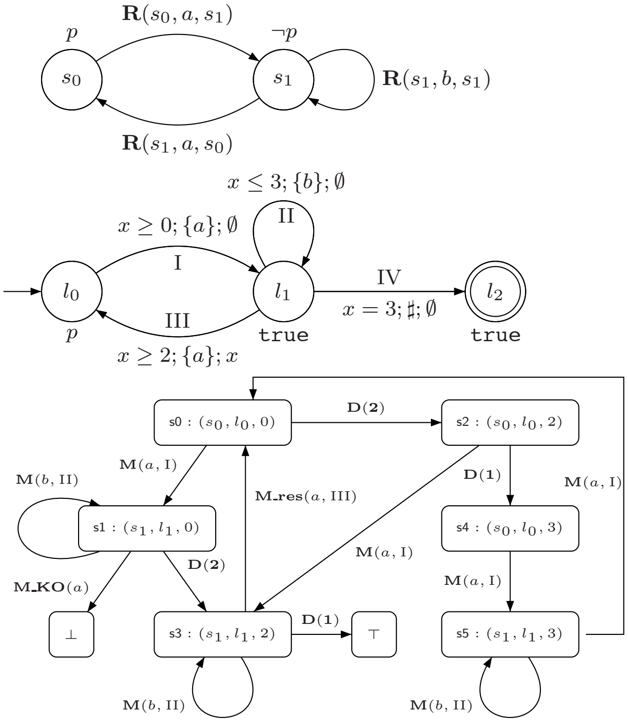
$\mathbf{R}(s_0, a, s_1)$
p
$\neg p$
s_0
s_1
$\mathbf{R}(s_1, b, s_1)$
$\mathbf{R}(s_1, a, s_0)$
$x \leq 3; \{b\}; \emptyset$
II
$x \geq 0; \{a\}; \emptyset$
I
l_0
l_1
IV
l_2
III
$x = 3; \sharp; \emptyset$
p
true
true
$x \geq 2; \{a\}; x$
s0 : $(s_0, l_0, 0)$
$\mathbf{D(2)}$
s2 : $(s_0, l_0, 2)$
$\mathbf{M}(a, \mathrm{I})$
$\mathbf{M}(b, \mathrm{II})$
$\mathbf{D(1)}$
$\mathbf{M}(a, \mathrm{I})$
$\mathbf{M\_res}(a, \mathrm{III})$
s1 : $(s_1, l_1, 0)$
s4 : $(s_0, l_0, 3)$
$\mathbf{M}(a, \mathrm{I})$
$\mathbf{D(2)}$
$\mathbf{M\_KO}(a)$
$\mathbf{M}(a, \mathrm{I})$
$\bot$
s3 : $(s_1, l_1, 2)$
$\mathbf{D(1)}$
$\top$
s5 : $(s_1, l_1, 3)$
$\mathbf{M}(b, \mathrm{II})$
$\mathbf{M}(b, \mathrm{II})$

Figure 7. An ASMC $\mathcal{M}$, a DTA $\mathcal{A}$, and their TRG

We now define TRS as the set of nodes reachable from the set of states $(s, l, 0)$, for all $s \in S$ and $l \in Init$, with $s \models_\Sigma \Lambda(l)$, by traversing the arcs expressed by the four rules above (note that we consider all states $s \in S$ because satisfaction needs to be checked on all states of $\mathcal{M}$). Then the TRG of $\mathcal{M} \times \mathcal{A}$ is defined as the graph over TRS where the arcs are described as above.

Observe that, if (s, l, c) is a node of the TRG, then any $(s, l, c + \delta)$ with $0 \leq \delta < \mathsf{next}(c) - c$ is a state of the MRP $N(t)$, and that a **D**-arc (resp. **M_res**-arc) to a node (s, l, c) means that upon event **D** (resp. **M_res**) the state of $\mathcal{M} \times \mathcal{A}$ is exactly (s, l, c), while if the same state is entered through an **M**-arc, the state of the process can be $(s, l, c + \delta)$ for any $0 < \delta < \mathsf{next}(c)$.

The upper part of Figure 7 shows an ASMC $\mathcal{M}$ and a DTA $\mathcal{A}$. DTA edges have been tagged with roman numerals to cross-reference them in the TRG of $\mathcal{M} \times \mathcal{A}$ shown in the lower part. Let us consider two paths in the TRG and, for each path, the corresponding realizations in the stochastic process $\mathcal{M} \times \mathcal{A}$. The path $\mathsf{s0}, \mathsf{s1}, \mathsf{s3}, \top$ corresponds to process evolutions in which an a-event occurs with clock x in $(0, 2)$ and then time elapses until clock reaches 3. Note that the intermediate state $\mathsf{s3}$ corresponds to reaching time 2. The path $\mathsf{s0}, \mathsf{s1}, \mathsf{s1}, \bot$ captures those process evolutions in which an a-event is followed by a b-event and then again an a-event, all occurring with a clock in $(0, 2)$. As the last event cannot be mimicked by the DTA, the process reaches $\bot$.

3.3 Building the embedded DTMC

To compute the probability of reaching $\top$, we need to identify in the TRG the states of the DTMC Y and the associated transition probabilities. The states are defined according to the specification of the MRP.

Definition 3.3 *Let* $\mathsf{s} \in TRS$*. Then* s *is a state of the DTMC embedded into the MRP* (Y, T) *if either:*

1. $\mathsf{s} = (s, l, c)$ *with* $l \in Init$ *and* $c = 0$ *(initial states),*

2. s *can be entered by an arc labelled* **D** *or* **M_res***,*

3. $\mathsf{s} = \top$*, or*

4. $\mathsf{s} = \bot$*.*

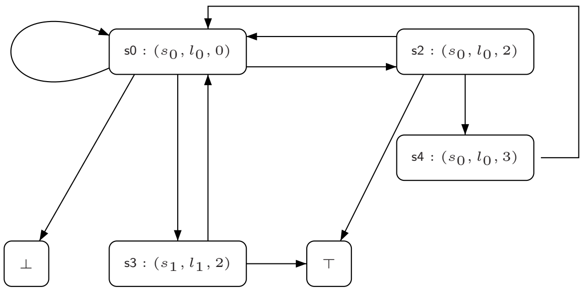

Figure 8. The embedded DTMC

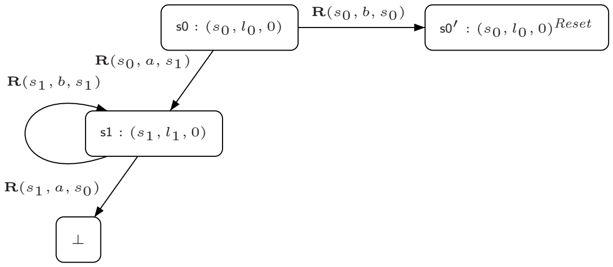
s0 : $(s_0, l_0, 0)$
$\mathbf{R}(s_0, b, s_0)$
s0′ : $(s_0, l_0, 0)^{Reset}$
$\mathbf{R}(s_0, a, s_1)$
$\mathbf{R}(s_1, b, s_1)$
s1 : $(s_1, l_1, 0)$
$\mathbf{R}(s_1, a, s_0)$
$\bot$

Figure 9. The subordinated CTMC with respect to $(s_0, l_0, 0)$

Note that not all states of the TRG are states of the DTMC: indeed in the example of Figure 7 state s5 is not a state of the DTMC. Intuitively speaking, in order to reach s5 the ASMC must perform an a-event *after* the clock has reached 3.

We have represented in Figure 8 the graph associated with the embedded DTMC of the example depicted in Figure 7. An arc between two states means that there is a non null probability to reach the destination from the source without going through a regeneration point. For instance, there is an edge from s4 to s0 because one possible path (in the TRG) goes through a **M** move followed by a **M_res** move triggering the regeneration point.

To compute the probabilities of the DTMC (i.e., to label the edges of the associated graph), we need to define, for each state $(s, l, c) \in TRS \setminus \{\bot, \top\}$ of the DTMC, how the process $\mathcal{M} \times \mathcal{A}$ can evolve before reaching the next regeneration point. This (transient) behavior is driven by the *subordinated* CTMC $\mathcal{C}_{(s,l,c)}$, which describes the evolution of the process from (s, l, c) until a successive state of $\mathcal{M} \times \mathcal{A}$ is reached, either due to a state change in $\mathcal{M}$, due to the clock having reached $\mathtt{next}(c)$, or due to the clock being reset.

The states of the subordinated CTMC $\mathcal{C}_{(s,l,c)}$ can be computed using, again, the TRG. From (s, l, c) we take in the TRG the transitive closure over arcs of type **M**, possibly followed by a **M_res**-arc or a **M_KO**-arc. More formally, the states of the subordinated CTMC $\mathcal{C}_{(s,l,c)}$ are defined as follows.

- $(s, l, c) \in \mathcal{C}_{(s,l,c)}$;
- $(s', l', c) \in \mathcal{C}_{(s,l,c)}$ if there exists a path in the TRG from (s, l, c) to (s', l', c) in which all arcs are of type **M**;
- $\top \in \mathcal{C}_{(s,l,c)}$ if there exists a path in the TRG from (s, l, c) to $\top$ in which either all arcs are of type **M**, or the path ends in an **M_res**-arc and all preceding arcs (if any) are of type **M**;
- $(s', l', 0)^{Reset} \in \mathcal{C}_{(s,l,c)}$ if there exists a (possibly empty) path in the TRG from (s, l, c) to (s'', l'', c) of arcs all of type **M**, and an arc from (s'', l'', c) to $(s', l', 0)$ of type **M_res**;
- $\bot \in \mathcal{C}_{(s,l,c)}$ if there exists a (possible empty) path in the TRG from (s, l, c) to (s'', l'', c) of arcs all of type **M**, and a **M_KO** arc from (s'', l'', c) to $\bot$.

We distinguish in the subordinated CTMCs the state $(s, l, 0)^{Reset}$, entered upon a clock reset, from the state $(s, l, 0)$ entered through a Markovian transition. Indeed the first state corresponds to the next regeneration point whereas the second one is only an intermediate state (see below). Observe that when $c > 0$ it is never the case that (s, l, c) can be entered through a non-Markovian transition[1]. The subset of "reset" states is denoted $Reset$. Summarizing, the states of the subordinated CTMC $\mathcal{C}_{(s,l,c)}$ are of the form $\perp$, $\top$, $(s', l', 0)^{Reset}$ or (s', l', c') for $s' \in S$, $l' \in L$ and $c' \in \{0, \ldots, c\}$. We shall indicate with s a generic state of the DTMC, of the various forms indicated above. Note that there is a subordinated CTMC built only for those states s of form $\mathsf{s} = (s, l, c)$ that we will denote as $\mathcal{C}_{(s,l,c)}$ or, equivalently, as $\mathcal{C}_{\mathsf{s}}$.

Figure 9 depicts the subordinated chain $\mathcal{C}_{(s_0,l_0,0)}$ for state $(s_0, l_0, 0)$, Observe that this subordinated chain is derived from the TRG but is not a subgraph of it due to the duplication of state $(s_0, l_0, 0)$. This CTMC represents all behaviors during the evolution of clock x in $(0, 2)$ until a regeneration point is reached. The (non-time-triggered) regeneration points correspond to the absorbing states $\perp$ and $(s_0, l_0, 0)^{Reset}$. Let us interpret the state of this CTMC at time 2. If it is $\perp$ (respectively, $(s_0, l_0, 0)^{Reset}$) then it means that the next regeneration point is $\perp$ (respectively, $(s_0, l_0, 0)$) since we have reached it *before* 2. If this state is $(s_0, l_0, 0)$ (respectively, $(s_1, l_1, 0)$), it means that the next regeneration point corresponds to $x = 2$ and we follow the corresponding $\mathbf{D}$ edge in the TRG to determine the next regeneration point, here $(s_0, l_0, 2)$ (respectively, $(s_1, l_1, 2)$). Therefore the probabilities of DTMC transitions from $(s_0, l_0, 0)$ are obtained from the transient probability distribution of the subordinated chain at time 2.

We can now generalize the example to consider the rates of the DTMC in the general case. Let $P_{\mathsf{s},\mathsf{s}'}$ be the transition probabilities of the DTMC. The elements of $P_{\mathsf{s},\mathsf{s}'}$ are computed using the subordinated CTMC $\mathcal{C}_{\mathsf{s}}$, and all the elements of row s of P are computed on the same subordinated CTMC $\mathcal{C}_{\mathsf{s}}$. Rows corresponding to the DTMC states $\top$ and $\perp$ are obviously identically zero, since they are absorbing. Let $\mathsf{s} = (s, l, c)$, we denote by $\pi_{\mathsf{s}}(\tau)$ the transient (respectively, steady-state) distribution of $\mathcal{C}_{\mathsf{s}}$ at time τ when τ is finite (respectively, when $\tau = \infty$).

We are now in position to give the formulae for the non null entries of P. By convention in the following formulae, $\perp^{Reset} = \perp$, $\top^{Reset} = \top$ and when $\mathsf{s}' \notin \mathcal{C}_{\mathsf{s}}$ then $\pi_{\mathsf{s}}(\tau)(\mathsf{s}') = 0$.

- If $c < c_m$ then $P_{\mathsf{s},\mathsf{s}'}$ equals:

$$\sum_{\mathsf{s}'' \in \mathcal{C}_{\mathsf{s}} \setminus Reset \wedge \mathsf{s}'' \xrightarrow{\mathbf{D}} \mathsf{s}'} \pi_{\mathsf{s}}(\mathsf{next}(c) - c)(\mathsf{s}'') \quad + \quad \pi_{\mathsf{s}}(\mathsf{next}(c) - c)(\mathsf{s}'^{Reset}) \, .$$

The first term (sum over s'') corresponds to the case where the next regeneration is triggered by $x = \mathsf{next}(c)$ and we look for $\mathbf{D}$ edges to determine the next regeneration point. Remember that $Reset$ is the set of $(\cdot, \cdot, 0)^{Reset}$ states. The second term corresponds to a regeneration point obtained by a clock reset ($(s', l', 0)^{Reset}$) or by reaching $\{\perp, \top\}$.

- If $c = c_m$ then

$$P_{\mathsf{s},\mathsf{s}'} = \pi_{\mathsf{s}}(\infty)(\mathsf{s}'^{Reset})$$

  for $\mathsf{s}' \in \{\perp, \top\} \cup \{(s', l', 0)\}$.

When $c = c_m$, the only way to obtain a regeneration point is to reset the clock or to reach $\{\perp, \top\}$.

The first case requires transient analysis of the subordinated CTMCs, which is usually performed by uniformization [28]. The second case only requires steady-state analysis which is generally less computationally expensive.

Note that there are two peculiarities of the embedded DTMC. First, we can re-enter the same state due to a clock reset. This has no effect on the computation. Second, the transition matrix can be substochastic, because for some DTMC states there is a non-null probability to never reach another state of the MRP. Again, this is not problematic, because the reachability probability computation with a substochastic matrix is identical to that with a stochastic transition matrix.

Finally, as often in a probabilistic setting, checking whether the set of paths of $\mathcal{M}$ accepted by $\mathcal{A}$ has probability 0 or 1 can be performed without any numerical computation. The only relevant information in the DTMC, given its transition probability $\mathbf{P}$, is (for every pair of states (s, s')) whether $\mathbf{P}(s, s') > 0$, and this information is obtained by a simple examination of the TRG.

4 Expressiveness of CSL$^{\text{TA}}$

In this section we study the relationship between CSL$^{\text{TA}}$, CSL [9], and asCSL [10]. Formulae interpreted on ASMCs are described as being equivalent if, for any ASMC, the same states of the ASMC satisfy the formulae. Formally, we say that the

[1] Note that in the TRG a **M_res** transition corresponds in Deterministic Stochastic Petri Nets to the case of an exponential transition that preempts a deterministic transition and then immediately re-enables it, which, as explained in [26], requires a duplication of the states of the subordinated CTMC.

formula Φ_1 (of the logic L_1, with the satisfaction relation $\models_{L_1}$) and Φ_2 (of the logic L_2, with the satisfaction relation $\models_{L_2}$) are *equivalent* if, for any ASMC $\mathcal{M}$, and for any state s of $\mathcal{M}$, we have $\mathcal{M}, s \models_{L_1} \Phi_1$ if and only if $\mathcal{M}, s \models_{L_2} \Phi_2$. The logic L_1 is *at least as expressive as* the logic L_2 if, for each formula Φ_2 of the logic L_2, there exists a formula Φ_1 of the logic L_1 such that Φ_1 and Φ_2 are equivalent. The logic L_1 is *strictly more expressive than* the logic L_2 if L_1 is at least as expressive as L_2 and there exists a formula Φ_1 of L_1 such that there does not exist an equivalent formula Φ_2 of logic L_2. Given an ASMC $\mathcal{M}$ with state set S and the satisfaction relation $\models_L$ of the logic L, let $Sat_L^{\mathcal{M}}(\Phi) = \{s \in S \mid \mathcal{M}, s \models_L \Phi\}$ (when $\mathcal{M}$ is clear from the context we write $Sat_L(\Phi)$).

4.1 CSL$^{\text{TA}}$ is at least as expressive as CSL

In this section, we recall the definition of CSL [9].

Definition 4.1 *The syntax of CSL is defined as follows:*

$$\Phi ::= p \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathcal{S}_{\sim\lambda}(\Phi) \mid \mathcal{P}_{\sim\lambda}(\mathcal{X}^I \Phi) \mid \mathcal{P}_{\sim\lambda}(\Phi \mathcal{U}^I \Phi)$$

where $a \in AP$ is an atomic proposition, $I \subseteq \mathbb{R}_{\geq 0}$ is a nonempty interval, $\sim \in \{<, \leq, \geq, >\}$ is a comparison operator, and $\lambda \in [0,1]$ is a probability.

For any infinite path $\sigma = s_0 \xrightarrow{a_0,\tau_0} s_1 \xrightarrow{a_1,\tau_1} \cdots$ and $t \in \mathbb{R}_{\geq 0}$, if i the smallest index such that $t \leq \sum_{j=0}^{i} \tau_j$, then we let $\sigma@t = s_i$; that is, $\sigma@t$ is used to denote the state along σ occupied at time t.

Definition 4.2 *For $\mathcal{M} = \langle S, Act, AP, lab, \mathbf{R}\rangle$, and state $s \in S$, the satisfaction relation $\models_{\text{CSL}}$ is defined as follows:*

$$\begin{array}{lllll}
\mathcal{M}, s & \models_{\text{CSL}} & p & \Leftrightarrow & p \in lab(s) \\
\mathcal{M}, s & \models_{\text{CSL}} & \Phi_1 \wedge \Phi_2 & \Leftrightarrow & \mathcal{M}, s \models_{\text{CSL}} \Phi_1 \text{ and } \mathcal{M}, s \models_{\text{CSL}} \Phi_2 \\
\mathcal{M}, s & \models_{\text{CSL}} & \neg\Phi & \Leftrightarrow & \mathcal{M}, s \not\models_{\text{CSL}} \Phi \\
\mathcal{M}, s & \models_{\text{CSL}} & \mathcal{S}_{\sim\lambda}(\Phi) & \Leftrightarrow & \sum_{s' \in Sat_{\text{CSL}}^{\mathcal{M}}(\Phi)} \pi(s, s') \sim \lambda \\
\mathcal{M}, s & \models_{\text{CSL}} & \mathcal{P}_{\sim\lambda}(\varphi) & \Leftrightarrow & \Pr_s^{\mathcal{M}}\{\sigma \in Path^{\mathcal{M}}(s) \mid \mathcal{M}, \sigma \models_{\text{CSL}} \varphi\} \sim \lambda \\
\mathcal{M}, \sigma & \models_{\text{CSL}} & \mathcal{X}^I \Phi & \Leftrightarrow & \mathcal{M}, \sigma(1) \models_{\text{CSL}} \Phi \text{ and } \sigma = s \xrightarrow{a,\tau} \sigma' \text{ with } \tau \in I \\
\mathcal{M}, \sigma & \models_{\text{CSL}} & \Phi_1 \mathcal{U}^I \Phi_2 & \Leftrightarrow & \exists t \in I. \mathcal{M}, \sigma@t \models_{\text{CSL}} \Phi_2 \text{ and } \forall t' \in [0,t). \mathcal{M}, \sigma@t' \models_{\text{CSL}} \Phi_1.
\end{array}$$

In the following, when clear from the context, we write $s \models_{\text{CSL}} \Phi$ for $\mathcal{M}, s \models_{\text{CSL}} \Phi$ and $\sigma \models_{\text{CSL}} \Phi$ for $\mathcal{M}, \sigma \models_{\text{CSL}} \Phi$. The following proposition shows that CSL$^{\text{TA}}$ is at least as expressive as CSL.

Proposition 4.3 *For any formula Φ of CSL there is a formula Φ' of CSL$^{\text{TA}}$ equivalent to Φ. The size of Φ' is linear with respect to the size of Φ.*

Proof. The semantics of constructors for state formulae are identical for CSL and CSL$^{\text{TA}}$; therefore it suffices to prove that any path formula of CSL is equivalent to some path formula of CSL$^{\text{TA}}$. The idea of the proof is to translate the path operator $\mathcal{X}^{[\alpha,\beta]}\Phi$ of CSL with the DTA $\mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}}$ of Figure 4, and the path operator $\Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2$ with the DTA $\mathcal{A}_{\Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2}$ of Figure 5. In the following, we concentrate on the case in which the time interval of a CSL formula is of the form $[\alpha, \beta]$, where $\alpha > 0$ (the translation of path operators with time intervals other than $[\alpha, \beta]$ is similar, and will be discussed briefly at the end of the proof). Therefore our task consists in showing that, for a given ASMC $\mathcal{M} = \langle S, Act, AP, lab, \mathbf{R}\rangle$, a given state $s \in S$, and an infinite path $\sigma \in Path^{\mathcal{M}}(s)$, we have:

1. $\sigma \models_{\text{CSL}} \Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2$ iff $\sigma \in AccPath^{\mathcal{M}}(s, \mathcal{A}_{\Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2})$, and
2. $\sigma \models_{\text{CSL}} \mathcal{X}^{[\alpha,\beta]}\Phi$ iff $\sigma \in AccPath^{\mathcal{M}}(s, \mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}\Phi})$.

Consider **case (1)**. We show first that $\sigma \models_{\text{CSL}} \Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2$ implies $\sigma \in AccPath^{\mathcal{M}}(s, \mathcal{A}_{\Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2})$. Let $\sigma_{\mathcal{M}} = s_0 \xrightarrow{a_0,\tau_0} s_1 \xrightarrow{a_1,\tau_1} \cdots$ be a path such that $\sigma_{\mathcal{M}} \models_{\text{CSL}} \Phi_1\mathcal{U}^{[\alpha,\beta]}\Phi_2$. Then, by Definition 4.2, there exists $t \in [\alpha, \beta]$ such that $\sigma_{\mathcal{M}}@t \models_{\text{CSL}} \Phi_2$ and $\sigma_{\mathcal{M}}@t' \models_{\text{CSL}} \Phi_1$ for all $t' < t$. There are two cases to consider:

(1.a) Φ_2 is satisfied when the value of the clock x reaches α;

(1.b) Φ_2 is not yet satisfied when the value of the clock x reaches α.

Consider **case (1.a)**. Observe that there exists some $i \in \mathbb{N}$ such that $s_i \models_{\mathrm{CSL}} \Phi_1 \wedge \Phi_2$, $s_j \models_{\mathrm{CSL}} \Phi_1$ for all $j < i$, and $\sum_{k=0}^{i-1} \tau_k < \alpha \leq \sum_{k=0}^{i} \tau_k$. We assert that the ASMC path $\sigma_{\mathcal{M}}$ is accepted by the path $\sigma_{\mathcal{A}} = (l_0, \bar{x}_0) \xrightarrow{\tau_0, e_{00}} \ldots (l_0, \bar{x}_{i-1}) \xrightarrow{\tau_{i-1}, e_{00}} (l_0, \bar{x}_i) \xrightarrow{\tau_i, e_{02}} (l_2, \bar{x}_{i+1})$ of $\mathcal{A}_{\Phi_1 \mathcal{U}^{[\alpha,\beta]} \Phi_2}$ (that is, to accept $\sigma_{\mathcal{M}}$, the DTA performs i loops of the edge e_{00}, then traverses the edge e_{02}). To verify that $\sigma_{\mathcal{M}}$ is accepted by $\sigma_{\mathcal{A}}$, consider Definition 2.7. First, observe that $l_0 \in \mathit{Init}$, $\bar{x}_0 = 0$ and $l_2 \in \mathit{Final}$. Second, recalling that $\Lambda(l_0) = \Phi_1$, we observe that $s_j \models_{\mathrm{CSL}} \Lambda(l_0)$ for all $j \leq i$. Furthermore, recalling that $\Lambda(l_2) = \Phi_2$, we observe that $s_i \models_{\mathrm{CSL}} \Lambda(l_2)$. Third, we have that $a_j \in \mathit{Act}$, and hence $a_j \in \mathsf{action}(e_{00})$, for all $j < i$. The final requirements of Definition 2.7, which concern the durations of $\sigma_{\mathcal{M}}$ and $\sigma_{\mathcal{A}}$, follow directly from the observation that the durations of the transitions of $\sigma_{\mathcal{A}}$ are equal to the durations of the first i transitions of $\sigma_{\mathcal{M}}$.

Now consider **case (1.b)**. In this case, there exists $i \in \mathbb{N}$ such that $s_i \models_{\mathrm{CSL}} \Phi_2$, $s_j \models_{\mathrm{CSL}} \Phi_1$ for all $j < i$, and $\alpha < \sum_{k=0}^{i-1} \tau_k < \beta$. Furthermore, we let $i^\alpha < i$ be the largest index for which $\sum_{k=0}^{i^\alpha - 1} \tau_k \leq \alpha$. Intuitively, the state s_{i^α} will be the state along $\sigma_{\mathcal{M}}$ when α time units have elapsed. Let $\tau' = \alpha - \sum_{k=0}^{i^\alpha - 1} \tau_k$ and $\tau'' = \tau_{i^\alpha} - \tau'$. We claim that the ASMC path $\sigma_{\mathcal{M}}$ is accepted by the path $\sigma_{\mathcal{A}} = (l_0, \bar{x}_0) \xrightarrow{\tau_0, e_{00}} \ldots \xrightarrow{\tau_{i^\alpha - 1}, e_{00}} (l_0, \bar{x}_{i^\alpha}) \xrightarrow{\tau', e_{01}} (l_1, \bar{x}_{i^\alpha + 1}) \xrightarrow{\tau'', e_{11}} (l_1, \bar{x}_{i^\alpha + 2}) \xrightarrow{\tau_{i^\alpha + 1}, e_{11}} \ldots \xrightarrow{\tau_{i-1}, e_{11}} (l_1, \bar{x}_i) \xrightarrow{\tau_i, e_{12}} (l_2, \bar{x}_{i+1})$ of $\mathcal{A}_{\Phi_1 \mathcal{U}^{[\alpha,\beta]} \Phi_2}$ (that is, to accept $\sigma_{\mathcal{M}}$, the DTA performs i^α loops of the edge e_{00}, then traverses the edge e_{01}, then performs $i - (i^\alpha + 1)$ loops of the edge e_{11}, then traverses the edge e_{12}). Consider Definition 2.7: we note that the index described in point 2 of Definition 2.7 is i, and the function $\kappa : \{0, \ldots, i+1\} \to \{0, \ldots, i\}$ is defined by $\kappa(j) = j$ for all $j \leq i^\alpha$, and $\kappa(j) = j - 1$ for all $i^\alpha < j \leq i + 1$. We now verify that the choice of $\sigma_{\mathcal{A}}$, index i, and function κ satisfies the conditions of Definition 2.7. First, observe that $l_0 \in \mathit{Init}$, $\bar{x}_0 = 0$ and $l_2 \in \mathit{Final}$. Second, recalling that $\Lambda(l_0) = \Phi_1$ ($\Lambda(l_1) = \Phi_1 \wedge \neg\Phi_2$, $\Lambda(l_2) = \Phi_2$, respectively), we observe that $s_j \models_{\mathrm{CSL}} \Lambda(l_0)$ for all $j \leq i^\alpha$ ($s_j \models_{\mathrm{CSL}} \Lambda(l_1)$ for all $i^\alpha < j < i$, $s_i \models_{\mathrm{CSL}} \Lambda(l_2)$, respectively). Third, we have that $a_j \in \mathit{Act}$, and hence $a_j \in \mathsf{action}(e_{00})$, $a_j \in \mathsf{action}(e_{11})$ and $a_i \in \mathsf{action}(e_{12})$, for all $j \leq i$. The final requirements of Definition 2.7, concerning the durations of $\sigma_{\mathcal{M}}$ and $\sigma_{\mathcal{A}}$, follow by the following facts: for all $j \leq i$ such that $j \neq i^\alpha$, we have that the duration of the jth transition of $\sigma_{\mathcal{M}}$ is equal to the duration of the $\kappa(j)$th transition of $\sigma_{\mathcal{M}}$; furthermore, $\tau_{i^\alpha} = \tau' + \tau''$.

The **reverse direction of case (1)** follows in a similar manner, and we omit the details.

Consider **case (2)**. We show that $\sigma \models_{\mathrm{CSL}} \mathcal{X}^{[\alpha,\beta]}\Phi$ implies $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}\Phi})$. Let $\sigma_{\mathcal{M}}$ be a path such that $\mathcal{M}, \sigma_{\mathcal{M}} \models_{\mathrm{CSL}} \mathcal{X}^{[\alpha,\beta]}\Phi$. Then, by Definition 4.2, we have $\sigma_{\mathcal{M}} = s \xrightarrow{a, \tau} \sigma'$ for some $a \in \mathit{Act}$, $\tau \in I$ and path σ', where $\sigma_{\mathcal{M}}(1) \models_{\mathrm{CSL}} \Phi$. By the definition of the DTA $\mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}\Phi}$, there exists a path $(l_0, 0) \xrightarrow{\tau, e} (l_1, \tau)$ where $e = (l_0, \alpha \leq x \leq \beta, \mathit{Act}, \emptyset, l_1)$ is the edge from l_0 to l_1. From the fact that $l_0 \in \mathit{Init}$, $l_1 \in \mathit{Final}$, and $\sigma_{\mathcal{M}}(1) \models_{\mathrm{CSL}} \Phi$, we have that $\sigma_{\mathcal{M}}$ is accepted by $\mathcal{A}_{\mathcal{X}^{[\alpha,\beta]}\Phi}$ according to the criteria of Definition 2.7. The **reverse direction of case (2)** follows in a similar manner.

We now consider briefly the case for other types of time intervals. For the Next operator, the DTA of Figure 4 requires only modifications to the guard of its single edge (for example, the time interval (α, ∞) is represented by the guard $x > \alpha$). Similarly, open or half-open time intervals of the Until operator can be represented by changing the associated inequalities of constraints from non-strict to strict: for example, the time interval $(\alpha, \beta]$ can be represented by changing the constraint $x \leq \alpha$ to $x < \alpha$ in the DTA of Figure 5. For a time interval of the form $[\alpha, \infty)$ or (α, ∞) for $\alpha > 0$, the guards of the form $x \leq \beta$ in the DTA of Figure 5 are changed to `true`. Instead, for a time interval of the form $[0, \beta]$ or $[0, \beta)$, the location l_0 and its outgoing edges are removed, and both l_1 and l_2 become initial locations.

Finally, the assertion on formula sizes is straightforward. □

We observe that the verification of a CSL formula of the form $\mathcal{P}_{\sim\lambda}(\Phi_1 \mathcal{U}^I \Phi_2)$ and a CSL$^{\mathrm{TA}}$ formula of the form $\mathcal{P}_{\sim\lambda}(\mathcal{A}_{\Phi_1 \mathcal{U}^I \Phi_2})$ involve similar computation steps: for example, in the case of $I = [\alpha, \beta]$ with $\alpha > 0$, a transient analysis of two CTMCs is required, both in the CSL model-checking algorithm of [9], and in the CSL$^{\mathrm{TA}}$ model-checking algorithm of Section 3. The computational complexity of model checking CSL$^{\mathrm{TA}}$ properties transformed from equivalent CSL properties is the same as that for model checking the original CSL properties with the algorithm of [9].

4.2 CSL$^{\mathrm{TA}}$ is at least as expressive as asCSL

In this section, we recall the stochastic temporal logic asCSL [10], and show that every asCSL formula can be expressed as a CSL$^{\mathrm{TA}}$ formula.

4.2.1 Definition of asCSL

First we present the syntax and semantics of asCSL. In contrast to the original presentation of asCSL in [10], we consider *nondeterministic program automata* as path operators, as opposed to regular expressions (called programs in [10]). As asCSL programs can be translated into nondeterministic program automata, the presentation of asCSL is as general as the original presentation with regular expressions. Also note that we use the special action $\surd$, which in a similar way to $\sharp$, allows a transition in the automaton without a corresponding transition in the ASMC. Note the distinction between $\sharp$-labelled transitions of DTA and $\surd$-labelled transitions of nondeterministic program automata: the latter are not triggered by behavior of the ASMC, whereas, in contrast, $\sharp$-labelled transitions are triggered by the passage of time.

Definition 4.4 *A* nondeterministic program automaton *(NPA) is a tuple $\mathcal{N} = \langle Z, \Xi, \delta, Z_{Init}, Z_F \rangle$, where Z is a finite set of states with the set $Z_{Init} \subseteq Z$ of initial states and the set $Z_F \subseteq Z$ of final states, Ξ is a finite input alphabet, and $\delta : Z \times \Xi \to 2^Z$ is a transition function. We say that $\mathcal{N}$ is a* deterministic program automaton *(DPA) if $|Z_{Init}| = 1$ and $|\delta(z, u)| \leq 1$ for each $z \in Z$ and $u \in \Xi$.*

Definition 4.5 *The syntax of asCSL is defined as follows:*

$$\Phi ::= p \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathcal{S}_{\sim\lambda}(\Phi) \mid \mathcal{P}_{\sim\lambda}(\mathcal{N}(\Xi)^I)$$

where $a \in AP$ is an atomic proposition, $I \subseteq \mathbb{R}_{\geq 0}$ is a nonempty interval, $\sim \in \{<, \leq, \geq, >\}$ is a comparison operator, $\lambda \in [0, 1]$ is a probability, and $\mathcal{N}(\Xi)$ is a NPA with input alphabet Ξ such that $\Xi \subseteq \{(\Phi, b) \mid \Phi$ is an asCSL formula $\wedge\, b \in Act \cup \{\surd\}\}$.

We write $z \xrightarrow{u} z'$ to denote $z' \in \delta(z, u)$ (that is, to denote a transition of an NPA).

Definition 4.6 *A* run *of an NPA $\mathcal{N}$ is a (finite) sequence $r = z_0 \xrightarrow{u_0} z_1 \xrightarrow{u_1} \cdots \xrightarrow{u_{n-1}} z_n$ of transitions of $\mathcal{N}$. If $z_0 \in Z_{Init}$ and $z_n \in Z_F$ then we say that the run r is* accepting. *We use $last(r)$ to denote the last state z_n of r. For a state z of $\mathcal{N}$, let $\mathsf{Runs}^{\mathcal{N}}(z)$ be the set of runs of $\mathcal{N}$ with the first state z, and for $Z' \subseteq Z$, let $\mathsf{Runs}^{\mathcal{N}}(Z') = \bigcup_{z \in Z'} \mathsf{Runs}^{\mathcal{N}}(z)$.*

Definition 4.7 *For $\mathcal{M} = \langle S, Act, AP, lab, \mathbf{R} \rangle$, and state $s \in S$, the satisfaction relation $\models_{\mathrm{asCSL}}$ is defined as follows:*

$$\begin{array}{lcl}
\mathcal{M}, s \models_{\mathrm{asCSL}} p & \Leftrightarrow & p \in lab(s) \\
\mathcal{M}, s \models_{\mathrm{asCSL}} \neg\Phi & \Leftrightarrow & \mathcal{M}, s \not\models_{\mathrm{asCSL}} \Phi \\
\mathcal{M}, s \models_{\mathrm{asCSL}} \Phi_1 \wedge \Phi_2 & \Leftrightarrow & \mathcal{M}, s \models_{\mathrm{asCSL}} \Phi_1 \textit{ and} \\
 & & \mathcal{M}, s \models_{\mathrm{asCSL}} \Phi_2 \\
\mathcal{M}, s \models_{\mathrm{asCSL}} \mathcal{S}_{\sim\lambda}(\Phi) & \Leftrightarrow & \sum_{s' \in Sat^{\mathcal{M}}_{\mathrm{asCSL}}(\Phi)} \pi(s, s') \sim \lambda \\
\mathcal{M}, s \models_{\mathrm{asCSL}} \mathcal{P}_{\sim\lambda}(\mathcal{N}(\Xi)^I) & \Leftrightarrow & \mathrm{Pr}^{\mathcal{M}}_s(AccPath^{\mathcal{M}}(s, \mathcal{N}(\Xi)^I)) \sim \lambda
\end{array}$$

where $AccPath^{\mathcal{M}}(s, \mathcal{N}(\Xi)^I)$ is defined in the following way. Let z be a state of $\mathcal{N}$ and σ be a finite path of $\mathcal{M}$. We define $\mathsf{Runs}^{\mathcal{N}}(z, \sigma)$ as the greatest set of runs $z \xrightarrow{\Phi_0 b_0} z_1 \xrightarrow{\Phi_1 b_1} \cdots \xrightarrow{\Phi_{n-1} b_{n-1}} z_n$ such that:

1. *$z \in \mathsf{Runs}^{\mathcal{N}}(z, \sigma)$ if and only if $|\sigma| = 0$;*

2. *if $z \xrightarrow{\Phi_0 b_0} z_1 \xrightarrow{\Phi_1 b_1} \cdots \xrightarrow{\Phi_{n-1} b_{n-1}} z_n \in \mathsf{Runs}^{\mathcal{N}}(z, \sigma)$ and $n \geq 1$, then:*

   - *$\mathcal{M}, \sigma(0) \models_{\mathrm{asCSL}} \Phi_0$;*

   - *if $b_0 \in Act$, then $\sigma = s \xrightarrow{b_0, \tau} \sigma'$ with $z_1 \xrightarrow{\Phi_1 b_1} \cdots \xrightarrow{\Phi_{n-1} b_{n-1}} z_n \in \mathsf{Runs}^{\mathcal{N}}(z_1, \sigma')$;*

   - *if $b_0 = \surd$, then $z_1 \xrightarrow{\Phi_1 b_1} \cdots \xrightarrow{\Phi_{n-1} b_{n-1}} z_n \in \mathsf{Runs}^{\mathcal{N}}(z_1, \sigma)$.*

For the set $Z' \subseteq Z$ of states of $\mathcal{N}$, we let $\mathsf{Runs}^{\mathcal{N}}(Z', \sigma) = \bigcup_{z \in Z'} \mathsf{Runs}^{\mathcal{N}}(z, \sigma)$. The set of ASMC paths accepted by $\mathcal{N}$ with time interval $I \subseteq \mathbb{R}_{\geq 0}$, denoted by $AccPath^{\mathcal{M}}(s, \mathcal{N}^I)$, is defined as:

$$\{\sigma \in Path^{\mathcal{M}}(s) \mid \exists \textit{ finite prefix } \sigma_{\mathrm{fin}} \textit{ of } \sigma \textit{ s.t. } \exists \textit{ accepting } r \in \mathsf{Runs}^{\mathcal{N}}(Z_{Init}, \sigma_{\mathrm{fin}}) \textit{ and } \tau(\sigma_{\mathrm{fin}}) \in I\}\ .$$

When clear from the context, we write $\mathsf{Runs}(Z', \sigma)$ instead of $\mathsf{Runs}^{\mathcal{N}}(Z', \sigma)$.

4.2.2 From NPA to DTA

To show that every asCSL formula can be expressed as a CSL$^{\text{TA}}$ formula, it suffices to show how a formula $\mathcal{P}_{\sim\lambda}(\mathcal{N}(\Xi)^I)$ can be encoded as a CSL$^{\text{TA}}$ formula $\mathcal{P}_{\sim\lambda}(\mathcal{A})$, similarly to the translation from CSL to CSL$^{\text{TA}}$ of Section 4.1. In order to obtain the required DTA $\mathcal{A}$ from $\mathcal{N}(\Xi)^I$, two steps are required: first, it is necessary to construct a DPA, which will be used as the graph of $\mathcal{A}$, from the NPA $\mathcal{N}(\Xi)$, using a standard subset construction; then it is necessary to represent the time interval I within $\mathcal{A}$ using clock guards, in particular to constrain the global time of entry to final locations to those times in the interval I.

We first present the determinization of a NPA. Note that $\surd$-transitions are eliminated from the NPA to obtain the determinized automaton; unlike the case of $\sharp$-transitions in DTA, $\surd$-transitions do not have priority over other transitions. Hence, if a state z has an outgoing a-transition (for $a \in Act$) and an outgoing $\surd$-transition leading to a state with an outgoing a-transition, when reading an a-transition of an ASMC, there will be a nondeterministic choice between the a-transition and the $\surd$-transition from z. We choose to eliminate sequences of $\surd$-transitions in order to avoid such situations, noting that we also have to consider the case in which sequences of $\surd$-transitions which reach final states are taken after an action.

We require the following notation. Let $\mathcal{N} = \langle Z, \Xi, \delta, Z_{Init}, Z_F \rangle$ be an NPA, $Z' \subseteq Z$, Φ be an asCSL formula, and $a \in Act$. Then we let $\mathsf{Runs}(Z', (\Phi, a))$ equal:

$$\{z_0 \xrightarrow{\Phi_0 \surd} \cdots \xrightarrow{\Phi_{n-2} \surd} z_{n-1} \xrightarrow{\Phi_{n-1} a} z_n \in \mathsf{Runs}(Z') \mid z_0 \in Z' \wedge \Phi \Rightarrow \bigwedge_{i<n} \Phi_i\} .$$

Therefore $\mathsf{Runs}(Z', (\Phi, a))$ is the set of runs of $\mathcal{N}$ starting from a state in Z' which perform $\surd$-transitions before performing a single a-transition, where the conjunction of the asCSL formulae labelling transitions along the path is implied by Φ. We also let $\mathsf{Runs}(Z', (\Phi, \surd), Z_F)$ equal:

$$\{z_0 \xrightarrow{\Phi_0 \surd} \cdots \xrightarrow{\Phi_{n-1} \surd} z_n \in \mathsf{Runs}(Z') \mid z_0 \in Z' \wedge \Phi \Rightarrow \bigwedge_{i<n} \Phi_i \wedge z_n \in Z_F\} .$$

Hence $\mathsf{Runs}(Z', (\Phi, \surd), Z_F)$ is the set of runs from a state in Z' to a state in Z_F which perform $\surd$-transitions only, where the conjunction of formulae along the path is implied by Φ (it is possible to have a run of length 0 featuring a single state, which must be both in Z' and Z_F).

Before defining the DPA corresponding to an NPA, we must identify the set of formulae which can appear in transition labels of the DPA, and which will be used subsequently to define the set of state propositions of the DTA. This set of formulae will be constructed such that an ASMC path is accepted by at most one run of the DTA obtained from an NPA. In particular, we construct a set of disjoint state propositions. Formally, let Σ be the smallest set of asCSL formulae such that:

1. $\Phi_1 \wedge \Phi_2 \Leftrightarrow \texttt{false}$ for all $\Phi_1, \Phi_2 \in \Sigma$;
2. for each Φ such that $\mathsf{Runs}(Z, (\Phi, a)) \neq \emptyset$, for some $a \in Act$, or $\mathsf{Runs}(Z, (\Phi, \surd), Z_F) \neq \emptyset$, there exists $\{\Phi_1, \ldots, \Phi_n\} \subseteq \Sigma$ such that $\Phi \equiv \bigvee_{1 \leq i \leq n} \Phi_n$.

Definition 4.8 *Let $\mathcal{N} = \langle Z, \Xi, \delta, Z_{Init}, Z_F \rangle$ be an NPA. The* determinization *of $\mathcal{N}$ is the DPA $\det(\mathcal{N}) = \langle \mathsf{Q}, \Xi', \Delta, \mathsf{q}_{Init}, \mathsf{Q}_F \rangle$, where:*

- $\mathsf{Q} = 2^Z$, $\mathsf{q}_{Init} = Z_{Init}$*, and* $\mathsf{Q}_F = \{\mathsf{q} \in 2^Z \mid \exists \Phi \in \Sigma$ *s.t.* $\mathsf{Runs}(\mathsf{q}, (\Phi, \surd), Z_F) \neq \emptyset\}$*;*
- $\Xi' = \{(\Phi, a) \mid \Phi \in \Sigma \wedge a \in Act\}$*;*
- $\Delta : \mathsf{Q} \times \Xi' \to 2^{\mathsf{Q}}$ *is the transition function defined by* $\Delta(\mathsf{q}, (\Phi, a)) = \{z \in Z \mid \exists r \in \mathsf{Runs}(Z, (\Phi, a)) \wedge last(r) = z\}$ *for all* $\mathsf{q} \in \mathsf{Q}$ *and* $(\Phi, a) \in \Xi'$*.*

It can be verified that $|\Delta(\mathsf{q}, (\Phi, a))| \leq 1$ for all $\mathsf{q} \in \mathsf{Q}$ and $(\Phi, a) \in \Xi'$.

We now define the DTA $\mathcal{A}(\mathcal{N}^I)$ by pushing the asCSL state formulae featured in transition labels in $\det(\mathcal{N})$ into location labels of $\mathcal{A}(\mathcal{N}^I)$, and using the interval I in guards of edges leading directly to final locations (which correspond to the set E'' of DTA edges in the following definition). We also have to consider the case in which a final state of the NPA is reached at a time before the interval I (considered in the set E' of DTA edges below), which does not correspond to acceptance.

Definition 4.9 *Let $\mathcal{N}$ be an NPA, $[\alpha, \beta] \subseteq \mathbb{R}_{\geq 0}$ such that $\alpha > 0$, and $\det(\mathcal{N}) = \langle \mathsf{Q}, \Xi', \Delta, \mathsf{q}_{Init}, \mathsf{Q}_F \rangle$ be the determinization of $\mathcal{N}$. We let $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]}) = \langle \Sigma, Act, (\mathsf{Q} \times \Sigma) \cup (\overline{\mathsf{Q}_F} \times \Sigma), \Lambda, Init, Final, \to \rangle$ be such that:*

- $\overline{\mathsf{Q}_F} = \{\overline{\mathsf{q}} \mid \mathsf{q} \in \mathsf{Q}_F\}$;
- $\Lambda(\mathsf{q}, \Phi) = \Lambda(\overline{\mathsf{q}}, \Phi) = \Phi$ *for each* $\mathsf{q} \in \mathsf{Q}$ *and* $\Phi \in \Sigma$;
- $\mathit{Init} = \{(\mathsf{q}_{\mathit{Init}}, \Phi) \mid \Phi \in \Sigma\}$;
- $\mathit{Final} = \{(\overline{\mathsf{q}}, \Phi) \in \overline{\mathsf{Q}_F} \times \Sigma \mid \mathsf{q} \in \mathsf{Q}_F \wedge \mathsf{Runs}(\mathsf{q}, (\Phi, \surd), Z_F) \neq \emptyset\}$;
- $\rightarrow$ *is equal to the set:*

$$\bigcup_{(\mathsf{q},\Phi)\in\mathsf{Q}\times\Sigma, a\in Act, \Phi'\in\Sigma} E((\mathsf{q}, \Phi), a, \Phi') \cup E'((\mathsf{q}, \Phi), a, \Phi') \cup E''((\mathsf{q}, \Phi), a, \Phi') ,$$

where:

  - $E((\mathsf{q}, \Phi), a, \Phi') = \{(\mathsf{q}, \Phi) \xrightarrow{\mathtt{true}, a, \emptyset} (\mathsf{q}', \Phi') \mid \mathsf{q}' = \Delta(\mathsf{q}, (\Phi, a)) \wedge \mathsf{q}' \notin \mathsf{Q}_F\}$;
  - $E'((\mathsf{q}, \Phi), a, \Phi') = \{(\mathsf{q}, \Phi) \xrightarrow{x<\alpha, a, \emptyset} (\mathsf{q}', \Phi') \mid \mathsf{q}' = \Delta(\mathsf{q}, (\Phi, a)) \wedge \mathsf{q}' \in \mathsf{Q}_F\}$;
  - $E''((\mathsf{q}, \Phi), a, \Phi') = \{(\mathsf{q}, \Phi) \xrightarrow{\alpha\leq x\leq\beta, a, \emptyset} (\overline{\mathsf{q}'}, \Phi') \mid \mathsf{q}' = \Delta(\mathsf{q}, (\Phi, a)) \wedge \mathsf{q}' \in \mathsf{Q}_F\}$.

Definition 4.9 considers the case in which $\alpha > 0$; the case for $\alpha = 0$ is similar, but the set E' of DTA edges is empty. It can be verified that $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ satisfies the conditions of initial determinism and determinism on actions of Definition 2.3. The other two conditions of Definition 2.3 are irrelevant because $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ does not have any boundary edges.

Proposition 4.10 *Let $\mathcal{M}$ be an ASMC, s be a state of $\mathcal{M}$, $\mathcal{N}$ be an NPA and $[\alpha, \beta] \subseteq \mathbb{R}_{\geq 0}$. Then $AccPath^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]}) = AccPath^{\mathcal{M}}(s, \mathcal{A}(\mathcal{N}^{[\alpha,\beta]}))$.*

The proof of the proposition can be found in the appendix. From Proposition 4.10 and the observation that $\mathrm{Pr}_s^{\mathcal{M}}(AccPath^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})) = \mathrm{Pr}_s^{\mathcal{M}}(AccPath^{\mathcal{M}}(s, \mathcal{N}^{(\alpha,\beta]})) = \mathrm{Pr}_s^{\mathcal{M}}(AccPath^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta)})) = \mathrm{Pr}_s^{\mathcal{M}}(AccPath^{\mathcal{M}}(s, \mathcal{N}^{(\alpha,\beta)}))$, the subsequent corollary then follows.

Corollary 4.11 *Let $\mathcal{M}$ be an ASMC, s be a state of $\mathcal{M}$, $\mathcal{N}$ be an NPA, $I \subseteq \mathbb{R}_{\geq 0}$, $\sim \in \{<, \leq, \geq, >\}$ and $\lambda \in [0, 1]$. Then $\mathcal{M}, s \models_{\mathrm{asCSL}} \mathcal{P}_{\sim\lambda}(\mathcal{N}^I)$ if and only if $\mathcal{M}, s \models \mathcal{P}_{\sim\lambda}(\mathcal{A}(\mathcal{N}^I))$.*

4.3 CSL$^{\mathrm{TA}}$ is strictly more expressive than CSL

In this section, we give an example of a CSL$^{\mathrm{TA}}$ formula for which no equivalent CSL formula exists. We note that the DTA of the CSL$^{\mathrm{TA}}$ formula considered does not feature time constraints, and therefore we can obtain an NPA which is equivalent to the DTA in a straightforward manner; hence our result suffices also to show that there exists an asCSL formula for which no equivalent CSL formula exists.

Proposition 4.12 *There is a formula of CSL^{TA} for which there is no equivalent CSL formula.*

The proof of Proposition 4.12 follows a scheme that is different from proofs of similar results on expressiveness of temporal logics for transition systems. We first define the left-hand delimiter $\langle$ for intervals, where $\langle$ denotes either $[$ or $($. Similarly, the right-hand delimiter $\rangle$ denotes either $]$ or $)$. Consider the family of ASMCs $\mathcal{M}[\mu, \mu']$ of Figure 10 (left), for $0 < \mu, \mu' < 1$. Let Φ be a formula of CSL or CSL$^{\mathrm{TA}}$ (for simplicity, we write $\models$ to denote the satisfaction relation of both CSL and CSL$^{\mathrm{TA}}$). Then $[\Phi](s) = \{(\mu, \mu') \in (0, 1)^2 \mid \mathcal{M}[\mu, \mu'], s \models \Phi\}$. For any $0 < \zeta < 1$, let $\Phi^{\zeta} = \mathcal{P}_{\geq\zeta}(\mathcal{A})$, where $\mathcal{A}$ is the DTA depicted in Figure 10 (right). It follows that $[\Phi^{\zeta}](s_0) = \{(\mu, \mu') \in (0, 1)^2 \mid \mu \cdot \mu' \geq \zeta\}$.

The following lemma specifies that, for any CSL formula and any state $s \in \{s_0, s_1, s_2, s_3\}$, the sets of parameters μ, μ' which result in the satisfaction of the CSL formula in state s of $\mathcal{M}[\mu, \mu']$ will be of a particular form. The lemma will then be used as the basis of our expressiveness result: sets of parameters with the form described in the lemma cannot be used to obtain the set $[\Phi^{\zeta}](s_0) = \{(\mu, \mu') \in (0, 1)^2 \mid \mu \cdot \mu' \geq \zeta\}$ of parameters which result in the satisfaction of Φ^{ζ} in s_0, and hence no CSL formula is equivalent to Φ^{ζ}.

Lemma 4.13 *Let Φ be a formula of CSL. Then:*

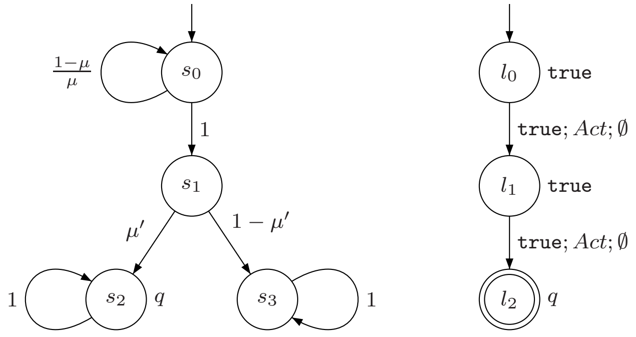

Figure 10. A family of Markov chains $\mathcal{M}[\mu, \mu']$ and a DTA $\mathcal{A}$ of CSL$^{\text{TA}}$

1. *for* $i \in \{2, 3\}$, $[\Phi](s_i)$ *is either* $(0,1)^2$ *or* $\emptyset$*;*
2. $[\Phi](s_1)$ *is a finite union of rectangles of the form* $(0,1) \times \langle \mathsf{a}, \mathsf{b} \rangle$*;*
3. $[\Phi](s_0)$ *is a finite union of (open, closed, or mixed) rectangles of* $(0,1)^2$.

Proof. **Assertion (1).** When starting from s_2 or s_3, the satisfaction of Φ does not depend on μ or μ'. Therefore assertion (1) is satisfied trivially.

Assertion (2). We prove assertion (2) by induction on the size of the formula, taking into account all CSL operators one by one. Let Φ be a formula of CSL. If Φ is an atomic proposition, then $[\Phi](s_1)$ is either $(0,1)^2$ or $\emptyset$. If $\Phi = \neg\Phi'$, then $[\Phi](s_1) = (0,1)^2 \setminus [\Phi'](s_1)$, and thus $[\Phi](s_1)$ is a finite union of rectangles of the form $(0,1) \times \langle \mathsf{a}, \mathsf{b} \rangle$. If $\Phi = \Phi' \wedge \Phi''$, then $[\Phi](s_1) = [\Phi'](s_1) \cap [\Phi''](s_1)$, and thus $[\Phi](s_1)$ is a finite union of rectangles of the form $(0,1) \times \langle \mathsf{a}, \mathsf{b} \rangle$.

If $\Phi = \mathcal{S}_{\geq\lambda}(\Phi')$, then first we observe that the steady-state distribution π of $\mathcal{M}[\mu, \mu']$ starting from s_1 is such that $\pi(s_1, s_2) = \mu'$ and $\pi(s_1, s_3) = 1 - \mu'$. Now we distinguish different cases depending on whether $s_2 \models_{\text{CSL}} \Phi'$ and $s_3 \models_{\text{CSL}} \Phi'$. If both states satisfy Φ', then $[\Phi](s_1) = (0,1)^2$; if neither satisfies Φ' then $[\Phi](s_1) = \emptyset$; if $s_2 \models_{\text{CSL}} \Phi'$ and $s_3 \not\models_{\text{CSL}} \Phi'$ then $[\Phi](s_1) = (0,1) \times [\lambda, 1)$; if $s_2 \not\models_{\text{CSL}} \Phi'$ and $s_3 \models_{\text{CSL}} \Phi'$ then $[\Phi](s_1) = (0,1) \times (0, 1-\lambda]$. The cases of $\Phi = \mathcal{S}_{\sim\lambda}(\Phi')$, with $\sim \in \{\leq, <, >\}$, follow similarly.

If $\Phi = \mathcal{P}_{\geq\lambda}(\mathcal{X}^{[\alpha,\beta]}\Phi')$ we distinguish different cases depending on whether $s_2 \models_{\text{CSL}} \Phi'$ and $s_3 \models_{\text{CSL}} \Phi'$. All the cases are handled similarly, and we only consider that in which $s_2 \models_{\text{CSL}} \Phi'$ and $s_3 \not\models_{\text{CSL}} \Phi'$. Then $[\Phi_1](s_1) = \{(\mu, \mu') \in (0,1)^2 \mid (e^{-\alpha} - e^{-\beta}) \cdot \mu' \geq \lambda\} = (0,1) \times [\frac{\lambda}{e^{-\alpha} - e^{-\beta}}, 1)$, which is of the required form. The cases of $\Phi = \mathcal{P}_{\sim\lambda}(\mathcal{X}^{[\alpha,\beta]}\Phi')$, with $\sim \in \{\leq, <, >\}$, follow similarly.

If $\Phi = \mathcal{P}_{\geq\lambda}(\Phi'\mathcal{U}^{[\alpha,\beta]}\Phi'')$, we make a case analysis with respect to to the rectangles where the satisfaction of Φ' and Φ'' by s_1 is invariant (that is, we consider rectangles of the partition of $(0,1)^2$ induced by the rectangles of $[\Phi'](s_1)$ and $[\Phi''](s_1)$). Our aim is to obtain $[\Phi](s_1)$ by replacing each such rectangle with a set of rectangles in which Φ is satisfied.

Given such a rectangle $\mathcal{R} \subseteq (0,1)^2$ for which $\mathcal{M}[\mu, \mu'], s_1 \models_{\text{CSL}} \Phi''$ for all $(\mu, \mu') \in \mathcal{R}$, then $\mathcal{M}[\mu, \mu'], s_1 \models_{\text{CSL}} \Phi$ for all $(\mu, \mu') \in \mathcal{R}$. Hence $\mathcal{R}$ is included in $[\Phi](s_1)$. Conversely, if $\mathcal{M}[\mu, \mu'], s_1 \models_{\text{CSL}} \neg\Phi' \wedge \neg\Phi''$ for all $(\mu, \mu') \in \mathcal{R}$, then $\mathcal{M}[\mu, \mu'], s_1 \not\models_{\text{CSL}} \Phi$ for all $(\mu, \mu') \in \mathcal{R}$. Hence no rectangle contained in $\mathcal{R}$ is included in $[\Phi](s_1)$.

Now consider a rectangle $\mathcal{R}$ for which, for all $(\mu, \mu') \in \mathcal{R}$, we have $\mathcal{M}[\mu, \mu'], s_1 \models_{\text{CSL}} \Phi' \wedge \neg\Phi''$. Assume that $\mathcal{M}[\mu, \mu'], s_2 \models_{\text{CSL}} \Phi''$ and $\mathcal{M}[\mu, \mu'], s_3 \not\models_{\text{CSL}} \Phi''$ (the other cases are handled similarly). Then we obtain $\{(\mu, \mu') \in \mathcal{R} \mid \mathcal{M}[\mu, \mu'], s_1 \models_{\text{CSL}} \Phi\} = \{(\mu, \mu') \in \mathcal{R} \mid (1 - e^{-\beta}) \cdot \mu' + e^{-\beta} \geq \lambda\} = \{(\mu, \mu') \in \mathcal{R} \mid \mu' \geq \frac{\lambda - e^{-\beta}}{1 - e^{-\beta}}\}$. Thus we include in $[\Phi](s_1)$ the rectangle $\mathcal{R} \cap ((0,1) \times [\frac{\lambda - e^{-\beta}}{1 - e^{-\beta}}, 1))$.

The cases of $\Phi = \mathcal{P}_{\sim\lambda}(\Phi'\mathcal{U}^{[\alpha,\beta]}\Phi'')$, $\sim \in \{\leq, <, >\}$, follow similarly.

Assertion (3). We now prove assertion (3) by induction on the size of the formulae. Let Φ be a formula of CSL. The cases in which Φ is an atomic proposition, $\Phi = \neg\Phi'$, $\Phi = \Phi' \wedge \Phi''$ and $\Phi = \mathcal{S}_{\sim\lambda}(\Phi')$ are proved exactly as for assertion (2) (the steady-state distribution of $\mathcal{M}[\mu, \mu']$ starting from s_0 is the same as that starting from s_1).

If $\Phi = \mathcal{P}_{\geq\lambda}(\mathcal{X}^{[\alpha,\beta]}\Phi')$, we make a case analysis with respect to to the rectangles in which the satisfaction of Φ' by s_0 and s_1 is invariant (that is, we consider rectangles of the partition of $(0,1)^2$ induced by the rectangles of $[\Phi'](s_0)$ and $[\Phi'](s_1)$). As above, we obtain $[\Phi](s_0)$ by replacing each such rectangle with a set of rectangles in which Φ is satisfied. We only consider one such case (the other cases are handled similarly). Consider a rectangle $\mathcal{R} \subseteq (0,1)^2$ for which, for all $(\mu, \mu') \in \mathcal{R}$, we

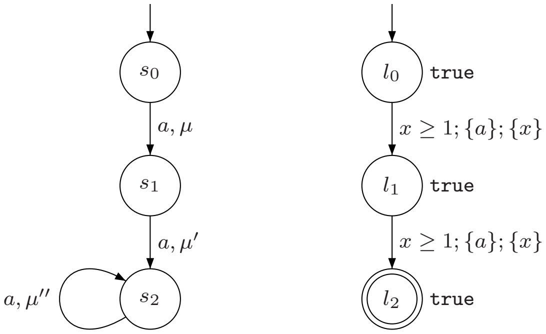

Figure 11. A family of ASMCs $\mathcal{M}[\mu, \mu'\mu'']$ and a DTA $\mathcal{A}$ of CSL$^{\text{TA}}$

have $\mathcal{M}[\mu, \mu'], s_0 \models_{\text{CSL}} \Phi'$ and $\mathcal{M}[\mu, \mu'], s_1 \not\models_{\text{CSL}} \Phi'$. Then $\{(\mu, \mu') \in \mathcal{R} \mid \mathcal{M}[\mu, \mu'], s_0 \models_{\text{CSL}} \Phi\} = \{(\mu, \mu') \in \mathcal{R} | (e^{-\alpha/\mu} - e^{-\beta/\mu}) \cdot \mu \geq \lambda\}$. Let $f(\mu) = (e^{-\alpha/\mu} - e^{-\beta/\mu}) \cdot \mu$. Note that the derivative of f changes its sign only a finite number of times inside $(0, 1)$ (in fact in $\mathbb{R}$). Therefore $(0, 1)$ may be decomposed into a finite number of consecutive intervals where inside an interval f is monotonic. As a consequence $(0, 1)$ may be partitioned into a finite number of consecutive intervals (different from the previous ones) where alternatively f is greater than or equal to λ or strictly smaller than λ. The intervals for which f is greater than or equal to λ induce a finite number of rectangles of the form $\langle \mathsf{a}, \mathsf{b} \rangle \times (0, 1)$, which are included $[\Phi](s_0)$. The cases of $\Phi = \mathcal{P}_{\sim\lambda}(\mathcal{X}^{[\alpha,\beta]}\Phi')$, with $\sim \in \{\leq, <, >\}$, follow similarly.

If $\Phi = \mathcal{P}_{\geq\lambda}(\Phi'\mathcal{U}^{[\alpha,\beta]}\Phi'')$, we make a case analysis with respect to the rectangles where the satisfaction of Φ' and Φ'' by s_0 and by s_1 is invariant (that is, we consider rectangles of the partition of $(0, 1)^2$ induced by the rectangles of $[\Phi'](s_0)$, $[\Phi'](s_1)$, $[\Phi''](s_0)$ and $[\Phi''](s_1)$). Again, we obtain $[\Phi](s_0)$ by replacing each such rectangle with a set of rectangles in which Φ is satisfied.

We handle only one case, noting that the other cases are handled similarly. Consider the rectangle $\mathcal{R} \subseteq (0, 1)^2$ such that, for $i \in \{0, 1\}$, we have $\mathcal{M}[\mu, \mu'], s_i \models_{\text{CSL}} \Phi' \wedge \neg\Phi''$ $\mathcal{M}[\mu, \mu'], s_2 \models_{\text{CSL}} \neg\Phi' \wedge \Phi''$ and $\mathcal{M}[\mu, \mu'], s_3 \models_{\text{CSL}} \neg\Phi' \wedge \neg\Phi''$. The key observation here is that, inside any such rectangle, the loop around s_0 is irrelevant due to the nature of the Until operator. Then we have $\{(\mu, \mu') \in \mathcal{R} \mid \mathcal{M}[\mu, \mu'], s_0 \models_{\text{CSL}} \Phi\} = \{(\mu, \mu') \in \mathcal{R} \mid (e^{-2\alpha}(1 + 2\alpha) - e^{-2\beta}(1 + 2\beta)) \cdot \mu' \geq \lambda\} = \{(\mu, \mu') \in \mathcal{R} \mid \mu' \geq \frac{\lambda}{e^{-2\alpha}(1+2\alpha) - e^{-2\beta}(1+2\beta)}\}$ (the first formula has been obtained by applying an Erlang distribution). Then we include the rectangle $\mathcal{R} \cap ((0, 1) \times [\frac{\lambda}{e^{-2\alpha}(1+2\alpha) - e^{-2\beta}(1+2\beta)}, 1))$ in $[\Phi](s_0)$.

The cases of $\Phi = \mathcal{P}_{\sim\lambda}(\Phi'\mathcal{U}^{[\alpha,\beta]}\Phi'')$, $\sim \in \{\leq, <, >\}$, follow similarly. □

Because $[\Phi^\zeta](s_0) = \{(\mu, \mu') \mid \mu \cdot \mu' \geq \zeta\}$ cannot be expressed as a finite union of rectangles, Lemma 4.13 establishes that Φ^ζ is not equivalent to any formula of CSL. Lemma 4.14 then gives a direct proof of Proposition 4.12.

Lemma 4.14 *For each $0 < \zeta < 1$, the CSL^{TA} formula Φ^ζ is not equivalent to any formula of CSL.*

We also conjecture that there exists a CSL$^{\text{TA}}$ formula for which no equivalent asCSL formula exists. This conjecture is based on the result shown in the next subsection, which shows that there exists a CSL$^{\text{TA}}$ formula which does not use nesting for which there exists no single equivalent asCSL formula which does not use nesting.

4.4 CSL$^{\text{TA}}$ without nesting is strictly more expressive than asCSL without nesting

Consider the ASMC $\mathcal{M}[\mu, \mu, \mu'']$ shown on the left of Figure 11. Given a CSL$^{\text{TA}}$ formula Φ, let $[\Phi] = \{(\mu, \mu', \mu'') \in (0, 1)^3 \mid \mathcal{M}[\mu, \mu', \mu''], s_0 \models \Phi\}$; similarly, given an asCSL formula Φ', let $[\Phi'] = \{(\mu, \mu', \mu'') \in (0, 1)^3 \mid \mathcal{M}[\mu, \mu', \mu''], s_0 \models_{\text{asCSL}} \Phi'\}$. (Note that, in contrast to the case of CSL in Section 4.3, we consider only the state s_0 of $\mathcal{M}$.) The DTA on the right of Figure 11 will be used to define the CSL$^{\text{TA}}$ formula for which there exists no equivalent asCSL formula without nesting.

Given that the action set Act of $\mathcal{M}[\mu, \mu', \mu'']$ is equal to $\{a\}$, the single state proposition of $\mathcal{M}[\mu, \mu', \mu'']$ is $\mathtt{true}$, and that, for the purposes of the following results, we do not allow nesting within asCSL formulae, we have that the input alphabet Ξ of any considered $\mathcal{N}$ is equal to the set $\{(\mathtt{true}, a), (\mathtt{true}, \surd)\}$: for this reason, we write $z \xrightarrow{a} z'$ and $z \xrightarrow{\surd} z'$ for $z \xrightarrow{\mathtt{true}_a} z'$ and $z \xrightarrow{\mathtt{true}_\surd} z'$, respectively.

We identify the set of NPA referring to sequences of a-transitions of at most length 2: let 2NPA refer to the set of NPA for which the longest path from the set of initial states to the set of final states features at most two a-transitions. Let $\mathcal{N} = \langle Z, \Xi, \delta, Z_{Init}, Z_F \rangle$ be an NPA, let $\mathsf{Runs}^{[\mathcal{N}]} = \mathsf{Runs}^{\mathcal{N}}(Z_{Init})$, and let $\mathsf{Runs}^{[\mathcal{N}]}_{\leq 2} = \{r \in \mathsf{Runs}^{[\mathcal{N}]} \mid r$ has at most two a-transitions$\}$ and let $\mathsf{Runs}^{[\mathcal{N}]}_{\geq 3} = \mathsf{Runs}^{[\mathcal{N}]} \setminus \mathsf{Runs}^{[\mathcal{N}]}_{\leq 2}$.

We can obtain an NPA $\mathcal{N}_{\leq 2}$ such that we have $\mathsf{Runs}^{[\mathcal{N}_{\leq 2}]} = \mathsf{Runs}^{[\mathcal{N}]}_{\leq 2}$. Informally, to obtain $\mathcal{N}_{\leq 2}$ we represent each state $z \in Z$ of $\mathcal{N}$ by three copies z^0, z^1 and z^2 in $\mathcal{N}_{\leq 2}$. The transition relation of $\mathcal{N}_{\leq 2}$ is defined such that a transition $z_1 \xrightarrow{\surd} z_2$ of $\mathcal{N}$ is represented by $z_1^i \xrightarrow{\surd} z_2^i$ in $\mathcal{N}_{\leq 2}$, for $i \in \{0, 1, 2\}$. A transition $z_1 \xrightarrow{a} z_2$ of $\mathcal{N}$ is represented in $\mathcal{N}_{\leq 2}$ by the transitions $z_1^i \xrightarrow{a} z_2^{i+1}$ for $i \in \{0, 1\}$, and by $z_1^2 \xrightarrow{a} z_{\mathrm{sink}}$, where z_{sink} is an additional state without outgoing transitions. The set of initial states of $\mathcal{N}_{\leq 2}$ is $\{z^0 \mid z \in Z_{Init}\}$, and the set of final states of $\mathcal{N}_{\leq 2}$ is $\cup_{i \in \{0,1,2\}} \{z^i \mid z \in Z_F\}$. It can be verified that $\mathcal{N}_{\leq 2}$ is a 2NPA.

Similarly, we can obtain an NPA $\mathcal{N}_{\geq 3}$ such that, for any finite path σ of $\mathcal{M}[\mu, \mu', \mu'']$, the set $\mathsf{Runs}^{[\mathcal{N}_{\geq 3}]} = \mathsf{Runs}^{[\mathcal{N}]}_{\geq 3}$. To obtain $\mathcal{N}_{\geq 3}$ we represent each state $z \in Z$ of $\mathcal{N}$ by three copies z^0, z^1 and z^2 in $\mathcal{N}_{\geq 3}$. The transition relation of $\mathcal{N}_{\geq 3}$ is defined such that a transition $z_1 \xrightarrow{\surd} z_2$ of $\mathcal{N}$ is represented by $z_1^i \xrightarrow{\surd} z_2^i$ in $\mathcal{N}_{\geq 3}$, for $i \in \{0, 1, 2\}$. A transition $z_1 \xrightarrow{a} z_2$ of $\mathcal{N}$ is represented by $z_1^i \xrightarrow{a} z_2^{i+1}$ for $i \in \{0, 1\}$ and by $z_1^2 \xrightarrow{a} z_2^2$. The set of initial states of $\mathcal{N}_{\geq 3}$ is $\{z^0 \mid z \in Z_{Init}\}$, and the set of final states of $\mathcal{N}_{\geq 3}$ is $\{z^2 \mid z \in Z_F\}$.

For NPA $\mathcal{N}$ and interval $I \subseteq \mathbb{R}_{\geq 0}$, we write $\Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}^I)$ for $\Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathit{AccPath}^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}^I))$ when clear from the context.

Lemma 4.15 *Let $\langle \alpha, \beta \rangle \subseteq \mathbb{R}_{\geq 0}$ such that $\beta < \infty$, and let $\mu, \mu', \mu'' \in \mathbb{R}_{>0}$. Then for any NPA $\mathcal{N}$, we have $\Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) = \lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}^{\langle \alpha,\beta \rangle})$.*

Proof. First note that $\mathit{AccPath}^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}^{\langle \alpha,\beta \rangle})$ equals:

$$\mathit{AccPath}^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) \cup \mathit{AccPath}^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}_{\geq 3}^{\langle \alpha,\beta \rangle}).$$

Hence:

$$\Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}^{\langle \alpha,\beta \rangle}) \leq \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) + \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}_{\geq 3}^{\langle \alpha,\beta \rangle}) \, .$$

We then observe the following fact:

$$\begin{aligned} \lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) &\leq \lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}^{\langle \alpha,\beta \rangle}) \\ &\leq \lim_{\nu \to 0} (\Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) + \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\geq 3}^{\langle \alpha,\beta \rangle})) \, . \end{aligned}$$

Now note that $\lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\geq 3}^{\langle \alpha,\beta \rangle}) = 0$ (from the fact that, as ν tends to 0, the probability that there exists a finite prefix of a path of $\mathcal{M}[\mu, \mu', \nu]$ comprising at least three a-transitions and with time duration in $\langle \alpha, \beta \rangle$ tends to 0). Hence $\lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) = \lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}^{\langle \alpha,\beta \rangle})$. From the fact that the value of μ'' is irrelevant to the probability of satisfying a property specified by a 2NPA, we have $\lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) = \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle})$ for any $\mu'' \in \mathbb{R}_{>0}$. Hence $\Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}_{\leq 2}^{\langle \alpha,\beta \rangle}) = \lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}^{\langle \alpha,\beta \rangle})$. □

Lemma 4.16 *Let $\langle \alpha, \infty) \subseteq \mathbb{R}_{\geq 0}$, and let $\mu, \mu', \mu'' \in \mathbb{R}_{>0}$. Then for any NPA $\mathcal{N}$, we have either $\lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}^{\langle \alpha,\infty)}) = 1$ or $\Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}_{\leq 2}^{\langle \alpha,\infty)}) = \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\mu'']}(\mathcal{N}^{\langle \alpha,\infty)})$.*

Proof. First note that, if $\mathit{AccPath}^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}_{\geq 3}^{\langle \alpha,\infty)}) \neq \emptyset$, then $\lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\geq 3}^{\langle \alpha,\infty)}) = 1$ from the fact that, as ν tends to 0, the probability that there exists a finite prefix of a path of $\mathcal{M}[\mu, \mu', \nu]$ comprising at least three a-transitions and with time duration in $\langle \alpha, \infty)$ tends to 1. In this case, noting that:

$$\lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}_{\geq 3}^{\langle \alpha,\infty)}) \leq \lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}^{\langle \alpha,\infty)}) \, ,$$

we conclude that $\lim_{\nu \to 0} \Pr_{s_0}^{\mathcal{M}[\mu,\mu',\nu]}(\mathcal{N}^{\langle \alpha,\infty)}) = 1$

If $AccPath^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}^{\langle\alpha,\infty)}_{\geq 3}) = \emptyset$, then:

$$AccPath^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}^{\langle\alpha,\infty)}) = AccPath^{\mathcal{M}[\mu,\mu',\mu'']}(s_0, \mathcal{N}^{\langle\alpha,\infty)}_{\leq 2}) \ .$$

Therefore $\mathrm{Pr}^{\mathcal{M}[\mu,\mu',\mu'']}_{s_0}(\mathcal{N}^{\langle\alpha,\infty)}_{\leq 2}) = \mathrm{Pr}^{\mathcal{M}[\mu,\mu',\mu'']}_{s_0}(\mathcal{N}^{\langle\alpha,\infty)})$. □

Hence, by Lemma 4.15 and Lemma 4.16, and by letting the rate of the self-loop labelling s_2 approach 0, it suffices to consider asCSL formulae which use 2NPA, rather than arbitrary NPA (noting that the case in which $\lim_{\nu\to 0} \mathrm{Pr}^{\mathcal{M}[\mu,\mu',\nu]}_{s_0}(\mathcal{N}^{\langle\alpha,\infty)}) = 1$ is not of interest).

Let $\mathcal{A}$ be the DTA of Figure 11. Then consider the CSL$^{\mathrm{TA}}$ formula $\Phi^{\zeta} = \mathcal{P}_{\geq\zeta}(\mathcal{A})$ for some $\zeta \in (0,1)$. It follows that $[\Phi^{\zeta}] = \{(\mu,\mu',\mu'') \in (0,1)^3 \mid e^{-\mu} \cdot e^{-\mu'} \geq \zeta\}$. Rewriting, we obtain $[\Phi^{\zeta}] = \{(\mu,\mu',\mu'') \in (0,1)^3 \mid \mu + \mu' \leq -\ln(\zeta)\}$.

Lemma 4.17 *Let $\Phi = \mathcal{P}_{\sim\lambda}(\mathcal{N}^{\langle\alpha,\beta\rangle})$ be an asCSL formula for which $\mathcal{N}$ is a 2NPA. Then $[\Phi] = \{(\mu,\mu',\mu'') \in (0,1)^3 \mid \mu \neq \mu' \wedge \Upsilon^{\neq} \sim \lambda\} \cup \{(\mu,\mu',\mu'') \in (0,1)^3 \mid \mu = \mu' \wedge \Upsilon^{=} \sim \lambda\}$, where either:*

- *$\Upsilon^{\neq}$ and $\Upsilon^{=}$ are both 0, or are both 1;*
- *$\Upsilon^{\neq}$ and $\Upsilon^{=}$ are both $e^{-\mu\alpha} - e^{-\mu\beta}$;*
- *$\Upsilon^{\neq}$ is of the form $\frac{\mu'(e^{-\mu\alpha}-e^{-\mu\beta})-\mu(e^{-\mu'\alpha}-e^{-\mu'\beta})}{\mu'-\mu}$, and $\Upsilon^{=}$ is of the form $\mu(e^{-\mu\alpha} - e^{-\mu\beta})$;*
- *$\Upsilon^{\neq}$ is of the form $(e^{-\mu\alpha} - e^{-\mu\beta}) + (\frac{\mu'(e^{-\mu\alpha}-e^{-\mu\beta})-\mu(e^{-\mu'\alpha}-e^{-\mu'\beta})}{\mu'-\mu})(1 - (e^{-\mu\alpha} - e^{-\mu\beta}))$ and $\Upsilon^{=}$ is of the form $(e^{-\mu\alpha} - e^{-\mu\beta}) + (\mu(e^{-\mu\alpha} - e^{-\mu\beta}))(1 - (e^{-\mu\alpha} - e^{-\mu\beta}))$.*

The case in which $\Upsilon^{\neq}$ or $\Upsilon^{=}$ are of the form $e^{-\mu\alpha} - e^{-\mu\beta}$ corresponds to the case in which $\mathcal{N}$ accepts paths which have the duration of the first transition in $\langle\alpha,\beta\rangle$. The third point considers the case in which $\mathcal{N}$ accepts paths which have the duration of the prefix consisting of the first two transitions in $\langle\alpha,\beta\rangle$ (where the expression is obtained by applying an Erlang distribution). The fourth point corresponds to the case in which $\mathcal{N}$ accepts paths which have the duration of the first transition in $\langle\alpha,\beta\rangle$ (first summand) and paths which do not have the duration of the first transition in $\langle\alpha,\beta\rangle$ but have the duration of the prefix consisting of the first two transitions in $\langle\alpha,\beta\rangle$ (second summand).

Given the fact that it is not possible to represent the set $[\Phi^{\zeta}] = \{(\mu,\mu',\mu'') \in (0,1)^3 \mid \mu + \mu' \leq -\ln(\zeta)\}$ using any of the expressions of Lemma 4.17, we have the following proposition.

Proposition 4.18 *There is a formula of CSL^{TA} without nesting for which there is no equivalent asCSL formula of the form $\mathcal{P}_{\sim\lambda}(\mathcal{N}^{I})$ without nesting.*

5 Conclusion

In this paper we have defined a new stochastic temporal logic CSL$^{\mathrm{TA}}$, based on timed automata, which we propose as a good trade-off between adding flexibility to property specification and limiting the explosion of complexity in analysis. With regard to the specification of properties, the most significant extension is the possibility of specifying an arbitrary number of timing constraints along an execution path which may also depend on the history of the process. We have shown that CSL$^{\mathrm{TA}}$ is at least as expressive as both CSL and asCSL. Furthermore, the evaluation process is handled in an uniform way *via* Markov regenerative processes rather than by *ad hoc* transformations as previously. We note that the two restrictions that we have placed on the timed automata used, namely that they are deterministic and have one clock, allow us to obtain a tractable stochastic process for the joint process of the system and the property, namely a Markov regenerative process, for which there exists well-known solution methods [26, 29].

Further work can consider an implementation of the proposed method (possibly exploiting existing Deterministic Stochastic Petri Net tools), and the extension of CSL$^{\mathrm{TA}}$ to allow for rewards [30]. We would also like to investigate the use of CSL$^{\mathrm{TA}}$ for the definition of properties of performance models generated automatically from the sequence diagrams of UML, where the ability of CSL$^{\mathrm{TA}}$ to reason about concatenated time intervals could be of use.

Acknowledgment

The work of S. Haddad was supported in part by the project ANR-06-SETI-002 CheckBound. The work of S. Donatelli and J. Sproston was supported in part by the EEC project 027513 Crutial. The authors would like to thanks the anonymous reviewers for the many useful and detailed suggestions and corrections.

References

[1] C. Smith, *Performance Engineering of Software Systems*. Addison-Wesley, 1990.

[2] *UML Profile for Schedulabibity, Performance and Time Specification*, Object Management Group, January 2005, version 1.1, formal/05-01-02.

[3] *A UML profile for Modeling and Analysis of Real Time Embedded Systems (MARTE)*, OMG, 2007, revised submission.

[4] V. Cortellessa and R. Mirandola, "Deriving a queueing network based performance model from UML diagrams," in *Proc. WOSP'00*. ACM, September 2000, pp. 58–70.

[5] S. Bernardi and J. Merseguer, "Performance evaluation of UML design with stochastic well–formed nets," *Journal of Systems and Software*, vol. 80, no. 11, pp. 1843–1865, November 2007.

[6] C. Canevet, S. Gilmore, J. Hillston, M. Prowse, and P. Stevens, "Performance modelling with UML and stochastic process algebras," *IEE Proceedings: Computers and Digital Techniques*, vol. 150, no. 2, pp. 107–120, March 2003.

[7] S. Balsamo, A. Di Marco, P. Inverardi, and M. Simeoni, "Model-based performance prediction in software development: a survey," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 295–310, May 2004.

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

[9] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Model-checking algorithms for continuous-time Markov chains," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 524–541, 2003.

[10] C. Baier, L. Cloth, B. Haverkort, M. Kuntz, and M. Siegle, "Model checking action- and state-labelled Markov chains," *IEEE Transactions on Software Engineering*, vol. 33, no. 4, pp. 209–224, 2007.

[11] M. Y. Vardi and P. Wolper, "Reasoning about infinite computations," *Information and Computation*, vol. 115, no. 1, pp. 1–37, 1994.

[12] E. M. Clarke, O. Grumberg, and R. P. Kurshan, "A synthesis of two approaches for verifying finite state concurrent systems," *Journal of Logic and Computation*, vol. 2, no. 5, pp. 605–618, 1992.

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

[14] G. Behrmann, A. David, K. G. Larsen, J. Håkansson, P. Pettersson, W. Yi, and M. Hendriks, "UPPAAL 4.0," in *Proc. QEST'06*. IEEE, 2006, pp. 125–126.

[15] R. Alur, C. Courcoubetis, and D. L. Dill, "Model-checking in dense real-time," *Information and Computation*, vol. 104, no. 1, pp. 2–34, 1993.

[16] R. Alur, T. Feder, and T. A. Henzinger, "The benefits of relaxing punctuality," *Journal of the ACM*, vol. 43, no. 1, pp. 116–146, 1996.

[17] E. A. Emerson and J. Y. Halpern, ""Sometimes" and "not never" revisited: On branching versus linear time temporal logic," *Journal of the ACM*, vol. 33, no. 1, pp. 151–178, 1986.

[18] G. Clark and J. Hillston, "Towards automatic derivation of performance measures from PEPA models," in *Proceedings of the UK Performance Engineering Workshop*, 1996.

[19] W. D. Obal II and W. H. Sanders, “State-space support for path-based reward variables,” *Performance Evaluation*, vol. 35, no. 3-4, pp. 233–251, 1999.

[20] A. Bouajjani, Y. Lakhnech, and S. Yovine, “Model-checking for extended timed temporal logics,” in *Proc. FTRTFT’96*, ser. LNCS, vol. 1134. Springer, 1996, pp. 306–325.

[21] F. Laroussinie, N. Markey, and P. Schnoebelen, “Model checking timed automata with one or two clocks,” in *Proc. CONCUR’04*, ser. LNCS, vol. 3170. Springer, 2004, pp. 387–401.

[22] J. Ouaknine and J. Worrell, “On the language inclusion problem for timed automata: Closing a decidability gap,” in *Proc. LICS’04*. IEEE, 2004, pp. 54–63.

[23] S. Donatelli, S. Haddad, and J. Sproston, “CSL$^{\text{TA}}$: an expressive logic for continuous-time Markov chains,” in *Proc. QEST’07*. IEEE, 2007, pp. 31–40.

[24] S. Yovine, “KRONOS: A verification tool for real-time systems,” *Software Tools for Technology Transfer*, vol. 1, no. 1-2, pp. 123–133, 1997.

[25] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 1999.

[26] R. German, *Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets*. Wiley, 2000.

[27] M. Ajmone Marsan and G. Chiola, “On Petri nets with deterministic and exponentially distributed firing times,” in *Proc. ICATPN’86*, ser. LNCS, vol. 266. Springer, 1986, pp. 132–145.

[28] A. Jensen, “Markov chains as an aid in the study of Markov processes,” *Skandinavisk Aktuarietidskrift*, vol. 36, pp. 87–91, 1953.

[29] C. Lindemann, *Performance Modelling with Deterministic and Stochastic Petri Nets*. Wiley, 1998.

[30] B. Haverkort, L. Cloth, H. Hermanns, J.-P. Katoen, and C. Baier, “Model checking performability properties,” in *Proc. DSN’02*. IEEE, 2002, pp. 103–112.

Appendix: proof of Proposition 4.10

Part (1). We first show that $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})$ implies $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{A}(\mathcal{N}^{[\alpha,\beta]}))$. From $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})$, there exists a finite prefix σ_{fin} of σ such that $\tau(\sigma_{\text{fin}}) \in [\alpha, \beta]$ and there exists an accepting run $r \in \mathsf{Runs}(Z_{\mathit{Init}}, \sigma_{\text{fin}})$. We concentrate on the case in which σ_{fin} has at least one transition. Let:

$$\sigma_{\text{fin}} = s^0 \xrightarrow{a^0,\tau^0} s^1 \xrightarrow{a^1,\tau^1} \dots \xrightarrow{a^m,\tau^m} s^{m+1} .$$

Assume that there does not exist a prefix σ' of σ_{fin} such that $\tau(\sigma') \in [\alpha, \beta]$ and that there exists an accepting run $r \in \mathsf{Runs}(Z_{\mathit{Init}}, \sigma')$. Then, by Definition 4.7, we can identify an accepting run in $\mathsf{Runs}(Z_{\mathit{Init}}, \sigma_{\text{fin}})$ of $\mathcal{N}$ which comprises m segments, each of which consists of a number of $\surd$-transitions, followed by a transition with an action label from Act, except possibly in the last segment which can also feature a final fragment comprising $\surd$-transitions. We refer to accepting runs not having a final fragment comprising $\surd$-transitions as runs of form 1, whereas the other runs are referred to as being of form 2.

Formally, a run of **form 1** is as follows:

$$r = (z_0^i \xrightarrow{\Phi_0^i \surd} \dots \xrightarrow{\Phi_{n^i-1}^i \surd} z_{n^i}^i \xrightarrow{\Phi_{n^i}^i a^i} z_0^{i+1})_{i=0,\dots,m}$$

We now show how the run r of $\mathcal{N}$ and the finite path σ_{fin} of $\mathcal{M}$ can be used to obtain a path

$$\sigma_{\mathcal{A}} = (l^0, \bar{x}^0) \xrightarrow{\tau^0,e^0} \dots \xrightarrow{\tau^m,e^m} (l^{m+1}, \bar{x}^{m+1})$$

of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ which can be used, via Definition 2.7, to accept σ. The path $\sigma_{\mathcal{A}}$ will be constructed such that, for each $i \leq m+1$, we have $l^i = (\mathsf{q}^i, \Phi^i)$ where $z_0^i \in \mathsf{q}^i$ and $\Phi^i \Rightarrow \bigwedge_{k<n^i} \Phi_k^i$ such that $s^i \models_{\text{asCSL}} \Phi^i$. We proceed by induction on the length of the path of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$.

The base case is as follows: by Definition 4.9, we have $l^0 = (\mathsf{q}_{\mathit{Init}}, \Phi)$ for some $\Phi \in \Sigma$. By Definition 4.8, we have $\mathsf{q}_{\mathit{Init}} = Z_{\mathit{Init}}$. Because r is accepting, we have that $z_0^0 \in Z_{\mathit{Init}}$, and hence $z_0^0 \in \mathsf{q}_{\mathit{Init}}$. We choose Φ such that both (1) $\Phi \Rightarrow \bigwedge_{k<n^0} \Phi_k^0$ and (2) $s^0 \models_{\text{asCSL}} \Phi$. To satisfy these criteria, we observe the following:

- $s^0 \models_{\mathrm{asCSL}} \Phi^i_k$ for all $k < n^0$ by Definition 4.7, and therefore $s^0 \models_{\mathrm{asCSL}} \bigwedge_{k<n^0} \Phi^i_k$.
- By the definition of Σ, there exists $\Sigma' \subseteq \Sigma$ such that $\bigvee_{\Phi' \in \Sigma'} \Phi' \equiv \bigwedge_{k<n^0} \Phi^i_k$ (because $\mathsf{Runs}(\mathsf{q}_{\mathit{Init}}, (\bigwedge_{k<n^0} \Phi^i_k, a^0)) \neq \emptyset$).

The combination of these two facts allows us to choose $\Phi \in \Sigma$ such that $s^0 \models_{\mathrm{asCSL}} \Phi$ (there will exist at least one such Φ).

Now assume that we have constructed the path

$$(l^0, \bar{x}^0) \xrightarrow{\tau^0, e^0} \dots \xrightarrow{\tau^{i-1}, e^{i-1}} (l^i, \bar{x}^i)$$

for $i \leq m+1$. Our task is to construct $(l^i, \bar{x}^i) \xrightarrow{\tau^i, e^i} (l^{i+1}, \bar{x}^{i+1})$ from the fragment

$$r^i = z^i_0 \xrightarrow{\Phi^i_0 \checkmark} \dots \xrightarrow{\Phi^i_{n^i-1} \checkmark} z^i_{n^i} \xrightarrow{\Phi^i_{n^i} a^i} z^{i+1}_0$$

of r. By definition, we have that $r^i \in \mathsf{Runs}(z^i_0, (\Phi^i, a^i))$, where $\Phi^i \in \Sigma$ is such that (1) $\Phi^i \Rightarrow \bigwedge_{k<n^i} \Phi^i_k$ and (2) $s^i \models_{\mathrm{asCSL}} \Phi^i$ (that such Φ^i exists follows from the reasoning given for Φ^0 above). Therefore by Definition 4.8 and the fact that $z^i_0 \in \mathsf{q}^i$, which holds by induction, we have that $z^{i+1}_0 \in \Delta(\mathsf{q}^i, (\Phi^i, a^i))$. Furthermore, again by Definition 4.8, we have $\mathsf{q}^{i+1} = \Delta(\mathsf{q}^i, (\Phi^i, a^i))$, and hence $z^{i+1}_0 \in \mathsf{q}^{i+1}$. Now we have to ensure the existence of the edge e^i of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ which can be taken after τ^i time units. If $i < m$, then by Definition 4.9, for any $\Phi \in \Sigma$, we have the edge $(\mathsf{q}^i, \Phi^i) \xrightarrow{\mathtt{true}, a^i, \emptyset} (\mathsf{q}^{i+1}, \Phi)$ if $\mathsf{q}^{i+1} \notin \mathsf{Q}_F$, and the edges $(\mathsf{q}^i, \Phi^i) \xrightarrow{x<\alpha, a^i, \emptyset} (\mathsf{q}^{i+1}, \Phi)$ $(\mathsf{q}^i, \Phi^i) \xrightarrow{\alpha \leq x \leq \beta, a^i, \emptyset} (\overline{\mathsf{q}}^{i+1}, \Phi)$ if $\mathsf{q}^{i+1} \in \mathsf{Q}_F$. Let $l^{i+1} = (\mathsf{q}^{i+1}, \Phi^{i+1})$ where $\Phi^{i+1} \in \Sigma$ is such that (1) $\Phi^{i+1} \Rightarrow \bigwedge_{k \leq n^i} \Phi^i_k$ and (2) $s^{i+1} \models_{\mathrm{asCSL}} \Phi^{i+1}$ (again, the existence of Φ^{i+1} follows from the reasoning given for Φ^0). If $i = m$, then we have the edge $(\mathsf{q}^m, \Phi^m) \xrightarrow{\alpha \leq x \leq \beta, a^m, \emptyset} (\overline{\mathsf{q}}^{m+1}, \Phi^{m+1})$, and let $l^{m+1} = (\overline{\mathsf{q}}^{m+1}, \Phi^{m+1})$.

We now verify that the path $\sigma_{\mathcal{A}}$ can be used to accept σ according to Definition 2.7.[2] First, we have that $l^0 \in \mathit{Init}$, $\bar{x}^0 = 0$ and $\overline{\mathsf{q}}^{m+1} \in \mathit{Final}$ (because $z^{m+1}_0 \in \mathsf{q}^{m+1}$ and $z^{m+1}_0 \in Z_F$ implies trivially that $\mathsf{Runs}(\mathsf{q}^{m+1}, (\Phi^{m+1}, \checkmark), Z_F) \neq \emptyset$). Second, for all $i \leq m+1$, observe that $s^i \models_{\mathrm{asCSL}} \Phi^i$. Then, recalling that $\Lambda(l^i) = \Lambda(\mathsf{q}^i, \Phi^i) = \Phi^i$, we have $s^i \models \Lambda(l^i)$.

Third, note that the condition of Definition 2.7 requiring that $a^i \in \mathsf{action}(e^i)$ is satisfied trivially because $\mathsf{action}(e^i) = \{a^i\}$. Fourth, we clearly have that the total time elapsed along σ_{fin} is equal to that elapsed along $\sigma_{\mathcal{A}}$ (because the same time durations τ^i, for all $i \leq m+1$ are used for both paths).

Finally (and significantly), we have to verify that $\sigma_{\mathcal{A}}$ is indeed a path of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$. The arguments concerning the existence of the appropriate edges are given above, and therefore it remains to show that the constraints of the edges are satisfied. For $i < m$, if $\mathsf{q}^{i+1} \notin \mathsf{Q}_F$ we have $\mathsf{guard}(e^i) = \mathtt{true}$, which is trivial. The case in which $\mathsf{q}^{i+1} \in \mathsf{Q}_F$ but $(\mathsf{q}^{i+1}, \Phi^{i+1}) \notin \mathit{Final}$, in which there exist edges with guards $x < \alpha$ and $\alpha \leq x \leq \beta$ can also be dealt with trivially. If $\mathsf{q}^{i+1} \in \mathsf{Q}_F$ and $(\mathsf{q}^{i+1}, \Phi^{i+1}) \in \mathit{Final}$, then e^i must be such that $\mathsf{guard}(e^i) = (x < \alpha)$. For this case, assume that there exists $i < m$ such that $\bar{x}^i + \tau^i \geq \alpha$. First, consider the case in which $\alpha \leq \bar{x}^i + \tau^i \leq \beta$. It then follows that we can find a run r' of $\mathcal{N}$ comprising i fragments such that $\mathit{last}(r') \in Z_F$ (by applying the reasoning given above). Hence the prefix of σ_{fin} up to the $(i+1)$th state is sufficient to show that $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})$, which contradicts the assumption that σ_{fin} is the shortest such prefix. Second, consider the case in which $\bar{x}^i + \tau^i > \beta$. From the fact that $\tau(\sigma_{\mathrm{fin}}) \geq \bar{x}^i + \tau^i > \beta$, clearly $\tau(\sigma_{\mathrm{fin}}) \notin [\alpha, \beta]$, which is a contradiction.

Next we consider the case of edge e^m, for which $\mathsf{guard}(e^m) = (\alpha \leq x \leq \beta)$. We have that $\tau(\sigma_{\mathrm{fin}}) = \sum_{i \leq m} \tau^i$. From the fact that the clock x is not reset on any edge of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$, it follows that $\bar{x}^m = \sum_{i<m} \tau_i$. Hence $\bar{x}^m + \tau^m$ (the value of x when the edge e^m should be taken) is equal to $\sum_{i \leq m} \tau^i$, and therefore equal to $\tau(\sigma_{\mathrm{fin}})$. From $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})$, we know that $\tau(\sigma_{\mathrm{fin}}) \in [\alpha, \beta]$. Hence $\alpha \leq \bar{x}^m + \tau^m \leq \beta$, and consequently $\bar{x}^m + \tau^m \models \mathsf{guard}(e^m)$. Therefore, we conclude that $\sigma_{\mathcal{A}}$ is a path of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$.

Form 2 differs from form 1 in the sense that the final part of the run of $\mathcal{N}$ consists of a sequence of $\checkmark$-transitions. More precisely, form 2 is the following:

$$r = ((z^i_0 \xrightarrow{\Phi^i_0 \checkmark} \dots \xrightarrow{\Phi^i_{n^i-1} \checkmark} z^i_{n^i} \xrightarrow{\Phi^i_{n^i} a^i} z^{i+1}_0)_{i=0,\dots,m}) \xrightarrow{\Phi^{m+1}_0 \checkmark} \dots \xrightarrow{\Phi^m_{n^{m+1}-1} \checkmark} z^{m+2}_0 .$$

[2] We note that κ is the identity function, because there are no boundary edges in $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$.

Most of the details of this case are similar to those of form 1, and therefore we only briefly sketch the differences. Again, the aim is to show how the run r of $\mathcal{N}$ and the finite path σ_{fin} of $\mathcal{M}$ can be used to obtain a path

$$\sigma_{\mathcal{A}} = (l^0, \bar{x}^0) \xrightarrow{\tau^0, e^0} \dots \xrightarrow{\tau^m, e^m} (l^{m+1}, \bar{x}^{m+1})$$

of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ which can be used, via Definition 2.7, to accept σ. The proof proceeds in the same manner as that for form 1, except that, for $i = m$, we consider the final fragment:

$$r^{m+1} = z_0^{m+1} \xrightarrow{\Phi_0^{m+1}\surd} \dots \xrightarrow{\Phi^m_{n^{m+1}-1}\surd} z_0^{m+2} \ .$$

By definition, we have that $r^{m+1} \in \mathsf{Runs}(z_0^m, (\Phi^m, \surd), Z_F)$. Noting also that $z_0^{m+2} \in Z_F$, the remainder of the reasoning follows as for form 1.

Part (2). We now show that $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{A}(\mathcal{N}^{[\alpha,\beta]}))$ implies $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})$. From $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{A}(\mathcal{N}^{[\alpha,\beta]}))$, there exists a path $\sigma_{\mathcal{A}} = (l^0, \bar{x}^0) \xrightarrow{\tau^0, e^0} \dots \xrightarrow{\tau^m, e^m} (l^{m+1}, \bar{x}^{m+1})$ of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ satisfying the conditions of Definition 2.7. We show how $\sigma_{\mathcal{A}}$ can be used to identify a finite prefix

$$\sigma_{\text{fin}} = s^0 \xrightarrow{a^0, \tau^0} s^1 \xrightarrow{a^1, \tau^1} \dots \xrightarrow{a^m, \tau^m} s^{m+1}$$

of σ such that $\tau(\sigma_{\text{fin}}) \in [\alpha, \beta]$ and there exists an accepting run $r \in \mathsf{Runs}(Z_{\mathit{Init}}, \sigma_{\text{fin}})$ of $\mathcal{N}$, which suffices for showing that $\sigma \in \mathit{AccPath}^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})$. The idea is to iterate backwards along the path $\sigma_{\mathcal{A}}$ from $(l^{m+1}, \bar{x}^{m+1})$ to $(l^0, \bar{x}^0)$.

First we consider $(l^{m+1}, \bar{x}^{m+1})$: by Definition 2.7, we have $l^{m+1} \in \mathit{Final}$. We choose $\bar{x}^{m+1} = \tau(\sigma_{\text{fin}})$.

Now consider the final transition $(l^m, \bar{x}^m) \xrightarrow{\tau^m, e^m} (l^{m+1}, \bar{x}^{m+1})$ of $\sigma_{\mathcal{A}}$. We construct the path fragment r^m, which is either of form 1 or of form 2 of part (1) of this proof. Note that e^m is of the form $(\mathsf{q}^m, \Phi^m) \xrightarrow{\alpha \le x \le \beta, a^m, \emptyset} (\overline{\mathsf{q}}^{m+1}, \Phi^{m+1})$. Hence, by Definition 4.9, we have that $\Delta(\mathsf{q}^m, (\Phi^m, a^m)) \in \mathsf{Q}_F$. Furthermore, by Definition 4.8, for all $z \in \Delta(\mathsf{q}^m, (\Phi^m, a^m))$, it follows that there exists $r' \in \mathsf{Runs}(\mathsf{q}^m, (\Phi^m, a^m))$ such that $\mathit{last}(r') = z$. Hence there exists $r' \in \mathsf{Runs}(\mathsf{q}^m, (\Phi^m, a^m))$ such that $\mathit{last}(r') = z_0^{m+1}$. Let $r^m = r'$. Note that the first state of r^m, namely z_0^m, is such that $z_0^m \in \mathsf{q}^m$.

Now let $0 < i \le m$, and assume that we have constructed the path $(r^j)_{j=i,\dots,m}$ of $\mathcal{N}$. We show how to construct the path fragment r^{i-1}. Consider the transition $(l^{i-1}, \bar{x}^{i-1}) \xrightarrow{\tau^{i-1}, e^{i-1}} (l^i, \bar{x}^i)$ of $\sigma_{\mathcal{A}}$. Writing $l^i = (\mathsf{q}^i, \Phi^i)$ and z_0^i as the first state of r^i, we have that $z_0^i \in \mathsf{q}^i$ by induction. By Definition 4.9 the edge e^{i-1} of the DTA transition will be of the form $e^{i-1} = (\mathsf{q}^{i-1}, \Phi^{i-1}) \xrightarrow{\mathtt{true}, a^{i-1}, \emptyset} (\mathsf{q}^i, \Phi^i)$ or $e^{i-1} = (\mathsf{q}^{i-1}, \Phi^{i-1}) \xrightarrow{x < \alpha, a^{i-1}, \emptyset} (\mathsf{q}^i, \Phi^i)$, where $\mathsf{q}^{i-1} \in \mathsf{Q}$ is such that $\mathsf{q}^i = \Delta(\mathsf{q}^{i-1}, (\Phi^{i-1}, a^{i-1}))$. By Definition 4.8, for all $z \in \mathsf{q}^i$, it follows that there exists $r' \in \mathsf{Runs}(\mathsf{q}^{i-1}, (\Phi^{i-1}, a^{i-1}))$ such that $\mathit{last}(r') = z$. Hence there exists $r \in \mathsf{Runs}(\mathsf{q}^{i-1}, (\Phi^{i-1}, a^{i-1}))$ such that $\mathit{last}(r') = z_0^i$. Let $r^{i-1} = r'$. The first state z_0^{i-1} of r^{i-1} is such that $z_0^{i-1} \in \mathsf{q}^{i-1}$.

We now show that the path $r = (r^i)_{i=0,\dots,m}$ of $\mathcal{N}$ is in the set $\mathsf{Runs}(Z_{\mathit{Init}}, \sigma_{\text{fin}})$. Assuming that $|\sigma_{\text{fin}}| > 0$, we proceed by induction on the length of path suffixes of σ_{fin}.

Consider r^m. Assume that the last transition r^m has an action label from Act (that is, r is of form 1 according to part (1) of the proof). We write:

$$r^m = z_0^m \xrightarrow{\Phi_0^m \surd} \dots \xrightarrow{\Phi^m_{n^m-1}\surd} z^m_{n^m} \xrightarrow{\Phi^m_{n^m} a^m} z_0^{m+1} \ .$$

First we consider the path of length 0 comprising s^{m+1}. Clearly, by Definition 4.7, we have that $z_0^{m+1} \in \mathsf{Runs}(z_0^{m+1}, s^{m+1})$. Next consider the transition $z^m_{n^m} \xrightarrow{\Phi^m_{n^m} a^m} z_0^{m+1}$. By Definition 2.7, we have $s^m \models \Lambda(l^m)$. Because $\Lambda(l^m) = \Phi^m \Rightarrow \bigwedge_{j \le n^m} \Phi^m_j$, it follows that $s^m \models_{\text{asCSL}} \Phi^m_j$ for all $j \le n^m$. By Definition 4.7, by $s^m \models_{\text{asCSL}} \Phi^m_{n^m}$, and using the fact that the final transition of σ_{fin} is $s^m \xrightarrow{a^m, \tau^m} s^{m+1}$ and $z_0^{m+1} \in \mathsf{Runs}(z_0^{m+1}, s^{m+1})$, we conclude that:

$$z^m_{n^m} \xrightarrow{\Phi^m_{n^m} a^m} z_0^{m+1} \in \mathsf{Runs}(z^m_{n^m}, s^m \xrightarrow{a^m, \tau^m} s^{m+1}) \ .$$

Given that, for each $j < n^m$, the transition $z^m_j \xrightarrow{\Phi^m_j \surd} z^m_{j+1}$ features $\surd$, we can conclude from Definition 4.7 that $\mathsf{Runs}(z^m_j, s^m \xrightarrow{a^m, \tau^m} s^{m+1})$ contains the path:

$$z^m_j \xrightarrow{\Phi^m_j \surd} \dots \xrightarrow{\Phi^m_{n^m-1}\surd} z^m_{n^m} \xrightarrow{\Phi^m_{n^m} a^m} z_0^{m+1} \ .$$

Now assume that the last transition of r^m is a $\surd$-transition (that is, r is of form 2 according to part (1) of the proof); we write $r^m = r_1^m r_2^m$ where:

$$\begin{array}{rcl} r_1^m & = & z_0^m \xrightarrow{\Phi_0^m \surd} \dots \xrightarrow{\Phi_{n^m-1}^m \surd} z_{n^m}^m \xrightarrow{\Phi_{n^m}^m a^m} z_0^{m+1} \\ r_2^m & = & z_0^{m+1} \xrightarrow{\Phi_0^{m+1} \surd} \dots \xrightarrow{\Phi_{n^{m+1}-1}^m \surd} z_0^{m+2} . \end{array}$$

For each $j \leq n^{m+1}$, by Definition 4.7 we can show that:

$$z_j^{m+1} \xrightarrow{\Phi_j^{m+1} \surd} \dots \xrightarrow{\Phi_{n^{m+1}-1}^m \surd} z_0^{m+2} \in \mathsf{Runs}(z_j^{m+1}, s^{m+1}) .$$

The proof for the remaining suffixes of r^m proceeds as for the previous case. Note that $s^m \models \Lambda(l^m)$ where $\Lambda(l^m) = \Phi^m \Rightarrow \bigwedge_{j \leq n^m} \Phi_j^m$, and hence $s^m \models_{\mathrm{asCSL}} \Phi_j^m$ for all $j \leq n^m$. Furthermore, we have $\Lambda(l^{m+1}) = \Phi^m \Rightarrow \bigwedge_{j \leq n^{m+1}} \Phi_j^{m+1}$, and hence $s^{m+1} \models_{\mathrm{asCSL}} \Phi_j^{m+1}$ for all $j \leq n^{m+1}$. Hence, for both cases, we have shown that $r^m \in \mathsf{Runs}(z_0^m, s^m \xrightarrow{a^m, \tau^m} s^{m+1})$.

Let $0 < i \leq m$, and assume that we have shown that $(r^j)_{j=i,\dots,m} \in \mathsf{Runs}(z_0^i, s^i \xrightarrow{a^i, \tau^i} \dots \xrightarrow{a^m, \tau^m} s^{m+1})$, where z_0^i is the first state of r^i. The task of showing that $(r^j)_{j=i-1,\dots,m} \in \mathsf{Runs}(z_0^{i-1}, s^{i-1} \xrightarrow{a^{i-1}, \tau^{i-1}} \dots \xrightarrow{a^m, \tau^m} s^{m+1})$ is similar to the case for r_m in the case that r is of form 1. First, from $s^{i-1} \models \Lambda(l^{i-1})$, we have that $s^m \models_{\mathrm{asCSL}} \Phi_j^{i-1}$ for all $j \leq n^{i-1}$. Using the fact that $(r^j)_{j=i,\dots,m} \in \mathsf{Runs}(z_0^i, s^i \xrightarrow{a^i, \tau^i} \dots \xrightarrow{a^m, \tau^m} s^{m+1})$, we have that $z_{n^{i-1}}^{i-1} \xrightarrow{\Phi_{n^{i-1}}^{i-1} a^{i-1}} z_0^i$ is contained in

$$\mathsf{Runs}(z_{n^{i-1}}^{i-1}, s^{i-1} \xrightarrow{a^{i-1}, \tau^{i-1}} \dots \xrightarrow{a^m, \tau^m} s^{m+1}) .$$

Then we have that, for each $j < n^{i-1}$,

$$z_j^{i-1} \xrightarrow{\Phi_j^{i-1} \surd} \dots \xrightarrow{\Phi_{n^{i-1}-1}^{i-1} \surd} z_{n^{i-1}}^{i-1} \xrightarrow{\Phi_{n^{i-1}}^{i-1} a^{i-1}} z_0^m$$

is contained in

$$\mathsf{Runs}(z_j^{i-1}, s^{i-1} \xrightarrow{a^{i-1}, \tau^{i-1}} \dots s^m \xrightarrow{a^m, \tau^m} s^{m+1}) .$$

Hence $r = (r^i)_{i=0,\dots,m}$ of $\mathcal{N}$ is in the set $\mathsf{Runs}(Z_{Init}, \sigma_{\mathrm{fin}})$. It remains to show that $\tau(\sigma_{\mathrm{fin}}) \in [\alpha, \beta]$. Recall that $\bar{x}^{m+1} = \tau(\sigma_{\mathrm{fin}})$. Given that $\sigma_{\mathcal{A}}$ is a path of $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ for which $\mathsf{guard}(e^m) = (\alpha \leq x \leq \beta)$, and that $\mathcal{A}(\mathcal{N}^{[\alpha,\beta]})$ never resets the clock x, we have that $\bar{x}^{m+1} = \bar{x}^m + \tau^m \in [\alpha, \beta]$. Hence $\tau(\sigma_{\mathrm{fin}}) \in [\alpha, \beta]$. We conclude that $\sigma \in AccPath^{\mathcal{M}}(s, \mathcal{N}^{[\alpha,\beta]})$ as required. □