Logical Methods in Computer Science
Vol. 8 (2:17) 2012, pp. 1–18
www.lmcs-online.org

Submitted Jun. 16, 2011
Published Jun. 28, 2012

EFFICIENT CSL MODEL CHECKING USING STRATIFICATION [*]

LIJUN ZHANG [a], DAVID N. JANSEN [b], FLEMMING NIELSON [c], AND HOLGER HERMANNS [d]

[a,c] Technical University of Denmark, DTU Informatics, Denmark
e-mail address: {zhang,nielson}@imm.dtu.dk

[b] Radboud Universiteit, Model-based System Design, Nijmegen, The Netherlands
e-mail address: dnjansen@cs.ru.nl

[d] Saarland University, Computer Science, Saarbrücken, Germany
e-mail address: hermanns@cs.uni-saarland.de

Abstract. For continuous-time Markov chains, the model-checking problem with respect to continuous-time stochastic logic (CSL) has been introduced and shown to be decidable by Aziz, Sanwal, Singhal and Brayton in 1996 [1, 2]. Their proof can be turned into an approximation algorithm with worse than exponential complexity. In 2000, Baier, Haverkort, Hermanns and Katoen [4, 5] presented an efficient polynomial-time approximation algorithm for the sublogic in which only binary until is allowed. In this paper, we propose such an efficient polynomial-time approximation algorithm for full CSL.

The key to our method is the notion of *stratified CTMCs* with respect to the CSL property to be checked. On a stratified CTMC, the probability to satisfy a CSL path formula can be approximated by a transient analysis in polynomial time (using uniformization). We present a measure-preserving, linear-time and -space transformation of any CTMC into an equivalent, stratified one. This makes the present work the centerpiece of a broadly applicable full CSL model checker.

Recently, the decision algorithm by Aziz *et al.* was shown to work only for stratified CTMCs. As an additional contribution, our measure-preserving transformation can be used to ensure the decidability for general CTMCs.

1. Introduction

Continuous-time Markov chains (CTMC) play an important role in performance evaluation of networked, distributed, and biological systems. The concept of formal verification for CTMCs was introduced by Aziz, Sanwal, Singhal and Brayton in 1996 [1, 2]. Their seminal paper defined continuous-time stochastic logic (CSL) to specify properties over CTMCs. It showed that the model checking problem for CTMCs, which asks whether the CTMC satisfies a given CSL property, is decidable, using algebraic and transcendental number theory. Their proof is constructive, so it can be turned into an approximation procedure

1998 ACM Subject Classification: G.3, F.4.1, F.3.1.

Key words and phrases: continuous-time Markov chains, continuous stochastic logic, model checking, approximation algorithm, stratification.

[*] A preliminary version of the paper has appeared in [22].

for the relevant probabilities. However, its complexity may be worse than exponential in the size of the formula.

The characteristic construct of CSL is a probabilistic formula of the form $\mathcal{P}_{<p}(\varphi)$, where $p \in [0, 1]$. Here φ is a path formula; more concretely, it is a *multiple until* formula $f_1 \; U_{I_1} \; f_2 \; U_{I_2} \; \dots \; U_{I_{k-1}} \; f_k$ where $k \geq 2$. The formula $\mathcal{P}_{<p}(\varphi)$ expresses a constraint on the probability to reach an f_k-state by passing only through (zero or more) f_1-, f_2-, …, f_{k-1}-states in the given order (together with a timing constraint indicated by the intervals $I_1, \dots, I_{k-1}$). The key to solve the model checking problem is to approximate this probability $\mathrm{Pr}_s(\varphi)$ closely enough to decide whether it is $< p$. The decision procedure in [2] first decomposes the formula into (up to) $(k-1)^{k-1}$ many subformulas with suitable timing constraints. For each subformula, it then exploits properties of algebraic and transcendental numbers, but the corresponding algorithm is unfortunately impractical. In 2000, Baier *et al.* [4, 5] presented an *approximate model checking algorithm* for the case $k = 2$. This algorithm is based on transient probability analysis for CTMCs. More precisely, it was shown that $\mathrm{Pr}_s(\varphi)$ can be approximated, up to an a priori given precision ε, by a sum of transient probabilities in the CTMCs. Their algorithm then led to further development of approximation algorithms for infinite CTMCs [11, 12] and abstraction techniques [15]. More importantly, several tools support approximate model checking, including PRISM [17] and MRMC [16].

Effective model checking of full CSL with multiple until formulas ($k > 2$) is an open problem. This problem is gaining importance e. g. in the field of system biology, where one is interested in oscillatory behavior of CTMCs [6, 19]. More precisely, if one intends to quantify the probability mass oscillating between high, medium and low concentrations (or numbers) of some species, a formula like $\mathcal{P}_{>0.2}(\mathit{high} \; U_{I_1} \; \mathit{medium} \; U_{I_2} \; \mathit{low} \; U_{I_3} \; \mathit{medium} \; U_{I_4} \; \mathit{high})$ is needed, but this is not at hand with the current state of the art. In CTL, multiple until formulas like $\forall(\mathit{high} \; U \; \mathit{medium} \; U \; \mathit{low} \; U \; \mathit{medium} \; U \; \mathit{high})$ do not increase expressivity because they are equivalent to something like $\forall(\mathit{high} \; U \; \forall(\mathit{medium} \; U \; \forall(\dots U \; \mathit{high})))$.

In this paper we propose an approximate algorithm for checking CSL with multiple until formulas. We introduce a subclass of *stratified CTMCs,* on which the approximation of $\mathrm{Pr}_s(\varphi)$ can be obtained by efficient transient analysis. Briefly, a CTMC is stratified with respect to $\varphi = f_1 \; U_{I_1} \; f_2 \; U_{I_2} \dots f_k$, if the transitions of the CTMC respect the order given by the f_i. This specific order makes it possible to express $\mathrm{Pr}_s(\varphi)$ recursively: more precisely, it is the product of a transient vector and $\mathrm{Pr}_{s'}(\varphi')$, where φ' is a kind of suffix subformula of φ. Stratified CTMCs are the key element for our analysis: in a stratified CTMC, the problem reduces to a transient analysis, for which efficient implementations using *uniformization* [10] exist. Thus, we extend the well-known result [5] for the case of binary until to multiple until formulas.

For a general CTMC, we present a measure-preserving transformation to a stratified CTMC. Our reduction is described using a *deterministic finite automaton* (DFA) over the alphabet $2^{\{f_1,\dots,f_k\}}$. The DFA accepts the finite word $w = w_1 w_2 \dots w_n$ if and only if the corresponding set of time-abstract paths in the CTMC contributes to $\mathrm{Pr}_s(\varphi)$, i. e., it respects the order of the f_i. The transformation does not require to construct the full DFA, but only the product of the CTMC and the DFA. We show that the product is a stratified CTMC, and moreover, the measure $\mathrm{Pr}_s(\varphi)$ is preserved. This product can be constructed in linear time and space in the size of the CTMC and k. Thus our method will be useful as the centerpiece of a full CSL model checker equipped with multiple until formulas.

Recently, the decision algorithm by Aziz *et al.* was shown to produce erroneous results on some non-stratified CTMCs [13]. Still, their algorithm is correct on stratified CTMCs. As an additional contribution, our measure-preservation theorem ensures the decidability of CSL model checking for general CTMCs.

Overview of the article. Section 2 sets the ground for the paper. In Section 3 we introduce *stratified CTMCs* formally. The first main result is shown in Section 4: it constructs a DFA for an until formula, and then shows that the product is a stratified CTMC and the relevant measures are preserved. Section 5 discusses the computations in the product CTMC. A model checking algorithm is presented in Section 6. Section 7 discusses related work, and the paper is concluded in Section 8.

2. Preliminaries

This section presents the definition of Markov chains, probability space, transient and steady-state distributions. For details please refer to [20, 18, 5].

2.1. **Markov Chains.**

Definition 2.1. A labeled *discrete-time Markov chain* (DTMC) is a tuple $\mathcal{D} = (S, \mathbf{P}, L)$, where S is a finite set of states, $\mathbf{P} : S \times S \to [0,1]$ is a probability matrix satisfying $\sum_{s' \in S} \mathbf{P}(s, s') \in \{0, 1\}$ for all $s \in S$, and $L : S \to 2^{AP}$ is a labeling function.

A labeled *continuous-time Markov chain* (CTMC) is a tuple $\mathcal{C} = (S, \mathbf{R}, L)$, where S and L are defined as for DTMCs, and $\mathbf{R} : S \times S \to \mathbb{R}_{\geq 0}$ is a rate matrix.

For $A \subseteq S$, define $\mathbf{R}(s, A) := \sum_{s' \in A} \mathbf{R}(s, s')$, and let $E(s) := \mathbf{R}(s, S)$ denote the *exit rate* of s. A state s is called *absorbing* if $E(s) = 0$. If $\mathbf{R}(s, s') > 0$, we say that there is a transition from s to s'.

The transition probabilities in a CTMC are exponentially distributed over time. If s is the current state of the CTMC, the probability that some transition will be triggered within time t is $1 - e^{-E(s)t}$. Furthermore, if $\mathbf{R}(s, s') > 0$ for more than one state s', the probability to take a particular transition to s' is $\frac{\mathbf{R}(s,s')}{E(s)} \cdot \left(1 - e^{-E(s)t}\right)$. The labeling function L assigns to each state s the set of atomic propositions $L(s) \subseteq AP$ which are valid in s.

A CTMC $\mathcal{C}$ (and also a DTMC) is usually equipped with an initial state $s_{\text{init}} \in S$ or, more generally, an initial distribution $\alpha_{\text{init}} : S \to [0,1]$ satisfying $\sum_{s \in S} \alpha_{\text{init}}(s) = 1$.

Paths and probabilistic measures. A (sample) path is a right-continuous function $\sigma : \mathbb{R}_{\geq 0} \to S$ (with the discrete topology on S). Then, $\sigma(t)$ denotes the state occupied at time t.

For $i \in \mathbb{N}$, let $\sigma_S[i] = s_i$ denote the $(i+1)$-th state visited, and $\sigma_T[i] = t_i$ denote the time spent in $\sigma_S[i]$. For finite paths, $\sigma_T[n]$ is defined to be ∞ if $\sigma_S[n]$ is the last (absorbing) state. Let $Path^{\mathcal{C}}$ denote the set of all (finite and infinite) paths, and $Path^{\mathcal{C}}(s)$ denote the subset of those paths starting from s.

We sometimes use a different notation to describe a path, namely a finite sequence $\sigma = s_0 t_0 s_1 t_1 \ldots s_n$ (meaning that $\sigma_S[i] = s_i$ and $\sigma_T[i] = t_i$ for all $i < n$, and $\sigma_S[n] = s_n$ is an absorbing state), or an infinite sequence $\sigma = s_0 t_0 s_1 t_1 \ldots$ if no absorbing state is hit. The relation between the two notations is: $\sigma(t) = s_i$ where i is the smallest index with $t < \sum_{j=0}^{i} t_j$ (as remarked by [18, p. 170], we have to use a strict inequality here for technical reasons, not the non-strict inequality as in [5].).

Let $s_0, s_1, \ldots, s_k$ be states in S with $\mathbf{R}(s_i, s_{i+1}) > 0$ for all $0 \le i < k$. Let $I_0, I_1, \ldots, I_{k-1}$ be nonempty intervals in $\mathbb{R}_{\ge 0}$. The *cylinder set* $\mathit{Cyl}(s_0, I_0, \ldots, s_{k-1}, I_{k-1}, s_k)$ is defined by:

$$\mathit{Cyl}(s_0, I_0, \ldots, s_{k-1}, I_{k-1}, s_k) := \{\sigma \in \mathit{Path}^{\mathcal{C}} \mid \forall 0 \le i \le k.\ \sigma_S[i] = s_i \wedge \forall 0 \le i < k.\ \sigma_T[i] \in I_i\}.$$

Let $\mathcal{F}(\mathit{Path}^{\mathcal{C}})$ denote the smallest σ-algebra on $\mathit{Path}^{\mathcal{C}}$ containing all cylinder sets. For initial distribution $\alpha : S \to [0, 1]$, a probability measure (denoted $\Pr_\alpha^{\mathcal{C}}$) on this σ-algebra is introduced as follows: $\Pr_\alpha^{\mathcal{C}}$ is the unique measure that satisfies: $\Pr_\alpha^{\mathcal{C}}(\mathit{Cyl}(s))$ equals $\alpha(s)$, and for $k > 0$,

$$\Pr_\alpha^{\mathcal{C}}(\mathit{Cyl}(s_0, I_0, \ldots, I_{k-1}, s_k)) = \Pr_\alpha^{\mathcal{C}}(\mathit{Cyl}(s_0, I_0, \ldots, I_{k-2}, s_{k-1})) \cdot \frac{\mathbf{R}(s_{k-1}, s_k)}{E(s_{k-1})} \cdot \eta(I_{k-1})$$

where $\eta(I_{k-1}) := \exp(-E(s_{k-1}) \inf I_{k-1}) - \exp(-E(s_{k-1}) \sup I_{k-1})$ is the probability to take a transition during time interval I_{k-1}. (As a consequence, the probability of a cylinder set containing a point interval $[t, t]$ is 0.) If $\alpha(s) = 1$ for some state $s \in S$, we sometimes simply write $\Pr_s^{\mathcal{C}}$ instead of $\Pr_\alpha^{\mathcal{C}}$. We omit the superscript $\mathcal{C}$ if it is clear from the context.

Transient and steady-state probability. Starting with distribution α, the transient probability vector at time t, denoted by $\pi(\alpha, t)$, is the probability distribution over states at time t. If $t = 0$, we have $\pi(\alpha, 0)(s') = \alpha(s')$. For $t > 0$, the transient probability is given by: $\pi(\alpha, t) = \pi(\alpha, 0)e^{\mathbf{Q}t}$ where $\mathbf{Q} := \mathbf{R} - \mathit{Diag}(E)$ is the infinitesimal generator matrix. $\mathit{Diag}(E)$ denotes the diagonal matrix with $\mathit{Diag}(E)(s, s) = E(s)$. The steady-state distribution is defined as the limit $\lim_{t\to\infty} \pi(\alpha, t)$, which always exists for finite CTMCs.

2.2. **Deterministic Finite Automata.**

Definition 2.2. A *deterministic finite automaton* is a tuple $\mathcal{B} = (\Sigma, Q, q_{in}, \delta, F)$, where Σ is a finite alphabet, Q is a finite set of states, $q_{in} \in Q$ is an initial state, $\delta : Q \times \Sigma \nrightarrow Q$ is a partial transition function, and $F \subseteq Q$ is a set of final states.

We call a finite sequence $w = w_1 w_2 \ldots w_n$ over Σ a *word* over Σ. w induces at most one path $\sigma(w) = q_0 q_1 \ldots q_n$ in $\mathcal{B}$ where $q_0 = q_{in}$ and $q_i = \delta(q_{i-1}, w_i)$ for $i = 1, \ldots, n$. This word w, and also the corresponding path $\sigma(w)$, is *accepting* if $\sigma(w)$ exists and $q_n \in F$.

2.3. **Continuous Stochastic Logic (CSL).**

We consider the branching-time temporal logic *Continuous Stochastic Logic* (CSL) introduced by Aziz *et al.* [2], which allows us to specify properties over CTMCs. Its syntax is defined as follows:

$$\Phi := a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathcal{P}_{\trianglelefteq p}(\varphi)$$
$$\varphi := \Phi_1\ U_{I_1}\ \Phi_2\ U_{I_2}\ \ldots\ U_{I_{k-1}}\ \Phi_k$$

where $a \in AP$ is an atomic proposition, $I_1, I_2, \ldots \subseteq \mathbb{R}_{\ge 0}$ are nonempty left-closed intervals with rational bounds, $\trianglelefteq \in \{<, \le, \ge, >\}$, $p \in \mathbb{Q} \cap [0, 1]$, and $k \ge 2$. We use the abbreviation $\Diamond_I \Phi = (\neg(a \wedge \neg a))\ U_I\ \Phi$, for an arbitrary atomic proposition a. The syntax of CSL consists of state formulas and path formulas: we use $\Phi, \Phi_1, \Psi, \Psi_1, \ldots$ for state formulas and $\varphi, \varphi_1, \psi, \psi_1, \ldots$ for path formulas.

Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a CTMC with $s \in S$. The semantics of most CSL state formulas is standard: $s \models a$ iff $a \in L(s)$; $s \models \neg\Phi$ iff $s \not\models \Phi$; $s \models \Phi \wedge \Psi$ iff $s \models \Phi$ and $s \models \Psi$. For probabilistic formulas, we have:

$$s \models \mathcal{P}_{\trianglelefteq p}(\varphi) \text{ iff } \Pr_s\{\sigma \in \mathit{Path} \mid \sigma \models \varphi\} \trianglelefteq p$$

where $\Pr_s\{\sigma \in \mathit{Path} \mid \sigma \models \varphi\}$, or $\Pr_s(\varphi)$ for short, denotes the probability measure of the set of all paths which start with s and satisfy φ.

The satisfaction relation for CSL path formulas is defined as follows: let σ be a path, and let $\varphi = \Phi_1 \; U_{I_1} \; \Phi_2 \; U_{I_2} \; \ldots \Phi_k$ be a path formula. Then $\sigma \models \varphi$ if and only if there exist real numbers $0 \leq t_1 \leq t_2 \leq \ldots \leq t_{k-1}$ such that $\sigma(t_{k-1}) \models \Phi_k$, and for each integer $0 < i < k$ we have $(t_i \in I_i) \wedge (\forall t' \in [t_{i-1}, t_i))(\sigma(t') \models \Phi_i)$, where t_0 is defined to be 0 for notational convenience.

For a CSL path formula $\varphi = \Phi_1 \; U_{[a_1,b_1)} \; \Phi_2 \; U_{[a_2,b_2)} \; \Phi_3$ with $a_2 < a_1$, one can replace the second interval by $[a_1, b_2)$ without changing the set of paths that satisfy the formula. Thus, we shall assume that the left endpoints – and similarly, the right endpoints – of the intervals in multiple until formulas are always nondecreasing.

3. Stratified CTMCs

The main challenge of model checking is the computation and the approximation of the probability $\Pr_s(\varphi)$. We now introduce the class of *stratified* CTMCs. This is the key for the computation of $\Pr_s(\varphi)$. For now, the path formula φ contains pairwise different atomic propositions as subformulas. In Section 6.1, we shall see that this definition is easily generalized to formulas containing more complex subformulas.

Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a CTMC. Let $\varphi = f_1 \; U_{I_1} \; f_2 \; U_{I_2} \; \ldots f_k$ be a CSL path formula with pairwise different atomic propositions. Moreover, we let $F := \{f_1, f_2, \ldots, f_k\}$, and $\sqsubseteq$ be an order on F such that $f_i \sqsubseteq f_j$ iff $i \leq j$. For a state s, if the set $L(s) \cap F$ is not empty, we let $f^s_{\min} := \min_{\sqsubseteq} L(s) \cap F$ denote the least element f_i with respect to the order $\sqsubseteq$. If such f_j does not exist, we define $f^s_{\min} := \bot$.

Definition 3.1 (Stratified CTMC)**.** We say that $\mathcal{C}$ is *stratified* with respect to φ iff for all s_1, s_2, it holds that:

- If $f^{s_1}_{\min} = \bot$ or $f^{s_1}_{\min} = f_k$, then $\mathbf{R}(s_1, s_2) = 0$.
- Otherwise (i. e., $f^{s_1}_{\min} \neq \bot$ and $f^{s_1}_{\min} \neq f_k$), if $\mathbf{R}(s_1, s_2) > 0$ and $f^{s_2}_{\min} \neq \bot$, then $f^{s_1}_{\min} \sqsubseteq f^{s_2}_{\min}$.

A state s with $f^s_{\min} = \bot$ is a bad state, and a state with $f^s_{\min} = f_k$ is a good state. (Note that there may be other states satisfying f_k as well.) Both good and bad states are absorbing. The intuition behind Def. 3.1 is that paths reaching bad states will not satisfy φ, while those reaching good states or other f_k-states may satisfy φ (provided the timing constraints are also satisfied).

Example 3.2. Consider the path formula $\varphi := f_1 \; U_{[0,2)} \; f_2 \; U_{[2,4)} \; f_3 \; U_{[2,4)} \; f_4 \; U_{[3,5)} \; f_5$. The CTMC in Fig. 1 is not stratified with respect to φ: we have $\mathbf{R}(s_2, s_1) > 0$, however, $f^{s_2}_{\min} = f_4 \not\sqsubseteq f_1 = f^{s_1}_{\min}$. Deleting this edge and the transition out of s_4 would result in a stratified CTMC with respect to φ. □

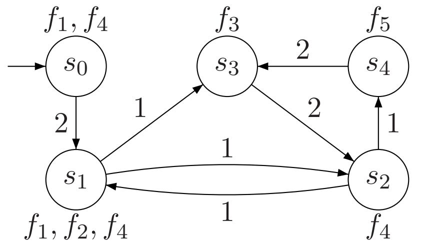


Figure 1: A non-stratified CTMC.

The notion of stratified CMTCs is the key to an efficient approximation algorithm. The essential idea is that we can reduce the model checking problem to one on a similar, stratified CTMC that preserves the relevant reachability probabilities. Further, our notion of stratified CTMCs solves a semantical problem in [2]: please refer to Section 6.2 for details.

4. Product CTMC

Given a CTMC and a CSL path formula φ, in this section we construct a stratified CTMC with respect to φ preserving the probability to satisfy φ. We first construct a deterministic finite automaton for φ in Subsection 4.1. Then, in Subsection 4.2 we build a product CTMC with the desired property.

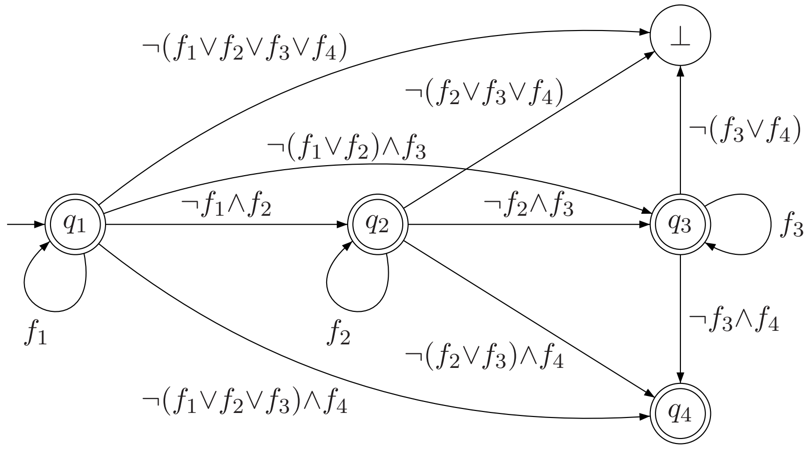
4.1. **Automaton for a CSL Formula.**

For a path formula $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2}\ \ldots f_k$, we first construct a simple deterministic finite automaton (DFA) that describes the required order of f_1-, f_2-, …, f_k-states.

Definition 4.1 (Formula automaton)**.** Let $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2}\ \ldots f_k$ be a CSL path formula with pairwise different atomic propositions. Then, the *formula automaton* $\mathcal{B}_\varphi = (\Sigma, Q, q_{in}, \delta, F)$ is defined by: $\Sigma = 2^{\{f_1,\ldots,f_k\}}$, $Q = \{q_1, q_2, \ldots, q_{k-1}, q_k, \perp\}$ with $q_{in} = q_1$ and $F = \{q_1, \ldots, q_k\}$. For $a \in \Sigma$, the transition relation δ is defined as follows:
(1) $\delta(q_i, a) = q_j$ if $i < k$; $i \le j$; $f_i, f_{i+1} \ldots f_{j-1} \notin a$; and $f_j \in a$;
(2) $\delta(q_i, a) = \perp$ if $i < k$ and the above clause does not apply;
(3) $\perp$ and q_k are absorbing.

As states $\perp$ and q_k have no outgoing transitions, δ is a partial transition function. Thus formula automata are actually partial DFAs. The words accepted by $\mathcal{B}_\varphi$ are finite traces $w \in \Sigma^*$ that can be extended to a trace $ww' \in \Sigma^\omega$ that satisfies the time-abstract formula of the form $f_1\ U\ f_2\ U \ldots U\ f_k$. The constructed finite automaton $\mathcal{B}_\varphi$ for this special class of formulas is deterministic, the number of states is linear in k. The number of transitions is $(k-1)2^k$; however, as we will see later, the product can be constructed in time (and size) linear in the size of the CTMC and in k.

Example 4.2. In Fig. 2 the formula automaton for $k = 4$ is illustrated. The initial state is q_1, final states are marked with a double circle. The transition labels indicate which subsets of AP are acceptable. For example, we have $\delta(q_1, \{f_1\}) = \delta(q_1, \{f_1, f_2\}) = q_1$, as both sets satisfy f_1. □



$\neg(f_1 \vee f_2 \vee f_3 \vee f_4)$
$\perp$
$\neg(f_2 \vee f_3 \vee f_4)$
$\neg(f_3 \vee f_4)$
$\neg(f_1 \vee f_2) \wedge f_3$
q_1
$\neg f_1 \wedge f_2$
q_2
$\neg f_2 \wedge f_3$
q_3
f_3
f_1
f_2
$\neg f_3 \wedge f_4$
$\neg(f_2 \vee f_3) \wedge f_4$
$\neg(f_1 \vee f_2 \vee f_3) \wedge f_4$
q_4

Figure 2: $\mathcal{B}_\varphi$ for $\varphi = f_1 \; U \; f_2 \; U \; f_3 \; U \; f_4$

4.2. **Product CTMC.**

Definition 4.3 (Product CTMC)**.** Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a CTMC and $\varphi = f_1 \; U_{I_1} \; f_2 \; U_{I_2} \dots f_k$ a path formula with pairwise different atomic propositions. Let $\mathcal{B}_\varphi$ be as constructed above. The product $\mathcal{C} \times \mathcal{B}_\varphi$ is a CTMC $(S', \mathbf{R}', L')$ where:

(1) $S' = S \times Q$,
(2) $\mathbf{R}'((s, q_i), (s', q'))$ equals $\mathbf{R}(s, s')$ if $s \models_{\mathcal{C}} f_i \vee f_{i+1} \vee \cdots \vee f_{k-1}$ and $q' = \delta(q_i, L(s') \cap \{f_1, \dots, f_k\})$, and equals 0 otherwise,
(3) the labeling function is defined by:
 - $L'(s, q_i) = L(s) \cap \{f_i, f_{i+1}, \dots, f_k\}$ for $1 \leq i \leq k$,
 - $L'(s, \perp) = \emptyset$.
(4) Given an initial distribution $\alpha : S \to [0, 1]$ of $\mathcal{C}$, the initial distribution of the product $\alpha' : S \times Q \to [0, 1]$ is defined by: $\alpha'(s, q)$ equals $\alpha(s)$ if $q = \delta(q_{in}, L(s) \cap \{f_1, \dots, f_k\})$, and equals 0 otherwise.

The product CTMC contains two kinds of absorbing states. In general, states (s, q) with $s \not\models \bigvee_{i=1}^{k} f_i$ are absorbing in the product, as well as states reached through a transition that does not follow the prescribed order of f_i. These two kinds of states can be considered bad states. On the other hand, good states of the form (s, q_k) with $s \models f_k$ are also absorbing. The behavior after such an absorbing state is irrelevant for the probability to satisfy φ.

Example 4.4. Consider the CTMC in Fig. 1, and consider the path formula $\varphi_1 := f_1 \; U_{[0,2)} \; f_2 \; U_{[0,2)} \; f_3 \; U_{[0,2)} \; f_4 \; U_{[0,2)} \; f_5$. The path $\sigma_1 := s_0 s_1 s_3 s_2 s_4 \dots$ does, if s_4 is reached before time 2, satisfy φ_1; however, the path $\sigma_2 := s_0 s_1 s_2 s_1 s_3 s_2 s_4 \dots$ does not. The product of this CTMC with $\mathcal{B}_{\varphi_1}$ is the CTMC depicted on the left of Fig. 3, which is stratified with respect to φ_1. State (s_4, q_5) is a *good state* – paths reaching this state before time 2 correspond to paths satisfying φ_1 in Fig. 1 –, while $(s_3, \perp)$ is a *bad state.*

For the same CTMC in Fig. 1, consider the path formula $\varphi_2 := f_1 \; U_{[1,3)} \; f_2 \; U_{[1,3)} \; f_3 \; U_{[1,3)} \; f_4$. The product CTMC $\mathcal{C} \times \mathcal{B}_{\varphi_2}$ is depicted on the right of Fig. 3. This product is stratified with respect to φ_2. The absorbing state (s_2, q_4) is a *good state.* □

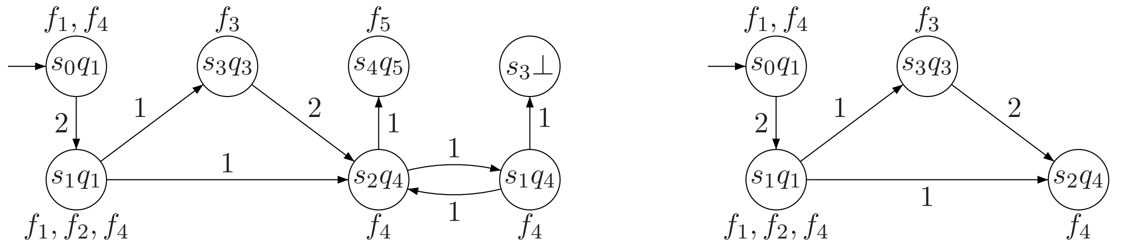


Figure 3: The reachable part of the product CTMC $\mathcal{C} \times \mathcal{B}_{\varphi_1}$ (left) and $\mathcal{C} \times \mathcal{B}_{\varphi_2}$ (right).

For a CTMC $\mathcal{C} = (S, \mathbf{R}, L)$ and a state $s \in S$, we use $\mathcal{C}|_s = (S', \mathbf{R}', L')$ to denote the *sub-CTMC* reachable from s, i. e., $S' \subseteq S$ is the states reachable from s, $\mathbf{R}'$ and L' are functions restricted to $S' \times S'$ and S', respectively.

Theorem 4.5 (Measure-preservation theorem)**.** *Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a CTMC and $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2}\ \dots f_k$ a path formula. Let $\mathcal{B}_\varphi$ denote the formula automaton. For $s \in S$, let $s_B = (s, \delta(q_{in}, L(s) \cap \{f_1, \dots, f_k\}))$. Then:*
(1) *$\mathcal{C} \times \mathcal{B}_\varphi|_{s_B}$ is stratified with respect to φ;*
(2) *$\Pr^{\mathcal{C}}_s(\varphi) = \Pr^{\mathcal{C}\times\mathcal{B}_\varphi|_{s_B}}_{s_B}(\varphi) = \Pr^{\mathcal{C}\times\mathcal{B}_\varphi}_{s_B}(\varphi)$.*

Proof. We prove first that $\mathcal{C}\times\mathcal{B}_\varphi|_{s_B}$ is stratified with respect to φ. Consider a state (s, q). By definition of the product CTMC, if $(s, q) \not\models_{\mathcal{C}\times\mathcal{B}_\varphi} \bigvee_{i=1}^{k-1} f_i$, then $s \not\models_{\mathcal{C}} \bigvee_{i=1}^{k-1} f_i$ or $q \in \{q_k, \bot\}$, so state (s, q) is absorbing and therefore trivially satisfies the stratification conditions. Now assume that $(s, q) \models_{\mathcal{C}\times\mathcal{B}_\varphi} \bigvee_{i=1}^{k-1} f_i$, $q \notin \{q_k, \bot\}$, and moreover assume (s', q') is a state with $\mathbf{R}'((s, q), (s', q')) > 0$ (with $\mathbf{R}'$ as in Def. 4.3). By the definition of the transitions of $\mathcal{B}_\varphi$, we have $q' = \delta(q, L(s') \cap \{f_1, \dots, f_k\})$. Now assume $f^{(s',q')}_{\min} \neq \bot$: it remains to be shown that $f^{(s,q)}_{\min} \sqsubseteq f^{(s',q')}_{\min}$. Let $1 \leq x \leq k$ such that $f_x = f^{(s,q)}_{\min}$, and let $1 \leq y \leq k$ be such that $q = q_y$. The indices x' and y' are defined similarly for (s', q'). By definition of transitions of $\mathcal{B}_\varphi$ and product CTMC, it is routine to verify that $x = y$ and $x' = y'$. Moreover, in $\mathcal{B}_\varphi$, $q' = \delta(q, L(s') \cap \{f_1, \dots, f_k\})$ implies that $y' \geq y$, which shows that $x \leq x'$, proving $f^{(s,q)}_{\min} \sqsubseteq f^{(s',q')}_{\min}$.

Now we prove the second clause. Obviously, states not reachable from s_B can be safely removed, thus $\Pr^{\mathcal{C}\times\mathcal{B}_\varphi|_{s_B}}_{s_B}(\varphi) = \Pr^{\mathcal{C}\times\mathcal{B}_\varphi}_{s_B}(\varphi)$. We next prove that $\Pr^{\mathcal{C}}_s(\varphi) = \Pr^{\mathcal{C}\times\mathcal{B}_\varphi}_{s_B}(\varphi)$ by showing that $\sigma \mapsto \sigma_B$ (the canonical mapping from paths in $\mathcal{C}$ to paths in $\mathcal{C}\times\mathcal{B}_\varphi$) preserves the standard probability measures between the probability spaces. To this end, it is enough to show that given a cylinder set C_B over $\mathcal{C} \times \mathcal{B}_\varphi$, its reverse image $C = \{\sigma | \sigma_B \in C_B\}$ satisfies $\Pr^{\mathcal{C}}_s(C) = \Pr^{\mathcal{C}\times\mathcal{B}_\varphi}_{s_B}(C_B)$.

Stated briefly, we now show that paths in $\mathcal{C}$ and in $\mathcal{C} \times \mathcal{B}_\varphi$ correspond to each other because we only add some (bounded) information about the past to the states.

Let us first describe the canonical mapping $\sigma \mapsto \sigma_B$. Assume given a path $\sigma = s_0t_0s_1t_1\dots$ in $\mathcal{C}$. The corresponding path in $\mathcal{C} \times \mathcal{B}_\varphi$ is $\sigma_B = (s_0, q^0)t_0(s_1, q^1)t_1\dots$, where $q^0 = \delta(q_{in}, L(s_0) \cap \{f_1, \dots, f_k\})$ and $q^{i+1} = \delta(q^i, L(s_{i+1}) \cap \{f_1, \dots, f_k\})$ for all $i \geq 1$, as long as the (s_i, q^i) are not absorbing. However, if (s_n, q^n) is absorbing for some n, then σ_B is defined to be the finite path $(s_0, q^0)t_0(s_1, q^1)t_1\dots(s_n, q^n)$, where (s_n, q^n) is the first absorbing state encountered. Note that $\sigma \models_{\mathcal{C}} \varphi$ iff $\sigma_B \models_{\mathcal{C}\times\mathcal{B}_\varphi} \varphi$.

Let $C_B = Cyl((s_0, q^0), I_0, (s_1, q^1), \ldots, (s_n, q^n))$ and C be as above. By definition of a cylinder set, $\mathbf{R}'((s_i, q^i), (s_{i+1}, q^{i+1})) > 0$ for all $i < n$, therefore (s_i, q^i) is not absorbing (for $i < n$) and $q^{i+1} = \delta(q^i, L(s_{i+1}) \cap \{f_1, \ldots, f_k\})$. Now assume that some path $\sigma = s'_0 t_0 s'_1 t_1 \ldots \in C$; then it must hold that $s'_0 = s_0$, $t_0 \in I_0$, $s'_1 = s_1$, $t_1 \in I_1$, $\ldots$, and $s'_n = s_n$. Therefore, $C \subseteq Cyl(s_0, I_0, s_1, \ldots, s_n)$. On the other hand, for all paths $\sigma \in Cyl(s_0, I_0, s_1, \ldots, s_n)$, it is easy to prove that $\sigma_B \in C_B$. So, $C \supseteq Cyl(s_0, I_0, s_1, \ldots, s_n)$, and together, $C = Cyl(s_0, I_0, s_1, \ldots, s_n)$. It is now an easy calculation to verify that $\Pr^{\mathcal{C}}_s(C) = \Pr^{\mathcal{C}\times\mathcal{B}_\varphi}_{s_B}(C_B)$.

The reverse image of the set of $\mathcal{C} \times \mathcal{B}_\varphi$-paths satisfying φ is exactly the set of $\mathcal{C}$-paths satisfying φ. Since these sets are measurable, both can be decomposed into countable unions of corresponding cylinder sets in $\mathcal{C}$ and $\mathcal{C} \times \mathcal{B}_\varphi$, respectively. Thus, the theorem follows. $\square$

5. Characterizing the Probability $\Pr_\alpha(\varphi)$

For a path formula φ, together with a stratified CTMC with respect to φ, this section aims at a recursive characterization of the probability $\Pr_\alpha(\varphi)$ starting from an arbitrary initial distribution α.

We first introduce some notation. For an interval I and $0 \le x$, we let $I \ominus x$ denote the set $\{t - x \mid t \in I \wedge t \ge x\}$. For example, $[3, 8) \ominus 5 = [0, 3)$. Then, for $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2} \ldots f_k$ and $x < \sup I_1$, we let $\varphi \ominus x$ denote the formula $f_1\ U_{I_1 \ominus x}\ f_2\ U_{I_2 \ominus x} \ldots f_k$. For $1 \le j', j \le k$, define $f_{j'\ldots j} := \bigvee_{i=j'}^{j} f_i$; for $1 \le j < k$, define $\varphi_j := f_j\ U_{I_j}\ f_{j+1}\ U_{I_{j+1}} \ldots f_k$. As a degenerate case of $\Pr_s(\varphi_j)$, let $\Pr_s(f_k) := 1$ if $s \models f_k$ and 0 otherwise. For Φ, we denote by $\mathcal{C}[\Phi]$ the CTMC obtained by $\mathcal{C}$ by making states satisfying Φ absorbing – by cutting transitions out of all states satisfying Φ. Moreover, let $\mathbf{I}_\Phi$ denote the indicator matrix defined by: $\mathbf{I}_\Phi(s, s) = 1$ if $s \models \Phi$, and $\mathbf{I}_\Phi(s, s') = 0$ otherwise.

5.1. Left-Closed Intervals.

For the moment, we restrict our attention to until formulas where all timing constraints have the form $I_i = [a_i, b_i)$. The following theorem characterizes the probability for this case:

Theorem 5.1. *Let $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2} \ldots f_k$ be a CSL path formula with pairwise different atomic propositions, and assume all $I_i = [a_i, b_i)$ are left-closed. Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a stratified CTMC with respect to φ. We write the vector $(\Pr^{\mathcal{C}}_s(\psi))_{s \in S}$ as $\Pr^{\mathcal{C}}_{(\cdot)}(\psi)$.*

(1) *Assume $0 < a_1$. Then,*
$$\Pr^{\mathcal{C}}_\alpha(\varphi) = \pi^{\mathcal{C}[\neg f_1]}(\alpha, a_1) \cdot \mathbf{I}_{f_1} \cdot \Pr^{\mathcal{C}}_{(\cdot)}(\varphi \ominus a_1) \tag{5.1}$$
where $\pi^{\mathcal{C}[\neg f_1]}(\alpha, a_1)$ is the transient distribution at time a_1 in the CTMC $\mathcal{C}[\neg f_1]$.

(2) *Assume $0 = a_1 = \ldots = a_{j-1} < a_j < b_1$ for some $j \in \{2, \ldots, k-1\}$. Then,*
$$\Pr^{\mathcal{C}}_\alpha(\varphi) = \pi^{\mathcal{C}[\neg f_{1\ldots j}]}(\alpha, a_j) \cdot \mathbf{I}_{f_{1\ldots j}} \cdot \Pr^{\mathcal{C}}_{(\cdot)}(\varphi \ominus a_j) \tag{5.2}$$

(3) *Assume $0 = a_1 = \ldots = a_{j-1} < b_1 \le a_j$ for some $j \in \{2, \ldots, k-1\}$. Let $j' \le j$ be the largest integer such that $b_1 \notin I_{j'-1}$. Then,*
$$\Pr^{\mathcal{C}}_\alpha(\varphi) = \pi^{\mathcal{C}[\neg f_{1\ldots j}]}(\alpha, b_1) \cdot \mathbf{I}_{f_{j'\ldots j}} \cdot \Pr^{\mathcal{C}[\neg f_{j'\ldots k-1}]}_{(\cdot)}(\varphi_{j'} \ominus b_1) \tag{5.3}$$

(4) *Assume* $0 = a_1 = \ldots = a_{k-1}$. *Let* $j' \leq k$ *be the largest integer such that* $b_1 \notin I_{j'-1}$. *Then,*

$$\Pr^{\mathcal{C}}_{\alpha}(\varphi) = \pi^{\mathcal{C}[f_k]}(\alpha, b_1) \cdot \mathbf{I}_{f_{j' \ldots k}} \cdot \Pr^{\mathcal{C}[\neg f_{j' \ldots k-1}]}_{(\cdot)}(\varphi_{j'} \ominus b_1) \tag{5.4}$$

If $b_1 = \infty$, *we replace* $\pi^{\mathcal{C}[f_k]}(\alpha, b_1)$ *in this equation by the corresponding steady-state distribution.*

The key idea of the theorem is a *property-driven transient analysis*. In the first clause we have $a_1 > 0$, thus for any path σ satisfying φ it must hold $\sigma(t) \models f_1$ for all $t \in [0, a_1)$. Thus, we make all states satisfying $\neg f_1$ absorbing, and compute the transient distribution $\pi^{\mathcal{C}[\neg f_1]}(\alpha, a_1)$. Furthermore, the multiplication with the matrix $\mathbf{I}_{f_1}$ removes the probabilities in states satisfying $\neg f_1$ – thus resulting in a subdistribution. Starting with this subdistribution, the formula will also be reduced by duration a_1. In the other clauses, we consider the interval $[0, a_j)$ or $[0, b_1)$, which is the common prefix of the intervals $I_1, \ldots, I_{j-1}$. Thus during this time the formula $f_{1 \ldots j}$ must be satisfied. Here the assumption of stratification is crucial: otherwise one might be able jump forward and back between states satisfying f_1 and f_j, which is illustrated in the following example.

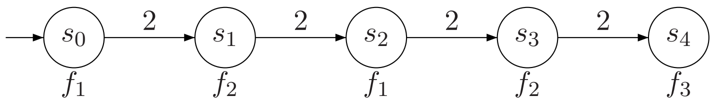


Figure 4: A CTMC with $\Pr_{s_0}(f_1\ U_{[0,1)}\ f_2\ U_{[0,1)}\ f_3) = 0$.

Example 5.2. Consider the CTMC depicted in Fig. 4 and consider the path formula $\varphi = f_1\ U_{[0,1)}\ f_2\ U_{[0,1)}\ f_3$. Obviously the probability of the set of paths starting from s_0 satisfying φ is 0. Since the CTMC is not stratified with respect to φ, Thm. 5.1 cannot be applied directly: the product shall be constructed first. In the product CTMC, no states labelled with f_3 will be reached, thus giving the probability 0, as desired.

Proof of Thm. 5.1. We start with Eqn. (5.1). Let a_1 and the other notation be as in the theorem. For $s' \in S$, define the event $Z_1(s') := \{\sigma \mid \sigma(a_1) = s' \wedge \forall t \in [0, a_1).\ \sigma(t) \models f_1\}$, consisting of paths which occupy state s' at time a_1 and occupy f_1-states during the time interval $[0, a_1)$. Obviously, $\{\sigma \mid \sigma \models \varphi\} \subseteq \bigcup_{s' \models f_1} Z_1(s')$.[1] Fix first α_s as an initial distribution with $\alpha_s(s) = 1$ and $s \models f_1$. By the law of total probability, we have:

$$\begin{aligned} \Pr^{\mathcal{C}}_{s}(\varphi) &= \sum_{s' \models f_1} \Pr^{\mathcal{C}}_{s}(Z_1(s')) \cdot \Pr^{\mathcal{C}}_{s}(\varphi \mid Z_1(s')) \\ &= \sum_{s' \models f_1} \pi^{\mathcal{C}[\neg f_1]}(s, a_1)(s') \cdot \Pr^{\mathcal{C}}_{s}(\varphi \mid Z_1(s')) \end{aligned}$$

[1]Strictly speaking, this does not hold always because there may be paths that enter an $(f_2 \wedge \neg f_1)$-state exactly at time a_1; however, such paths are contained in a (generalization of) cylinder sets like $Cyl(s, [a_1, a_1], \ldots)$, whose measure is 0.

The latter equality follows from the definition of $Z_1(s')$. By the Markov property of CTMCs:

$$\begin{aligned} &= \sum_{s' \models f_1} \pi^{\mathcal{C}[\neg f_1]}(s, a_1)(s') \cdot \mathrm{Pr}^{\mathcal{C}}_{s'}(\varphi \ominus a_1) \\ &= \sum_{s' \in S} \pi^{\mathcal{C}[\neg f_1]}(s, a_1)(s') \cdot \mathbf{1}_{s' \models f_1} \cdot \mathrm{Pr}^{\mathcal{C}}_{s'}(\varphi \ominus a_1) \end{aligned}$$

where $\mathbf{1}_{s' \models f_1}$ is 1 if $s' \models f_1$ and 0 otherwise. Note that $\mathrm{Pr}^{\mathcal{C}}_{\alpha}(\varphi) = \sum_{s \in S} \alpha(s) \mathrm{Pr}^{\mathcal{C}}_s(\varphi) = \sum_{s \models f_1} \alpha(s) \mathrm{Pr}^{\mathcal{C}}_s(\varphi)$, thus Eqn. (5.1) follows.

We now jump to the proof of Eqn. (5.3). This proof is more involved, but follows the same lines. Define the event $Z_3(s') := \{\sigma \mid \sigma(b_1) = s' \wedge \forall t \in [0, b_1).\sigma(t) \models f_{1\ldots j}\}$. Again, $\{\sigma \mid \sigma \models \varphi\} \subseteq \bigcup_{s' \models f_{j' \ldots j}} Z_3(s')$, and again, fix α_s as an initial distribution with $\alpha_s(s) = 1$ and $s \models f_{1\ldots j}$. We have:

$$\begin{aligned} \mathrm{Pr}^{\mathcal{C}}_s(\varphi) &= \sum_{s' \models f_{j' \ldots j}} \mathrm{Pr}^{\mathcal{C}}_s(Z_3(s')) \cdot \mathrm{Pr}^{\mathcal{C}}_s(\varphi \mid Z_3(s')) \\ &= \sum_{s' \models f_{j' \ldots j}} \pi^{\mathcal{C}[\neg f_{1\ldots j}]}(s, b_1)(s') \cdot \mathrm{Pr}^{\mathcal{C}}_s(\varphi \mid Z_3(s')) \end{aligned}$$

where the latter equality follows from the definition of $Z_3(s')$. Now let $\sigma \in Z_3(s')$, thus $\sigma(b_1) = s'$, and $\sigma(t) \models f_{1\ldots j}$ for all $0 \leq t < b_1$. Let σ' denote the suffix path defined by $\sigma'(x) := \sigma(x + b_1)$.

Now, $\sigma \models \varphi$ implies that at time b_1, σ has reached a state in a stratum from $q_{j'}, \ldots, q_j$, so σ' satisfies $\varphi_{j'} \ominus b_1$. On the other hand, every path $\sigma \in Z_3(s')$ whose corresponding σ' satisfies $\varphi_{j'} \ominus b_1$ also satisfies φ (because $\mathcal{C}$ is stratified). Again, $\mathrm{Pr}^{\mathcal{C}}_s(\varphi \mid Z_3(s')) = \mathrm{Pr}^{\mathcal{C}}_{s'}(\varphi_{j'} \ominus b_1)$, thus

$$\begin{aligned} \mathrm{Pr}^{\mathcal{C}}_s(\varphi) &= \sum_{s' \models f_{j' \ldots j}} \pi^{\mathcal{C}[\neg f_{1\ldots j}]}(s, b_1)(s') \cdot \mathrm{Pr}^{\mathcal{C}}_{s'}(\varphi_{j'} \ominus b_1) \\ &= \sum_{s' \in S} \pi^{\mathcal{C}[\neg f_{1\ldots j}]}(s, b_1)(s') \cdot \mathbf{1}_{s' \models f_{j' \ldots j}} \cdot \mathrm{Pr}^{\mathcal{C}}_{s'}(\varphi_{j'} \ominus b_1) \quad . \end{aligned}$$

However, $\mathcal{C}$ needs not be stratified w. r. t. $\varphi_{j'} \ominus b_1$, so to simplify the subsequent calculations, we restratify it: $\mathcal{C}[\neg f_{j' \ldots k-1}]$ is stratified w. r. t. $\varphi_{j'} \ominus b_1$. Eqn. (5.3) for general initial distribution α follows as in the case of Eqn. (5.1).

The proof for Eqn. (5.2) is similar to the proof for Eqn. (5.3), except that b_1 has to be replaced by a_j and j' by 1.

For Eqn. (5.4), we can again make a similar proof. First assume that $j' = k$. In that case, the paths that have reached an f_k-state at any time in the interval $I_{k-1} = I_1$ are exactly the paths that satisfy φ. They have the same probability as the paths in $\mathcal{C}[f_k]$ that are in an f_k-state exactly at time b_1. Therefore,

$$\mathrm{Pr}^{\mathcal{C}}_s(\varphi) = \sum_{s' \models f_k} \pi^{\mathcal{C}[f_k]}(s, b_1)(s') = \sum_{s' \in S} \pi^{\mathcal{C}[f_k]}(s, b_1)(s') \cdot \mathbf{1}_{s' \models f_k} \tag{5.5}$$

With the usual assumption $f_{k\ldots k-1} = \mathit{false}$, the theorem follows immediately.

If $j' < k$, besides the paths mentioned above, other paths satisfy φ, namely paths that reach an f_k-state during the interval $I_{k-1} \setminus I_1 = [b_1, b_{k-1})$ (and avoid f_k-states earlier).

These are the paths that satisfy $(f_1 \wedge \neg f_k)\ U_{I_1}\ \ldots\ U_{I_{k-2}}\ (f_{k-1} \wedge \neg f_k)\ U_{I_{k-1} \setminus I_1}\ f_k$. Their probability is, according to Eqn. (5.3),

$$\sum_{s' \in S} \pi^{\mathcal{C}[\neg(f_{1\ldots k-1} \wedge \neg f_k)]}(s, b_1)(s') \cdot \mathbf{I}_{s' \models f_{j' \ldots k-1} \wedge \neg f_k} \cdot \Pr_{s'}^{\mathcal{C}[\neg f_{j' \ldots k-1}]}(\varphi_{j'} \ominus b_1)$$

Note that $\mathcal{C}[\neg(f_{1\ldots k-1} \wedge \neg f_k)] = \mathcal{C}[f_k]$. Adding this term to Eqn. (5.5) produces the desired probability.

We still have to prove Eqn. (5.4) for $b_1 = \infty$. In that case, all timing constraints are trivial ($[a_i, b_i) = [0, \infty)$) and $j' = k$. Therefore, $\Pr_s^{\mathcal{C}}(\varphi)$ is just the probability to reach an f_k-state eventually, which is exactly $\lim_{b_1 \to \infty} \pi^{\mathcal{C}[f_k]}(s, b_1) \mathbf{I}_{f_k} \Pr_{(\cdot)}^{\mathcal{C}[\neg f_{k\ldots k-1}]}(f_k)$. □

5.2. **Closed Intervals.** In Thm. 5.1, we have considered formula φ with left-closed intervals. Now we discuss that a slight generalization of it can be used to handle closed intervals. Thus, below we assume that $I_i = [a_i, b_i]$.

The proof of Thm. 5.1 can be extended easily to hold also for closed intervals. Clause 3 may lead to formulas containing degenerate intervals $[0, 0]$: As $b_1 \in [a_1, b_1]$, often $j' = 1$ in this clause. (We have to assume, as an additional simplification of notation, $I_0 := \emptyset$.) As a consequence, $\varphi_{j'} \ominus b_1 = f_1\ U_{[0,0]}\ f_2\ U_{I_2 \ominus b_1}\ \ldots f_k$.

Further, if the original φ already contained a degenerate interval, say $a_1 = b_1$, so $I_1 = \{a_1\}$, applying Clause 1 will also lead to a formula containing $[0, 0]$. These situations can be handled by the following lemma:

Lemma 5.3. *Let $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2} \ldots f_k$ be a CSL path formula. Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a stratified CTMC with respect to φ. Moreover, assume $I_1 = \ldots = I_{j-1} = [0, 0]$ for $2 \leq j \leq k$. Then, $\Pr_s^{\mathcal{C}}(\varphi) = \Pr_s^{\mathcal{C}}(f_j\ U_{I_j}\ \ldots f_k)$ for all $s \in S$.*

Proof. Assume a path σ satisfies φ. The degenerate intervals force $t_1 = t_2 = \ldots = t_{j-1} = 0$, thus no conditions relating to $f_1, \ldots, f_{j-1}$ need to be checked. □

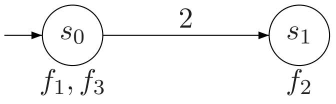


Figure 5: A CTMC with $\Pr_{s_0}(f_1\ U_{[0,1]}\ f_2\ U_{[1,2]}\ f_3) \neq \Pr_{s_0}(f_1\ U_{[0,1)}\ f_2\ U_{[1,2]}\ f_3)$.

Example 5.4. Consider the CTMC in Fig. 5 and the path formula $\varphi = f_1\ U_{[0,1]}\ f_2\ U_{[1,2]}\ f_3$. Then, $\Pr_{s_0}(\varphi)$ is the probability to stay in s_0 for at least one time unit ($\psi(0, E(s_0) \cdot 1)$ in the notation of Section 6.3 below), since we can choose $t_1 = t_2 = 1$ if $\sigma_T[0] \geq 1$. Applying Clause 3 of Thm. 5.1, we get $j = 2$, $j' = 1$ and $\Pr_{s_0}(\varphi) = \pi^{\mathcal{C}[\neg f_{1\ldots 2}]}(s_0, 1) \cdot \mathbf{I}_{f_{1\ldots 2}} \cdot \Pr_{(\cdot)}^{\mathcal{C}[\neg f_{1\ldots 2}]}(f_1\ U_{[0,0]}\ f_2\ U_{[0,1]}\ f_3) = \pi^{\mathcal{C}}(s_0, 1) \cdot \mathbf{I} \cdot \Pr_{(\cdot)}^{\mathcal{C}}(f_2\ U_{[0,1]}\ f_3) = (e^{-2}, 0) \cdot (1, 0)^T = e^{-2}$, the correct value. (In [22], we defined j' slightly differently, producing $j' = 2$ and consequently $\Pr_{s_0}(\varphi) = 0$. Our earlier definition worked only for left-closed intervals.) □

Below we apply the theorem to two formulas, and thereby get the well-known result [5] for the case of binary until for the case $k = 2$. As above, $\mathcal{C}$ is stratified and $\varphi = f_1 \; U_{I_1} \; f_2 \; U_{I_2} \ldots f_k$.

(1) *Reachability probability.* Assume that $I_1 = \ldots = I_{k-1} = [0, b]$. Then, it holds $\Pr_s(\varphi) = \Pr_s(\Diamond_{[0,b]} f_k)$, which is the probability to reach an f_k-state within time b.
(2) *Interval reachability.* Assume that $I_1 = \ldots = I_{k-1} = [a, b]$ with $a < b$. Then, it holds $\Pr_s(\varphi) = \sum_{s' \models f_1} \pi^{\mathcal{C}[\neg f_1]}(s, a)(s') \cdot \Pr_{s'}(\Diamond_{[0,b-a]} f_k)$, which is the interval reachability probability of staying in f_1-states until time a and then moving to an f_k-state before time b has passed.

5.3. **Other Intervals.** First, the following lemma states properties of the probabilities for binary until with different interval types:

Lemma 5.5 (Closure of Intervals for Binary Until)**.** *Let $s \in S$. Assume given two nonempty intervals I, J such that* $\inf I = \inf J$ *and* $\sup I = \sup J$*. Then, it holds:*

(1) *If $0 \in I \Leftrightarrow 0 \in J$, then $s \models \mathcal{P}_{\trianglelefteq p}(\Phi \; U_I \; \Psi)$ iff $s \models \mathcal{P}_{\trianglelefteq p}(\Phi \; U_J \; \Psi)$ for $\trianglelefteq \in \{<, \leq, \geq, >\}$.*
(2) *Otherwise, assume w. l. o. g. $0 \in I$ and $0 \notin J$, and assume $0 < p < 1$. Then, $s \models \mathcal{P}_{\trianglerighteq p}(\Phi \; U_I \; \Psi) \wedge \Phi$ iff $s \models \mathcal{P}_{\trianglerighteq p}(\Phi \; U_J \; \Psi)$, for $\trianglerighteq \in \{\geq, >\}$. Similarly, $s \models \mathcal{P}_{\trianglelefteq p}(\Phi \; U_I \; \Psi) \vee \neg\Phi$ iff $s \models \mathcal{P}_{\trianglelefteq p}(\Phi \; U_J \; \Psi)$, for $\trianglelefteq \in \{<, \leq\}$.*

The lemma follows immediately from the definition of the measure of cylinder set. To see why we have to treat the case $\inf I = 0$ separately (not distinguished in [5]), assume that $\Phi = \mathcal{P}_{\leq 0.1}(f_2 \; U_{(0,1]} \; f_1)$ and consider the CTMC depicted in Fig. 5: obviously we have $s_0 \models \Phi$ as $s_0 \not\models f_2$. However, $s_0 \not\models \mathcal{P}_{\leq 0.1}(f_2 \; U_{[0,1]} \; f_1)$ as s_0 satisfies f_1 directly. The formula Φ is equivalent to $\mathcal{P}_{\leq 0.1}(f_2 \; U_{[0,1]} \; f_1) \vee \neg f_2$.

For until formulas with arbitrary multiplicity, we have discussed the case that all of the intervals are left-closed or closed. Other cases can be handled in a way similar to Lemma 5.5. However, to avoid too many technicalities, we skip these details.

6. Model Checking Algorithm

Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a CTMC, $s \in S$, and Φ be a CSL formula. The model checking problem is to check whether $s \models \Phi$. In the following two sections, we discuss that the model checking problem is decidable and provide an efficient algorithm for approximate computation of $\Pr_s(\psi)$.

6.1. **Model Checking CSL is Decidable.** The standard algorithm to solve CTL-like model checking problems recursively computes the sets of states satisfying Ψ, denoted by $Sat(\Psi)$, for all state subformulas Ψ of Φ. For CSL, the cases where Ψ is an atomic proposition, a negation or a conjunction are given by: $Sat(a) = \{s \in S \mid a \in L(s)\}$, $Sat(\neg\Psi_1) = S \backslash Sat(\Psi_1)$ and $Sat(\Psi_1 \wedge \Psi_2) = Sat(\Psi_1) \cap Sat(\Psi_2)$.

The case that Ψ is the probabilistic operator is the challenging part. Let $\Psi = \mathcal{P}_{\trianglelefteq p}(\varphi)$ with $\varphi = \Psi_1 \; U_{I_1} \; \Psi_2 \; U_{I_2} \ldots \Psi_k$. By the semantics, checking Ψ is equivalent to checking whether $\Pr_s(\varphi)$ meets the bound $\trianglelefteq p$, i. e., whether $\Pr_s(\varphi) \trianglelefteq p$. Assume that the sets $Sat(\Psi_i)$ have been calculated recursively. We replace $\Psi_1, \ldots, \Psi_k$ by fresh (pairwise different) atomic propositions $f_1, \ldots, f_k$ and extend the label of state s by f_i if $s \in Sat(\Psi_i)$. The so obtained path formula is $\psi := f_1 \; U_{I_1} \; f_2 \; U_{I_2} \ldots f_k$, and obviously we have $\mathrm{Pr}_s(\varphi) = \mathrm{Pr}_s(\psi)$. The steps needed to characterize $\mathrm{Pr}_s(\psi)$ are:

(i) Construct the formula automaton $\mathcal{B}_\psi$.
(ii) Build the product $\mathcal{C} \times \mathcal{B}_\psi$, which by Thm. 4.5 is a stratified CTMC w. r. t. ψ.
(iii) Apply Thm. 5.1 repeatedly to compute $\mathrm{Pr}_s(\psi)$.

Thus, the decidability for the probabilistic formula reduces to checking whether $\mathrm{Pr}_s(\psi) \trianglelefteq p$ holds true in the product CTMC. After applying Thm. 5.1 a finite number of times, we see that $\mathrm{Pr}_s(\psi)$ reduces to a product of transient probabilities. We can now follow the argumentation in [2]: Although the calculations differ slightly, $\mathrm{Pr}_s(\psi)$ still is a finite sum $\sum_k \eta_k e^{\delta_k}$ (with algebraic η_k and δ_k). For such an expression, [2] proved that it can be decided whether it is $\trianglelefteq p$, for $p \in \mathbb{Q}$. Thus, we still have:

Theorem 6.1 ([2], Thm. 1)**.** *Model checking CSL is decidable.*

6.2. **Usefulness of Stratification.** Our notion of stratified CTMCs solves a semantical problem in [2], which we recently pointed out in [13]. Very briefly, Aziz *et al.* [2] gave an algorithm that did not use the t_i (in the semantics of until formulas) explicitly, which led to incorrect results for non-stratified CTMCs.

Consider the CTMC depicted in Fig. 4 and the formula $\varphi = f_1 \; U_{[0,1)} \; f_2 \; U_{[0,1)} \; f_3$ in Example 5.2. For this example, the algorithm in [2] calculates the probability that a path satisfies, a. o., the conditions: it stays in $f_1 \vee f_2$-states during time $[0, 1)$, thus giving a wrong result. This problem does not occur provided that the CTMC is stratified.

6.3. **Efficient Algorithm for Approximating** $\mathrm{Pr}_s(\psi)$**.** We first explain how to combine steps (i) and (ii) mentioned above, without having to construct the full automaton $\mathcal{B}_\psi$. Most parts of the construction of $\mathcal{C} \times \mathcal{B}_\varphi$ depend on $\mathcal{C}$ only and do not require much information about $\mathcal{B}_\varphi$. For example, for the state space, it is enough to generate k copies of every state in $\mathcal{C}$, which requires time $\mathcal{O}(|S|k)$. When constructing the transitions according to Clause 2 of Def. 4.3, one has to check $q' = \delta(q_i, L(s') \cap \{f_1, \ldots, f_k\})$, but even this can be done without actually constructing $\mathcal{B}_\varphi$ by using the definition of δ (Def. 4.1) directly. Therefore, the overall time complexity to find all transitions of $\mathcal{C} \times \mathcal{B}_\varphi$ is $|\mathbf{R}|$ times the number of copies that its source state may have, i. e., $\mathcal{O}(|\mathbf{R}|k)$, which is also the maximal total number of transitions.

The usual numerical algorithm to compute the matrix exponential $e^{\mathbf{Q}t}$ is based on *uniformization* [20]. This algorithm executes most calculations on the uniformized DTMC. For a CTMC, we say that λ is a uniformization rate if $\lambda \geq \max_{s \in S}(E(s) - \mathbf{R}(s, s))$.

Definition 6.2. Let $\mathcal{C} = (S, \mathbf{R}, L)$ be a CTMC. The uniformized DTMC of $\mathcal{C}$ with respect to the uniformization rate λ is $\mathit{uni}(\mathcal{C}) = (S, \mathbf{P}, L)$ where $\mathbf{P}(s, s') = \mathbf{R}(s, s')/\lambda$ if $s \neq s'$ and $\mathbf{P}(s, s) = 1 - \mathbf{P}(s, S \setminus \{s\})$.

Let $\mathbf{P}$ denote the transition matrix of the uniformized DTMC $\mathit{uni}(\mathcal{C})$, thus it holds that $\mathbf{P} = I + \mathbf{Q}/\lambda$ where I denotes the identity matrix. For $t > 0$, then:

$$\pi(\alpha, t) = \pi(\alpha, 0) e^{(\mathbf{P}-I)\lambda t} = \pi(\alpha, 0) e^{-\lambda t} \sum_{i=0}^{\infty} \frac{(\lambda t)^i}{i!} \mathbf{P}^i = \sum_{i=0}^{\infty} \psi(i, \lambda t) \vec{v}(i) \tag{6.1}$$

In this formula, $\psi(i, \lambda t) = e^{-\lambda t} \cdot \frac{(\lambda t)^i}{i!}$ denotes the i-th Poisson probability with parameter λt, i. e., the probability to see precisely i transitions within time t. The vector $\vec{v}(i)$ is the transient probability of $uni(\mathcal{C})$ after i transitions, i. e., $\vec{v}(i) = \pi(\alpha, 0)\mathbf{P}^i$. The infinite sum is approximated, by picking $\mathcal{O}(\sqrt{\lambda t})$ terms with large $\psi(i, \lambda t)$, using the Fox–Glynn algorithm [9, 14]. To find the $\vec{v}(i)$ for Eqn. (6.1), one requires $\mathcal{O}(\lambda t)$ matrix–vector multiplications [5]. The following lemma states the complexity of our algorithm:

Lemma 6.3 (Complexity)**.** *Let* $|\mathbf{R}|$ *denote the number of transitions of* $\mathcal{C}$ *and* $\lambda \in \mathbb{R}_{>0}$ *the* uniformization rate *satisfying* $\lambda = \max_{s\in S}(E(s) - \mathbf{R}(s, s))$. *For each formula* $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2} \ldots f_k$, *the probability* $\Pr^{\mathcal{C}\times\mathcal{B}_\varphi}_{s_B}(\varphi)$ *can be approximated:*

- *in time in* $\mathcal{O}(|\mathbf{R}|k \cdot \lambda b)$ *if* $b = \sup I_{k-1}$ *is finite,*
- *in time in* $\mathcal{O}(|\mathbf{R}|k \cdot \lambda b + (|S|k)^3)$ *if* $\sup I_{k-1}$ *is infinite, where* $|S|$ *is the number of states in* $\mathcal{C}$ *and* $b = \max(\{\inf I_{k-1}\} \cup \{\sup I_i | 1 \le i < k\} \setminus \{\infty\})$.

The space complexity is in $\mathcal{O}(|\mathbf{R}|k)$.

Proof. Recall that the formula automaton $\mathcal{B}_\varphi$ is deterministic, and the size of the product automaton is $\mathcal{O}(|\mathbf{R}|k)$ which is both linear in the size of the CTMC and the formula. This proves the space complexity.

For the time complexity assume first $b < \infty$ with $b = \sup I_{k-1}$. Applying Thm. 5.1, the probability $\Pr^{\mathcal{C}\times\mathcal{B}_\varphi}_{s_B}(\varphi)$ can be expressed as a sequence of transient probability analyses, which can be efficiently approximated by a sequence of uniformization analyses. The complexity of these analyses is linear in the size of the product automaton, and also linear in λb.

For the second case $\sup I_{k-1} = \infty$, by Thm. 5.1, a sequence of transient probability analyses is followed by one steady-state analysis, which can be done with Gaussian elimination for the equation systems $\pi \cdot \mathbf{Q}' = 0$ and $\sum_{s\in S'} \pi(s) = 1$, the complexity of which is $\mathcal{O}((|S|k)^3)$. Thus the complexity for this case follows. □

Thus, with the notion of stratified CTMC, we achieve polynomial complexity. Our algorithm therefore improves the work of [2], where only multiple until formulas with suitable timing constraints can be checked polynomially. In the worst case, [2] has to decompose a CSL formula into $\mathcal{O}((k-1)^{k-1})$ formulas with suitable timing, thus resulting in an overall time complexity of $\mathcal{O}(|\mathbf{R}| \cdot \lambda b(k-1)^k)$ or $\mathcal{O}((|\mathbf{R}|k \cdot \lambda b + (|S|k)^3) \cdot (k-1)^{k-1})$, respectively.

7. Related Work

The logic CSL was first proposed in [1], in which the model checking problem is shown to be decidable. Our paper gives a practical solution: it shows that the relevant probabilities can be approximated efficiently. For the case of binary until path formula, Baier *et al.* [5] have presented an approximate algorithm for the model checking problem. Their method can be considered a special case of our approach.

Baier *et al.* [3] defined a logic asCSL that uses so-called programs as path formulas, i. e. regular expressions over state formulas and actions. Programs can express multiple until formulas of the form $\varphi_1\ U_{[0,b)}\ \varphi_2\ U_{[0,b)} \cdots U_{[0,b)}\ \varphi_k$ because asCSL cannot restrict the duration of individual program phases. The model checking algorithm translates the program to an automaton almost equal to the one in Fig. 2. Our work generalizes the method to multiple until formulas with multiple time bounds.

More recently, Donatelli *et al.* [8] have extended CSL such that path properties can be expressed via a deterministic timed automata (DTA) with a single clock. Chen *et al.* [7] take this approach further and consider DTA specifications with multiple clocks as well.

In principle, one can translate a multiple until formula to a DTA with a single clock. Its basic structure would look similar to Fig. 2, but Donatelli's and Chen's DTAs also include all timing information and would have a size in $\mathcal{O}(k^2)$ – an example construction with $k = 4$ is given in Appendix A. To check whether a CTMC satisfies a DTA specification, they build the product of the two, apply the region construction, and then solve a system of integral equations. Chen's method, applied directly to our specifications, would amount to a complexity in $\mathcal{O}(k^4|S|\lambda c + k^9|S|^3)$, where c is the largest difference between time constraints (roughly comparable to b in Lem. 6.3). Note that our algorithm has only a complexity in $\mathcal{O}(|\mathbf{R}|k \cdot \lambda b)$ if $b = \sup I_{k-1} < \infty$ or $\mathcal{O}(|\mathbf{R}|k \cdot \lambda b + (|S|k)^3)$ otherwise.

8. Conclusion

In this paper we have proposed an effective approximation algorithm for CSL with a multiple until operator. We believe that it is the centerpiece of a broadly applicable full CSL model checker.

The technique we have developed in this paper can also be applied to a subclass of PCTL* formulas. Let $\varphi = f_1\ U_{I_1}\ f_2\ U_{I_2} \ldots f_k$ be a CSL path formula. As we have seen in the paper, in case of $I_1 = \ldots = I_{k-1} = [0, \infty)$, our multiple until formula $f_1\ U_I\ f_2\ U_I \ldots f_k$ corresponds to the LTL formula $f_1\ U\ (f_2\ U\ (\ldots (f_{k-1}\ U\ f_k)\ldots))$. In general, φ is similar to a *step-bounded* LTL formula $\varphi = f_1\ U_{[i_1,j_1]}\ f_2\ U_{[i_2,j_2]} \ldots f_k$ with $i_1, j_1, \ldots$ integers specifying the step bounds. Such step-bounded until LTL formulas can be first transformed into nested *next-state* formulas, for example we have: $f_1\ U_{[2,3]}\ f_2 = f_1 \wedge X(f_1 \wedge X(f_2 \vee (f_1 \wedge X(f_2))))$. The approach we have established in this paper can be adapted slightly to handle this kind of formulas in complexity linear in j_{k-1} (assuming $j_{k-1} < \infty$).

We conclude the paper by noting the connection of our DFA-based approach with the classical *Büchi-automaton*-based LTL model checking algorithm by Vardi and Wolper [21]. The LTL formula φ is first transformed into a Büchi automaton – of exponential size in the worst case – accepting exactly the words satisfying φ. Then, model checking LTL can be reduced to automata-theoretic questions in the product. Instead of Büchi automata accepting infinite runs, we only need DFAs, which is due to the simple form of the multiple until formula: it does not encompass the full expressivity of LTL. This simplification, moreover, allows us to get a DFA whose number of states is only linear in the length of the CSL formula, and the size of the product automaton is then linear in both the size of the CTMC and the length of the CSL formula.

Acknowledgement

Lijun Zhang and Flemming Nielson are partially supported by IDEA4CPS and MT-LAB (a VKR Centre of Excellence). David N. Jansen and Holger Hermanns are partially supported by DFG/NWO Bilateral Research Programme ROCKS and by the European Union Seventh Framework Programme under grant agreement no. ICT-214755 (QUASIMODO). The work of Holger Hermanns has received support by the European Union Seventh Framework Programme under grant agreement no. 295261 (MEALS). We thank Yang Gao and Ming Xu for finding a bug in Thm. 5.1 in an early version of this paper.

References

[1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Verifying continuous time Markov chains. In R. Alur and T. A. Henzinger, editors, *Computer aided verification: … CAV*, volume 1102 of *LNCS*, pages 269–276. Springer, Berlin, 1996.

[2] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model-checking continous-time Markov chains. *ACM trans. comput. log.*, 1(1):162–170, 2000.

[3] C. Baier, L. Cloth, B. R. Haverkort, M. Kuntz, and M. Siegle. Model checking Markov chains with actions and state labels. *IEEE trans. softw. eng.*, 33(4):209–224, 2007.

[4] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In E. A. Emerson and A. P. Sistla, editors, *Computer aided verification: … CAV*, volume 1855 of *LNCS*, pages 358–372, Berlin, 2000. Springer.

[5] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE trans. softw. eng.*, 29(6):524–541, 2003.

[6] P. Ballarini, R. Mardare, and I. Mura. Analysing biochemical oscillation through probabilistic model checking. *Electr. notes theor. comp. sc.*, 229(1):3–19, 2009.

[7] T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Model checking of continuous-time markov chains against timed automata specifications. *Logical Methods in Computer Science*, 7(1), 2011.

[8] S. Donatelli, S. Haddad, and J. Sproston. Model checking timed and stochastic properties with $\mathrm{CSL}^{\mathrm{TA}}$. *IEEE trans. software eng.*, 35(2):224–240, 2009.

[9] B. L. Fox and P. W. Glynn. Computing Poisson probabilities. *Commun. ACM*, 31(4):440–445, 1988.

[10] W. K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In W. J. Stewart, editor, *Numerical solution of Markov chains*, volume 8 of *Probability, pure and applied*, pages 357–371, New York, 1991. Marcel Dekker.

[11] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang. Time-bounded model checking of infinite-state continuous-time Markov chains. *Fundam. inform.*, 95(1):129–155, 2009.

[12] T. A. Henzinger, M. Mateescu, and V. Wolf. Sliding window abstraction for infinite Markov chains. In A. Bouajjani and O. Maler, editors, *Computer aided verification: … CAV*, volume 5643 of *LNCS*, pages 337–352, Berlin, 2009. Springer.

[13] D. N. Jansen. Erratum to: Model-checking continuous-time Markov chains by Aziz et al. http://arxiv.org/abs/1102.2079v1, February 2011.

[14] D. N. Jansen. Understanding Fox and Glynn's "Computing Poisson probabilities". Technical Report ICIS–R11001, Radboud University Nijmegen, February 2011.

[15] J.-P. Katoen, D. Klink, M. Leucker, and V. Wolf. Three-valued abstraction for continuous-time Markov chains. In W. Damm and H. Hermanns, editors, *Computer aided verification: … CAV*, volume 4590 of *LNCS*, pages 311–324, Berlin, 2007. Springer.

[16] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen. The ins and outs of the probabilistic model checker MRMC. *Performance evaluation*, 68(2):90–104, 2011.

[17] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Computer aided verification: … CAV*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[18] P. Panangaden. *Labelled Markov processes*. Imperial College Press, London, 2009.

[19] D. Spieler. Model checking of oscillatory and noisy periodic behavior in Markovian population models. Master's thesis, Saarland University, Saarbrücken, 2009. http://alma.cs.uni-sb.de/data/david/mt.pdf.

[20] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton Univ. Pr., Princeton, N. J., 1994.

[21] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Symposium on logic in computer science*, pages 332–345, Los Alamitos, Calif., 1986. IEEE Comp. Soc..

[22] L. Zhang, D. N. Jansen, F. Nielson, and H. Hermanns. Automata-based CSL model checking. In L. Aceto, M. Henzinger, and J. Sgall, editors, *Automata, languages and programming: … ICALP. Part II*, volume 6756 of *LNCS*, pages 271–282, Berlin, 2011. Springer.

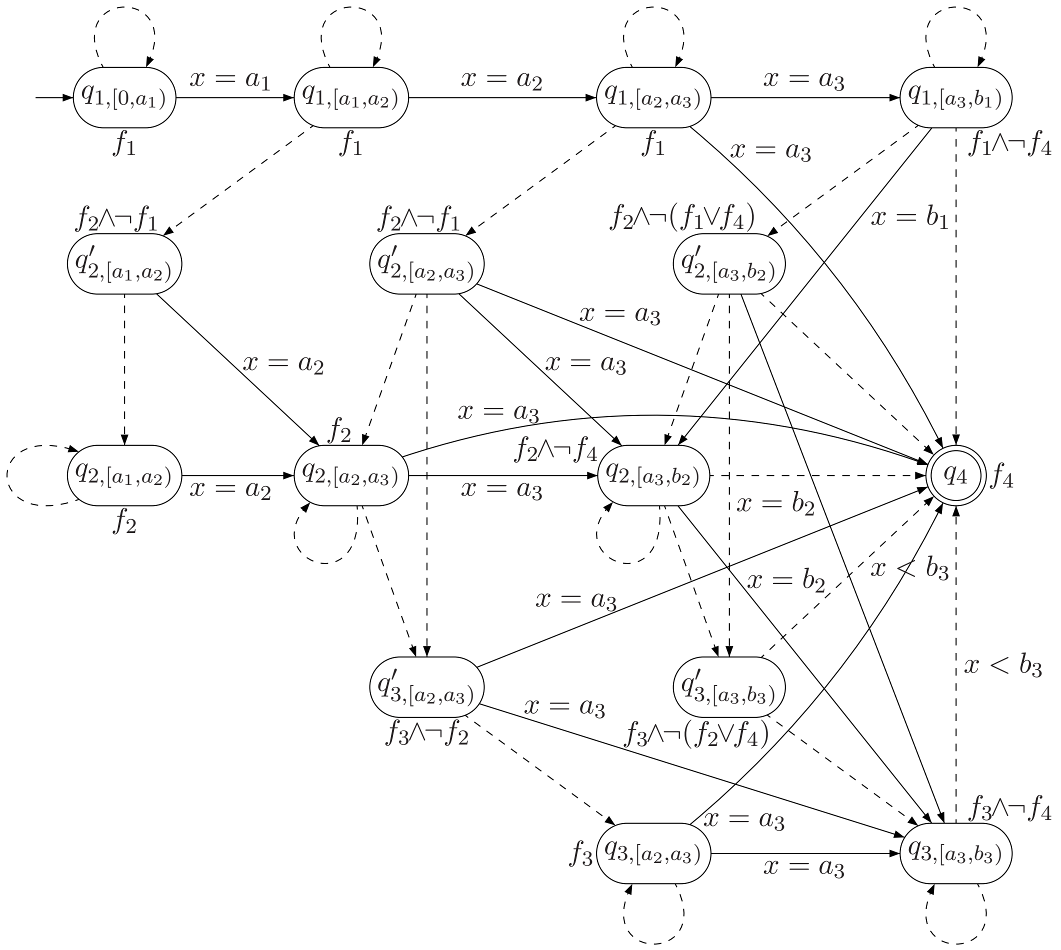


Figure 6: A deterministic CSL$^{\text{TA}}$-timed automaton for $f_1\ U_{[a_1,b_1)}\ f_2\ U_{[a_2,b_2)}\ f_3\ U_{[a_3,b_3)}\ f_4$.

Appendix A. Translating Fig. 2 to a DTA for CSL$^{\text{TA}}$

As mentioned in Section 7, Donatelli *et al.* [8] have extended CSL such that path properties can be expressed via a timed automaton. In Fig. 6, we include a DTA corresponding to the formula $f_1\ U_{[a_1,b_1)}\ f_2\ U_{[a_2,b_2)}\ f_3\ U_{[a_3,b_3)}\ f_4$ with $0 < a_1 < a_2 < a_3 < b_1 < b_2 < b_3$. Dashed lines correspond to transition edges; solid lines to boundary edges. The automaton has a single clock x. The state label $q_{i,I}$ indicates that time t_{i-1} has passed and that the current time is in the interval I. The guard $x < b_3$ is needed in states without boundary edge to ensure that q_4 is not entered too late.

The automaton illustrates that CSL$^{\text{TA}}$ may need a DTA with $\mathcal{O}(k^2)$ states, where k is the number of phases in the multiple until-formula.