Model checking conditional CSL for continuous-time Markov chains

Gao, Yang; Xu, Ming; Zhan, Naijun; Zhang, Lijun

Published in:
Information Processing Letters

Link to article, DOI:
10.1016/j.ipl.2012.09.009

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Information Processing Letters 113 (2013) 44–50

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Model checking conditional CSL for continuous-time Markov chains

Yang Gao [a,1], Ming Xu [b,2], Naijun Zhan [a,1], Lijun Zhang [c,*,3]

[a] State Key Lab. of Comp. Sci., Institute of Software, Chinese Academy of Sciences, China
[b] Department of Computer Science and Technology, East China Normal University, China
[c] Technical University of Denmark, DTU Informatics, Denmark

ARTICLE INFO

Article history:
Received 4 July 2012
Received in revised form 31 August 2012
Accepted 26 September 2012
Available online 28 September 2012
Communicated by J.L. Fiadeiro

Keywords:
Formal methods
Probabilistic systems
Continuous-time Markov chains
Continuous stochastic logic
Conditional logic

ABSTRACT

In this paper, we consider the model-checking problem of continuous-time Markov chains (CTMCs) with respect to conditional logic. To the end, we extend Continuous Stochastic Logic introduced in Aziz et al. (2000) [1] to Conditional Continuous Stochastic Logic (CCSL) by introducing a conditional probabilistic operator. CCSL allows us to express a richer class of properties for CTMCs. Based on a parameterized product obtained from the CTMC and an automaton extracted from a given CCSL formula, we propose an approximate model checking algorithm and analyse its complexity.

1. Introduction

Continuous-time Markov chains (CTMC) have received considerable attentions in network performance analysis, model checking, and system biology. In [1], Continuous Stochastic Logic (CSL) has been introduced, that has been widely used to specify properties over CTMCs.

In the paper [1], Aziz et al. focused on the decidability of the model-checking of CSL. Later, Baier et al. [2] presented an *approximate model checking algorithm* for the case restricted to binary until formulas. Recently, the approximate algorithm has been extended to handle nested until formulas in [3]. The main idea is to exploit the notion of *stratified CTMCs*, which are a subclass of CTMCs that have the nice feature allowing one to obtain the desired probability using a sequence of transient analysis. Then, the product of the CTMC and a deterministic finite automaton (DFA) obtained from the nested until formula is constructed, which is guaranteed to be stratified by construction. The product CTMC can then be analyzed efficiently, in a similar manner as the approach in [2].

In this paper, we propose the conditional continuous stochastic logic (CCSL), an extension of CSL with a conditional probabilistic operator. CCSL allows one to express a richer class of properties, such as:

> *The probability is at least* 0.1, *that the number of proteins is more than* 5 *and the gene becomes inactive within time interval* [10, 20), *under the condition that the proteins increasingly accumulated from* 0 *to k within the same time interval* [10, 20).

Such property can be expressed as a state formula of the form $\mathcal{P}_{\geqslant 0.1}(\Diamond_{[10,20)} f \wedge g \mid f_1\, U_{[10,20)}\, f_2\, U_{[10,20)} \cdots f_k)$ where $f, g, f_1, \ldots, f_k$ are appropriate atomic propositions. We believe that such conditional properties are an important extension because of the important role of conditional probabilities in stochastic models [4].

Essentially, the model checking for the conditional probabilistic operator deals with binary conjunction of CCSL path formulas, which is not allowed in the classical CSL, see [1,2]. Thus, in this paper, we extend the logic CSL

* Corresponding author.
E-mail addresses: gaoy@ios.ac.cn (Y. Gao), mxu@cs.ecnu.edu.cn (M. Xu), znj@ios.ac.cn (N. Zhan), zhang@imm.dtu.dk (L. Zhang).

[1] Supported in part by NSFC projects 91118007 and 60970031.
[2] Supported by NSFC project 11071273.
[3] Supported by IDEA4CPS and MT-LAB (a VKR Centre of Excellence).

0020-0190/$ – see front matter
http://dx.doi.org/10.1016/j.ipl.2012.09.009

with binary conjunction and disjunction operators for the path formulas.

We discuss how to compute the probability of a conjunctive path formula, and then present an approximate model checking algorithm, following the approach in [3]. First, a DFA $\mathcal{A}_\psi$ is constructed for a CCSL path formula ψ. The next step is to construct the DFA $\mathcal{A}_{\bigwedge_i \psi^i}$ for the conjunction from the automata $\mathcal{A}_{\psi^i}$. The first challenging step is to construct the product of the CTMC and the automaton $\mathcal{A}_{\bigwedge_i \psi^i}$. A plain product construction turns out to be insufficient: We have to pay special attention to whether some conjuncts of the formula have been satisfied. We propose a notion of parameterized product construction. The probability is then computed on this product. The size of the automaton could be exponential in the number of binary operators in the path formulas, arising from the product construct, and the approximation calculation for the transient probability distributions is linear in the size of the product.

Related work. There is a rich literature on model checking techniques for CTMCs, see [1,2,5–8]. In [5,6], deterministic timed automata (DTA) are used for specifying path properties. As discussed in [9], nested until CSL path formulas can be expressed in DTA as well, however with a much larger number of states. Real-time is considered in [8], with exponential complexity both in the size of the formula and in the time bound appearing in it. In this paper, we extend CSL path formulas by allowing conjunction and disjunction, then accordingly extend state formulas by introducing the conditional probabilistic operator.

The conditional probabilistic operator is directly inspired by the paper [10], in which the conditional probabilistic operator is introduced and analyzed for Markov decision processes (MDP). Path formulas considered there are restricted binary path operators, and the challenge for MDPs is to study the scheduler class guaranteeing the extreme (maximal or minimal) probabilities.

2. Preliminaries

In this section, we define some basic notions that will be used later. For details please refer to [2]. For convenience, we fix a set of propositions AP in the sequel, ranged by $f_1, f_2, g_1, g_2, \ldots$.

Definition 1. A labeled continuous-time Markov chain (CTMC) is a tuple $\mathcal{C} = (S, \mathbf{R}, L, \alpha)$ where S is a finite set of states, $\alpha : S \to [0,1]$ is the initial distribution satisfying $\sum_{s\in S} \alpha(s) = 1$, $\mathbf{R} : S \times S \to \mathbb{R}_{\geqslant 0}$ is a rate matrix, and $L : S \to 2^{AP}$ is a labeling function.

For $A \subseteq S$, define $\mathbf{R}(s, A) := \sum_{s'\in A} \mathbf{R}(s, s')$. We denote the *exit rate* of s by $E(s) := \mathbf{R}(s, S)$. 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'.

Consider the CTMC in Fig. 1. If s_1 is the current state of the CTMC, the probability that some transition will be triggered within time t is $1 - \mathrm{e}^{-2t}$. Furthermore, there is a competition between the transitions to s_2 and s_3: the probability to take the transition to s_2 is $\frac{\mathbf{R}(s_1,s_2)}{E(s_1)} \cdot (1 - \mathrm{e}^{-2t})$. The labeling function L assigns to each state s a set of atomic propositions $L(s) \subseteq AP$ which are valid in s.

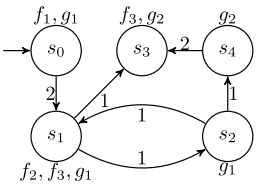

Fig. 1. $\mathcal{C} = (S, \mathbf{R}, L, \alpha)$.

Transient probability. Starting with distribution α of $\mathcal{C}$, the transient probability vector at time t, denoted by $\pi^{\mathcal{C}}(\alpha, t)$, is the probability distribution over states at time t. If $t = 0$, we have $\pi^{\mathcal{C}}(\alpha, 0)(s') = \alpha(s')$. For $t > 0$, the transient probability [11] is given by: $\pi^{\mathcal{C}}(\alpha, t) = \pi^{\mathcal{C}}(\alpha, 0)\mathrm{e}^{\mathbf{Q}t}$ where $\mathbf{Q} := \mathbf{R} - \mathrm{diag}(E)$ is the *generator* matrix and $\mathrm{diag}(E)$ denotes the diagonal matrix with $\mathrm{diag}(E)(s, s) = E(s)$.

Paths and probabilistic measures. A right continuous step function $\rho : \mathbb{R}_{\geqslant 0} \to S$ is called a *step function* (or *an infinite* (*sample*) *path*), where $\rho(t)$ stands for the state at t. For a given step function ρ and $i \in \mathbb{N}$, we denote by $\rho_S[i] = s_i$ the state at the $(i+1)$-th step, and by $\rho_T[i]$ the time spent at $\rho_S[i]$, i.e., the length of the step segment starting with $\rho_S[i]$. Let $Path^{\mathcal{C}}$ denote the set of all infinite paths, and $Path^{\mathcal{C}}(s)$ denote the subset of those paths starting from s.

Let $I_0, \ldots, I_{k-1}$ be nonempty intervals in $\mathbb{R}_{\geqslant 0}$. The *cylinder set* $Cyl(s_0, I_0, s_1, I_1, \ldots, s_{k-1}, I_{k-1}, s_k)$ is defined by:

$$\big\{\rho \in Path^{\mathcal{C}} \;\big|\; \forall 0 \leqslant i \leqslant k.\ \rho_S[i] = s_i \wedge\ \forall 0 \leqslant i < k.\ \rho_T[i] \in I_i\big\}.$$

Let $\mathcal{F}(Path^{\mathcal{C}})$ denote the smallest σ-algebra on $Path^{\mathcal{C}}$ containing all cylinder sets. For initial distribution $\alpha : S \to [0,1]$, a probability measure (denoted $\Pr^{\mathcal{C}}_\alpha$) on this σ-algebra is introduced as follows: $\Pr^{\mathcal{C}}_\alpha$ is the unique measure that satisfies: $\Pr^{\mathcal{C}}_\alpha(Cyl(s))$ equals $\alpha(s)$, and for $k > 0$, $\Pr^{\mathcal{C}}_\alpha(Cyl(s_0, I_0, \ldots, I_{k-1}, s_k))$ equals

$$\Pr^{\mathcal{C}}_\alpha\big(Cyl(s_0, I_0, \ldots, s_{k-1})\big) \cdot \frac{\mathbf{R}(s_{k-1}, s_k)}{E(s_{k-1})} \cdot \eta(I_{k-1}),$$

where $\eta(I_{k-1}) := \mathrm{e}^{-E(s_{k-1})\inf I_{k-1}} - \mathrm{e}^{-E(s_{k-1})\sup I_{k-1}}$ is the probability to take a transition during I_{k-1}. If $\alpha(s) = 1$ for some state $s \in S$, we sometimes simply write $\Pr^{\mathcal{C}}_s$ instead of $\Pr^{\mathcal{C}}_\alpha$. We omit the superscript $\mathcal{C}$ if it is clear from the context.

3. Conditional continuous stochastic logic (CCSL)

This section is devoted to introducing a conditional continuous stochastic logic (CCSL) by extending the *Continuous Stochastic Logic* (CSL) introduced by Aziz et al. [1] with a conditional probabilistic operator. Let I_i be a nonempty left-closed and right-open interval on $\mathbb{R}_{\geqslant 0}$. Let $\bowtie \in \{<, \leqslant, \geqslant, >\}$, $0 \leqslant p \leqslant 1$, and $K > 1$. The syntax of CCSL is defined as:

$$\Phi := f \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathcal{P}_{\bowtie p}(\varphi) \mid \mathcal{P}_{\bowtie p}(\varphi \mid \varphi);$$

$$\varphi := \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Phi_1\, U_{I_1}\, \Phi_2\, U_{I_2} \cdots \Phi_K$$

where $f \in AP$ is an atomic proposition. The syntax of CCSL consists of state formulas and path formulas. We use Φ, Ψ and their indexed versions for state formulas. The path formula $\Phi_1\, U_{I_1}\, \Phi_2\, U_{I_2} \cdots \Phi_K$ with $K > 1$ is referred to as the *atomic path formula*. Obviously, each path formula can be expressed into a *disjunctive normal form* (DNF) $\varphi = \bigvee_i \bigwedge_j \psi^{ij}$ where ψ^{ij} are atomic path formulas. We use ψ for atomic path formulas and φ for general path formulas in DNF.

Let $\mathcal{C} = (S, \mathbf{R}, L, \alpha)$ be a CTMC with $s \in S$. The semantics of CCSL state formulas is standard: $s \models \mathit{true}$ for all $s \in S$, $s \models a$ 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:

$$s \models \mathcal{P}_{\bowtie p}(\varphi) \quad \text{iff} \quad \Pr_s\big(\{\rho \in \mathit{Path} \mid \rho \models \varphi\}\big) \bowtie p,$$

$$s \models \mathcal{P}_{\bowtie p}(\varphi_1 \mid \varphi_2) \quad \text{iff} \quad \frac{\Pr_s(\{\rho \in \mathit{Path} \mid \rho \models \varphi_1 \wedge \varphi_2\})}{\Pr_s(\{\rho \in \mathit{Path} \mid \rho \models \varphi_2\})} \bowtie p$$

where $\Pr_s(\{\rho \in \mathit{Path} \mid \rho \models \varphi\})$, or $\Pr_s(\varphi)$ for short, denotes the probability measure of the set of all paths which start from s and satisfy φ. Similarly, $\Pr_s(\varphi_1 \mid \varphi_2)$ denotes the conditional probability $\frac{\Pr_s(\varphi_1 \wedge \varphi_2)}{\Pr_s(\varphi_2)}$ under the premise $\Pr_s(\varphi_2) \neq 0$.[4]

The semantics for the Boolean operators is standard, and the semantics of the atomic path formula is given by [1,3]:

> $\rho \models \varphi = \Phi_1\, U_{I_1}\, \Phi_2\, U_{I_2} \cdots \Phi_K$ *iff there exist real numbers* $0 \leqslant t_1 \leqslant t_2 \leqslant \cdots \leqslant t_{K-1}$ *such that* $\rho(t_{K-1}) \models \Phi_K$, *and for each integer* $0 < i < K$ *we have* $(t_i \in I_i) \wedge (\forall t' \in [t_{i-1}, t_i).\ \rho(t') \models \Phi_i)$, *where* t_0 *is defined to be* 0 *for notational convenience.*

4. Model checking algorithm for CCSL

In this section, we present an algorithm for checking CCSL properties. We first recall the deterministic finite automaton (DFA) construction for the atomic path formula ψ. Then, we extend the construction to the conjunctive path formula by introducing the notion of a *parameterized product construction* for the given CTMC and the conjunctive path formula. This is the key for computing the probability of the set of paths satisfying the conjunctive path formula. We further show how to compute the probabilities of general path formulas. Finally we describe an algorithm for model checking CCSL and analyze its complexity.

In the rest of the paper, let $\psi^i = f_1^i\, U_{I_1^i}\, f_2^i\, U_{I_2^i} \cdots f_{K_i}^i$ with $i = 1, \ldots, n$ be n special atomic path formulas. For simplicity, as in [3,9] we assume that i) $a_k^i \leqslant a_l^i$ and $b_k^i \leqslant b_l^i$ for any two intervals $I_k^i = [a_k^i, b_k^i)$ and $I_l^i = [a_l^i, b_l^i)$ with $k < l$; ii) all $f_k^i \in AP$ are pairwise distinct for $i = 1, \ldots, n$ and $k = 1, \ldots, K_i$. We will drop the supscript in case $n = 1$.

4.1. Formula automata

In this subsection, we recall how to construct a deterministic finite automaton (DFA) for $\bigwedge_{i=1}^n \psi^i$. Firstly, we consider the simple case when $n = 1$. So, the atomic path formula ψ^i describes the required order of f_1^i-, …, f_K^i-states.

Definition 2 *(Atomic path formula automaton).* (See [3].) The atomic path formula automaton $\mathcal{A}_\psi = (\Sigma, Q, q_{\text{in}}, \delta, F)$ is defined as follows:

- $\Sigma = 2^{\{f_1, \ldots, f_K\}}$.
- $Q = \{q_1, \ldots, q_K, \bot\}$ with $q_{\text{in}} = q_1$ and $F = \{q_1, \ldots, q_K\}$.
- For every $a \in \Sigma$, the transition relation δ is given by $\delta(q_K, a) = q_K$, $\delta(\bot, a) = \bot$, and for the rest $q_i \in Q \setminus \{q_K, \bot\}$,

$$\delta(q_i, a) = \begin{cases} q_j & \text{if } j \geqslant i \wedge f_i, \ldots, f_{j-1} \notin a \wedge f_j \in a; \\ \bot & \text{otherwise.} \end{cases}$$

Both q_K and $\bot$ are absorbing states, i.e., with only transitions leading to themselves. The former state is referred to a *good* absorbing state, the latter a *bad* absorbing state.

The words accepted by $\mathcal{A}_\psi$ are finite traces $w \in \Sigma^*$, such that they can be extended to a trace $ww' \in \Sigma^\omega$ that satisfies the time–abstract (LTL) formula of the form $f_1\, U\, (f_2\, U\, (\cdots (f_{K-1}\, U\, f_K) \cdots))$.

Transitions in $\mathcal{A}_\psi$ go always from lower goal states to higher goal states. The good state q_K implies that any path traversing q_K satisfies the atomic path formula ψ under suitable timing constraint; while the bad state $\bot$ implies that any path traversing $\bot$ refutes the atomic path formula ψ.

Below we define the automaton for the conjunction of several atomic path formulas, which is essentially the product construction.

Definition 3 *(Conjunctive path formula automaton).* Let $\varphi = \bigwedge_{i=1}^n \psi^i$ and $\mathcal{A}_{\psi^i} = (\Sigma^i, Q^i, q_{\text{in}}^i, \delta^i, F^i)$ be the formula automata for ψ^i respectively for $i = 1, \ldots, n$. Then the conjunctive path formula automaton $\mathcal{A}_\varphi = (\Sigma, Q, q_{\text{in}}, \delta, F)$ is defined as follows:

- $\Sigma = 2^{\bigcup_{i=1}^n \{f_1^i, \ldots, f_{K_i}^i\}}$.
- $Q = Q^1 \times \cdots \times Q^n$ with $q_{\text{in}} = (q_{\text{in}}^1, \ldots, q_{\text{in}}^n)$, and $F = F^1 \times \cdots \times F^n$.
- $\delta((q_{k_1}^1, \ldots, q_{k_n}^n), a) = (\delta^1(q_{k_1}^1, a^1), \ldots, \delta^n(q_{k_n}^n, a^n))$, where a^i is the projection of a onto Σ^i.

The state $(q_{k_1}^1, \ldots, q_{k_n}^n)$ is good if all elements $q_{k_i}^i$ are good; it is bad if at least one component $q_{k_i}^i$ is bad. Both good and bad states are absorbing.

[4] If $\Pr_s(\varphi_2) = 0$ for some state s, we say that $\Pr_s(\varphi_1 \mid \varphi_2)$ is undefined for the CTMC $\mathcal{C}$. In the rest of the paper, we assume all the conditional probability formulas are defined in the model checking algorithm. In fact, the proposed techniques of the paper suffice to check whether $\Pr_s(\varphi_2) = 0$.

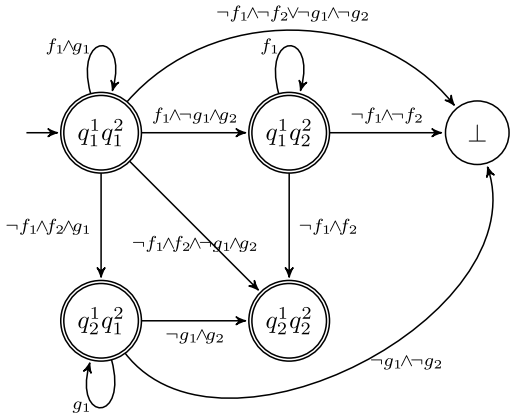

Fig. 2. The automaton $\mathcal{A}_{\psi^1\wedge\psi^2}$.

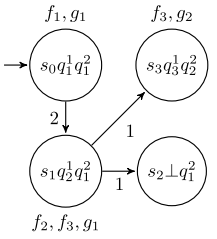

Fig. 3. $\mathcal{C}_{\psi^1\wedge\psi^2}$.

Example 1. In this example we consider the conjunctive path automaton $\mathcal{A}_{\psi^1\wedge\psi^2}$ with $\psi^1 = f_1\ U\ f_2$ and $\psi^2 = g_1\ U\ g_2$ (see Fig. 2). The initial state is (q_1^1, q_1^2), final states are marked with a double circle. The transition labels indicate which subsets of AP are acceptable. For example, we have $\delta((q_1^1, q_1^2), \{f_2, g_1\}) = (q_2^1, q_1^2)$, and (q_2^1, q_2^2) is good. The node labeled with $\perp$ represents five bad absorbing states $(\perp, q_1^2)$, $(\perp, q_2^2)$, $(q_1^1, \perp)$, $(q_2^1, \perp)$ and $(\perp, \perp)$. Transitions out of these two kinds of absorbing states are omitted.

4.2. Product construction

We have defined the conjunctive path formula automaton. Following the approach in [3], the next step would be to construct the product of the CTMC and the automaton. This step turns out to be more involved. Thus we first start with an example illustrating that the plain product does *not* work:

Example 2. Consider the CTMC in Fig. 1, and the conjunction $\psi^1 \wedge \psi^2$ with two atomic path formulas $\psi^1 = f_1\ U_{[0,2)}\ f_2\ U_{[2,3)}\ f_3$ and $\psi^2 = g_1\ U_{[1,3)}\ g_2$. We construct the product from the CTMC and the automaton $\mathcal{A}_{\psi^1\wedge\psi^2}$ in a straightforward way: Its reachable part is shown in Fig. 3. Notice that there is a transition from the state (s_1, q_2^1, q_1^2) to $(s_2, \perp, q_1^2)$, since $\delta((q_2^1, q_1^2), L(s_2)) = (\delta_1(q_2^1, \emptyset), \delta_2(q_1^2, \{g_1\})) = (\perp, q_1^2)$. Let us explain why this plain product is not sufficient for our purpose. The valid path $\rho = s_0, t_0, s_1, t_1, s_3, \ldots$ — assuming timing constraints are satisfied — is captured by this product, but not those paths like $\rho = s_0, t_0, s_1, t_1, s_2, t_2, s_4, \ldots$, since $(s_2, \perp, q_1^2)$ is marked as a bad absorbing state.

The information missing in the product $\mathcal{C}_{\psi^1\wedge\psi^2}$ is whether one of the atomic path formulas ψ^1 (or ψ^2) is already satisfied, and the other still needs to be checked. This motivates the definition of the parameterized product CTMC, in which the parameter identifies such relevant information.

Definition 4 *(Parameterized product).* Let $\mathcal{C} = (S, \mathbf{R}, L, \alpha)$ be a CTMC and $\mathcal{A}_\varphi$ be the formula automaton for $\varphi = \bigwedge_{i=1}^n \psi^i$. Let Λ be a subset of all atomic path formulas ψ^i. Then the parameterized product CTMC $\mathcal{C}_\varphi^\Lambda = (\widehat{S}, \widehat{\mathbf{R}}, \widehat{L}, \widehat{\alpha})$ is defined as follows:

- $\widehat{S} = S \times Q$.
- The rate $\widehat{\mathbf{R}}((s,q),(s',q'))$ equals $\mathbf{R}(s,s')$ if $q = (q_{k_1}^1, \ldots, q_{k_n}^n)$ is not bad, $q' = (q_{k_1'}^1, \ldots, q_{k_n'}^n)$, and for each $i = 1, \ldots, n$,

$$q_{k_i'}^i = \begin{cases} q_{K_i}^i, & \text{if } s' \models f_{K_i}^i \text{ and } \psi^i \in \Lambda; \\ \delta^i(q_{k_i}^i, L(s') \cap \{f_1^i, \ldots, f_{K_i}^i\}), & \text{otherwise.} \end{cases}$$

  All other elements of $\widehat{\mathbf{R}}$ are zero.
- The labeling function $\widehat{L}(s,q)$ is defined in two steps:
  1. If $q = (q_{k_1}^1, \ldots, q_{k_n}^n)$ is not bad, $\widehat{L}(s,q)$ equals $L(s) \cap \bigcup_{i=1}^n \{f_{k_i}^i, \ldots, f_{K_i}^i\}$; $\emptyset$ otherwise.
  2. For $i = 1, \ldots, n$, add the label $f_{K_i}^i$ to those states (s,q) with $q_{k_i}^i = q_{K_i}^i$.
- The initial distribution $\widehat{\alpha} : S \times Q \to [0,1]$ is given by:
  - $\widehat{\alpha}(s,q)$ equals $\alpha(s)$ if for $i = 1, \ldots, n$,

$$q_{k_i}^i = \begin{cases} q_{K_i}^i, & \text{if } s \models f_{K_i}^i \text{ and } \psi^i \in \Lambda; \\ \delta^i(q_{\text{in}}^i, L(s) \cap \{f_1^i, \ldots, f_{K_i}^i\}), & \text{otherwise,} \end{cases}$$

    where $q = (q_{k_1}^1, \ldots, q_{k_n}^n)$.
  - All other elements of $\widehat{\alpha}(s,q)$ are zero.

In the product, each state is of the form $(s, (\ldots, q_{k_i}^i, \ldots))$, in which $q_{k_i}^i$ from the path formula automaton of ψ_i is relevant to the atomic propositions in ψ^i that s satisfies. The parameter Λ is a set of path formulas ψ^i. If a transition in the product leads to a state s' which satisfies f_{K_i} and Λ contains ψ^i, the corresponding component of the state should be marked with the good absorbing state $q_{K_i}^i$ of $\mathcal{A}_{\psi_i}$, i.e., $(s', (\ldots, q_{K_i}^i, \ldots))$. In what follows, we will show that the set Λ plays an important role in keeping track of the path formulas which have been satisfied during the probability computation.

Example 3. The product $\mathcal{C}_{\psi^1\wedge\psi^2}^\emptyset$ is shown in Fig. 1. Since both ψ^1 and ψ^2 can be satisfied during $[2,3)$, we need to construct $\mathcal{C}_{\psi^1\wedge\psi^2}^{\{\psi^1,\psi^2\}}$. Its reachable part is shown in Fig. 4. Note that the previous state (s_1, q_2^1, q_1^2) is now renamed to (s_1, q_3^1, q_1^2) according to the initial distribution. It is easy to

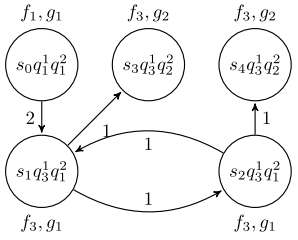

Fig. 4. The product $\mathcal{C}^{\{\psi^1,\psi^2\}}_{\psi^1\wedge\psi^2}$.

see that the path $\rho = s_0, t_0, s_1, t_1, s_2, t_2, s_4, \ldots$, which was recognized as a bad path in Example 2, is captured by the new parameterized product CTMC during $[2, 3)$ (assuming timing constraints are satisfied).

4.3. Probability computation

As in [3,9], the product CTMC *stratifies* the original CTMC in the sense that (time–abstract) bad paths will be uniformly directed towards the bad states. This allows us to reduce the computation by standard transient probability computation for CTMCs, which will be discussed in this section.

We fix a conjunctive path formula $\varphi = \bigwedge_{i=1}^n \psi^i$ together with a CTMC $\mathcal{C} = (S, \mathbf{R}, L, \alpha)$. Now we focus on how to compute the probability of such a path formula starting from an arbitrary initial distribution α in a forward way. We first introduce some notation for convenience:

- For an interval I and a positive number h, let $I \ominus h$ denote the set $\{t - h \mid t \in I \wedge t \geqslant h\}$, and let $\psi \ominus h$ denote the formula $f_1\, U_{I_1\ominus h}\, f_2\, U_{I_2\ominus h} \cdots f_K$.
- For $1 \leqslant i \leqslant j \leqslant K$, define $f_{i\ldots j} := \bigvee_{k=i}^{j} f_k$.
- For $1 \leqslant i < K$, define $f_{[i]} := f_i\, U_{I_i}\, f_{i+1}\, U_{I_{i+1}} \cdots f_K$.
- For a state formula Φ, let $\mathcal{C}[\Phi]$ be derived from $\mathcal{C}$ by making its states that satisfy Φ absorbing.

Definition 5 *(Indicator matrix).* Given a state formula Φ and a subset Λ of $\{\psi^i \mid i = 1, \ldots, n\}$, the indicator matrix $\mathbf{I}^{\Lambda}_{\Phi}$ is defined by:

- If $(s, (q^1_{k_1}, \ldots, q^n_{k_n})) \models \Phi$, $\mathbf{I}^{\Lambda}_{\Phi}((s, (q^1_{k_1}, \ldots, q^n_{k_n})), (s, (q^1_{k'_1}, \ldots, q^n_{k'_n}))) = 1$, where $q^i_{k'_i}$ is $q^i_{K_i}$ if $\psi^i \in \Lambda$; $q^i_{k_i}$ otherwise.
- All other entries of $\mathbf{I}^{\Lambda}_{\Phi}$ are zero.

Now we show how to compute the probability $\Pr^{\mathcal{C}}_{\alpha}(\bigwedge_{i=1}^n \psi^i)$ in a forward way.

Theorem 1 *(Probability computation). Given a conjunctive path formula $\varphi = \bigwedge_{i=1}^n \psi^i$ and a CTMC $\mathcal{C} = (S, \mathbf{R}, L, \alpha)$ equipped with the following notations*:

- *Let h be the least nonzero endpoint of all intervals I^i_k occurring in φ, and let Λ be the set $\{\psi^i \mid h > a^i_{K_i-1}\}$.*
- *If $h < \max_{i=1}^n\{b^i_{K_i-1}\}$, let h' be the least endpoint of all intervals I^i_k greater than h, and let Λ' be the set $\{\psi^i \mid h' > a^i_{K_i-1}\}$.*
- *For $i = 1, \ldots, n$, let $a^i_0 = 0$, $b^i_0 = a^1_1$, $a^i_{K_i} = b^i_{K_i-1}$, $b^i_{K_i} = \infty$ and $f^i_{K_i+1} = f^i_{K_i}$, and let k_i, l_i be the unique indices such that $b^i_{k_i-1} \leqslant h < b^i_{k_i}$ and $a^i_{l_i-1} < h \leqslant a^i_{l_i}$.*

The probability $\Pr^{\mathcal{C}}_{\alpha}(\varphi) = \Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{\widehat{\alpha}}(\varphi)$ is computed as follows:

1. *If $h < \max_{i=1}^n\{b^i_{K_i-1}\}$, then*

$$\Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{\widehat{\alpha}}(\varphi) = \pi^{\mathcal{C}^{\Lambda}_{\varphi}[\neg \bigwedge_{i=1}^n f^i_{1\ldots l_i}]}(\widehat{\alpha}, h) \cdot \mathbf{I}^{\Lambda'}_{\bigwedge_{i=1}^n f^i_{k_i\ldots l_i}} \cdot \Pr^{\mathcal{C}^{\Lambda'}_{\varphi}}_{(\cdot)}\left(\bigwedge_{i=1}^n f^i_{[k_i]} \ominus h\right), \tag{1}$$

   where $\Pr^{\mathcal{C}}_{(\cdot)}(\varphi)$ stands for the vector $(\Pr^{\mathcal{C}}_s(\varphi))_{s\in S}$.
2. *Otherwise $h = \max_{i=1}^n\{b^i_{K_i-1}\}$, then*

$$\Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{\widehat{\alpha}}(\varphi) = \pi^{\mathcal{C}^{\Lambda}_{\varphi}}(\widehat{\alpha}, h) \cdot \mathbf{I}^{\Lambda}_{\bigwedge_{i=1}^n f^i_{K_i}} \cdot (1, \ldots, 1)^{\mathrm{T}}. \tag{2}$$

The proof is given in Appendix A for completeness, which follows the same idea as the proof in [9]. Intuitively, the computation is performed by traversing through the time intervals in a forward way in the product CTMC. The time is partitioned into finitely many intervals using endpoints appearing in the formula. With the initial distribution $\widehat{\alpha}$, we compute the probability distribution of all states at the time point h for the first interval $[0, h)$, which determines the parameter Λ. The indicator matrix filters out all paths dissatisfying φ at the time point h. The parameter Λ' is determined by the next interval $[h, h')$. We recursively compute the probability distribution at the time point h', and repeat this until the last time point.

The parameter Λ will be repeatedly adjusted when we push the time forward. We illustrate the theorem by computing $\Pr_{s_0}(\psi^1 \wedge \psi^2)$ in the following example.

Example 4. Let $\psi^1 = f_1\, U_{[0,2)}\, f_2\, U_{[2,3)}\, f_3$ and $\psi^2 = g_1\, U_{[1,3)}\, g_2$. We compute $\Pr_{s_0}(\psi^1 \wedge \psi^2)$ in three phases.

Initially, $a^2_1 = 1$ and $b^1_1 = a^1_2 = 2$ are the first two least nonzero endpoints, so $h = 1$ and $h' = 2$. Accordingly, $\Lambda = \emptyset$ and $\Lambda' = \{\psi^2\}$. By Theorem 1, $k_1 = 1$, $l_1 = 2$, $k_2 = 1$ and $l_2 = 1$, which means we should pick out the states satisfying $f_1 \vee f_2$ and g_1 at time point h. Then, we compute $\pi^{\mathcal{C}^{\emptyset}_{\psi^1\wedge\psi^2}[\neg(f_1\vee f_2\wedge g_1)]}(\alpha, h)$ – the probability distribution at time point $h = 1$ w.r.t. the product CTMC equipped with parameter $\emptyset$ showed in Fig. 3. The indicator matrix has only two nonzero entries $\mathbf{I}^{\{\psi^2\}}_{f_{1\ldots2}\wedge g_{1\ldots1}}((s_0, q^1_1, q^2_1), (s_0, q^1_1, q^2_1)) = 1$ and $\mathbf{I}^{\{\psi^2\}}_{f_{1\ldots2}\wedge g_{1\ldots1}}((s_1, q^1_2, q^2_1), (s_1, q^1_2, q^2_1)) = 1$. After this step, we push forward the time, so ψ^1 and ψ^2 become $f_1\, U_{[0,1)}\, f_2\, U_{[1,2)}\, f_3$ and $g_1\, U_{[0,2)}\, g_2$ respectively.

In the next phase, $h = 1$, $h' = 2$, accordingly, $\Lambda = \{\psi^2\}$ and $\Lambda' = \{\psi^1, \psi^2\}$. Then, we compute the probability distribution at $h = 1$ w.r.t. the same CTMC in Fig. 3. At the next phase, ψ^1 and ψ^2 may be fully satisfied, we should relocate the state (s_1, q^1_2, q^2_1) to (s_1, q^1_3, q^2_1), so the indicator matrix has only one nonzero entry

$\mathbf{I}^{\{\psi^1,\psi^2\}}_{f_{2\ldots2}\wedge g_{1\ldots2}}((s_1,q_2^1,q_1^2),(s_1,q_3^1,q_1^2))=1$. Now the formulas become $f_2\,U_{[0,1)}\,f_3$ and $g_1\,U_{[0,1)}\,g_2$.

In the last phase, $h=\max\{b^1_{K_1-1},b^2_{K_2-1}\}=1$, $\Lambda=\{\psi^1,\psi^2\}$, and we get to the end of the computation. Then, we compute the probability distribution at $h=1$ w.r.t. the CTMC in Fig. 4. Now the indicator matrix has only two nonzero entries $\mathbf{I}^{\{\psi^1,\psi^2\}}_{f_3\wedge g_2}((s_3,q_3^1,q_2^2),(s_3,q_3^1,q_2^2))=1$ and $\mathbf{I}^{\{\psi^1,\psi^2\}}_{f_3\wedge g_2}((s_4,q_3^1,q_2^2),(s_4,q_3^1,q_2^2))=1$.

The total computational results are

t	0	1	2	3
(s_0,q_1^1,q_1^2)	1	e^{-2}	0	0
(s_1,q_2^1,q_1^2)	0	$2\mathrm{e}^{-2}$	0	0
(s_1,q_3^1,q_1^2)	0	0	$4\mathrm{e}^{-4}$	0
(s_3,q_3^1,q_2^2)	0	0	0	$\frac{8}{3}\mathrm{e}^{-4}-2\mathrm{e}^{-5}-\frac{2}{3}\mathrm{e}^{-7}$
(s_4,q_3^1,q_2^2)	0	0	0	$\frac{4}{3}\mathrm{e}^{-4}-2\mathrm{e}^{-5}+\frac{2}{3}\mathrm{e}^{-7}$

Therefore, we have $\Pr_{s_0}(\psi^1\wedge\psi^2)=4\mathrm{e}^{-4}-4\mathrm{e}^{-5}$ by collecting all desired probabilities.

Corollary 1. *Given a path formula $\varphi=\bigvee_{i=1}^m\bigwedge_j\psi^{ij}$ in DNF and a CTMC $\mathcal{C}$, the probability $\Pr^{\mathcal{C}}_{\alpha}(\varphi)$ can be computed by inclusion–exclusion principle.*

4.4. Model checking algorithm and its complexity

Let $\mathcal{C}=(S,\mathbf{R},L,\alpha)$ be a CTMC, $s\in S$, and Φ be a CCSL state formula. The model checking problem is to check whether $s\models\Phi$. 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 CCSL, the cases where Ψ is an atomic proposition, a negation or a conjunction are standard as for CTL. The case when Ψ is a (conditional) probabilistic formula is the challenging part. The model checking algorithm for $\mathcal{P}_{\bowtie p}(\psi)$ has been discussed in [1,9], thus below we discuss the case of $\Pr_s(\varphi\mid\psi)$.

Let $\Psi=\mathcal{P}_{\bowtie p}(\varphi\mid\psi)$ with $\varphi=\Phi_1\,U_{I_1}\,\Phi_2\,U_{I_2}\ldots\Phi_k$ and $\psi=\Psi_1\,U_{J_1}\,\Psi_2\,U_{J_2}\ldots\Psi_l$. By definition, checking Ψ is equivalent to checking whether $\Pr_s(\varphi\mid\psi)\bowtie p$, i.e., whether the quotient of $\Pr_s(\varphi\wedge\psi)$ and $\Pr_s(\psi)$ meets the bound $\bowtie p$. Now we focus on the conjunction part. Assume that the sets $Sat(\Phi_i)$ and $Sat(\Psi_j)$ have been calculated recursively. We replace[5] $\Phi_1,\ldots,\Phi_k$ and $\Psi_1,\ldots,\Psi_l$ by fresh (pairwise disjoint) atomic propositions $f_1,\ldots,f_k$ and $g_1,\ldots,g_l$, and add the label f_i (resp. g_j) to the state s if $s\in Sat(\Phi_i)$ (resp. $s\in Sat(\Psi_j)$). Thus, after applying Theorem 1 and Corollary 1 a finite number of times, $\Pr_s(\varphi\mid\psi)$ is reduced to a product of transient probabilities. We can now apply the results in [1] as follows: By definition, $\Pr_s(\varphi\mid\psi)$ can be expressed as a quotient of finite sum of the form $\sum_k\eta_k\mathrm{e}^{\gamma_k}$ (with algebraic η_k and γ_k). Aziz et al. proved that it is decidable whether such an expression is $\bowtie p$, for $p\in\mathbb{Q}$, which implies directly the decidability of the model checking problem of CCSL.

Finally we discuss the complexity of the approach for approximating $\Pr_s(\varphi)$. The size of the product CTMC is exactly the product of the sizes of the original CTMC and the automaton obtained from the conjunctive path formula, i.e. $\|\mathcal{C}\|\cdot\prod_i\|\psi^i\|$. Then, the usual numerical algorithm can be used to approximate the transient distributions, for instance via *uniformization* [11], or *Runge–Kutta* method, which is linear in the size of the product, the largest exit rate and the largest finite time bounds. Hence, the complexity for computing the probability of the conjunctive path formula is linear in the product of the sizes of the product CTMC and the conjunctive path formula, i.e. $\mathcal{O}([\sum_i\|\psi^i\|]\cdot\|\mathcal{C}\|\cdot[\prod_i\|\psi^i\|])$. Furthermore, the complexity for computing the probability of the path formula in DNF as in Corollary 1 is bounded by $\mathcal{O}([\sum_{ij}\|\psi^{ij}\|]\cdot\|\mathcal{C}\|\cdot[\prod_{ij}\|\psi^{ij}\|])$. It is also the worst case complexity of our model checking algorithm.

Appendix A. Proof of Theorem 1

The equation $\Pr^{\mathcal{C}}_{\alpha}(\varphi)=\Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{\widehat{\alpha}}(\varphi)$ can be proven by establishing mapping of the cylinder sets, similar as the proof in [9] for atomic path formula. We provide the proof of Eq. (1), by extending the proof in [9] for the parametrized product CTMC. For $s'\in\widehat{S}$, define the event $Z(s'):=\{\sigma\mid\sigma@h=s'\wedge\forall t\in[0,h).\ \sigma@t\models\bigwedge_{i=1}^n f^i_{1\ldots l_i}\}$, where $\sigma@h$ stands for the state of the path σ at time h. The following inclusion holds:

$$\{\sigma\mid\sigma\models\varphi\}\subseteq\bigcup_{s'\models\bigwedge_{i=1}^n f^i_{k_i\ldots l_i}}Z(s').$$

Note this property holds for the product CTMC, but not for general CTMCs. Intuitively, the deterministic automaton stratifies the original CTMC in a way, such that those paths σ with $\sigma\not\models\varphi$ will be directed to the bad absorbing state $\perp$. This is the crucial property allowing us to perform a forward transient analysis. The formal argument is done using the notion of stratification, and we refer to [3,9] for details.

Now we fix first $\widehat{\alpha}_s$ as an initial distribution with $\widehat{\alpha}_s=1$ and $s\models\bigwedge_{i=1}^n f^i_{1\ldots l_i}$. By the law of total probability, we have

$$\Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{s}(\varphi)=\sum_{s'\models\bigwedge_{i=1}^n f^i_{k_i\ldots l_i}}\Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{s}\big(Z(s')\big)\cdot\Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{s}\big(\varphi\mid Z(s')\big).$$

By definition of $Z(s')$, we have

$$\Pr^{\mathcal{C}^{\Lambda}_{\varphi}}_{s}\big(Z(s')\big)=\pi^{\mathcal{C}^{\Lambda}_{\varphi}[\neg(\bigwedge_{i=1}^n f^i_{1\ldots l_i})]}(s,h)(s').$$

Now let $\sigma\in Z(s')$ be a path. $\sigma\models\varphi$ implies that at time h, σ has reached a state in a phase from $q^1_{k_1},\ldots,q^1_{l_1},q^2_{k_2},\ldots,q^2_{l_2},\ldots,q^n_{k_n},\ldots,q^n_{l_n}$. So the suffix path of σ starting at time h satisfies $\bigwedge_{i=1}^n f^i_{[k_i]}\ominus h$. From the time point h, the labels $f^1_{1\ldots k_1-1},f^2_{1\ldots k_2-1},\ldots,f^n_{1\ldots k_n-1}$ have been irrelevant for checking the formula φ. Thus we could reconstruct the product CTMC with parameter Λ' for the next phase and put forward the formula. By the Markov property of CTMCs, we have

[5] If any state subformula is undefined, our algorithm will then report undefined.

$$\Pr_s^{\mathcal{C}_\varphi^\Lambda}(\varphi) = \sum_{s' \models \bigwedge_{i=1}^n f^i_{k_i \ldots l_i}} \pi^{\mathcal{C}_\varphi^\Lambda[\neg(\bigwedge_{i=1}^n f^i_{1\ldots l_i})]}(s,h)(s') \cdot \Pr_{s'}^{\mathcal{C}_\varphi^{\Lambda'}}\left(\bigwedge_{i=1}^n f^i_{[k_i]} \ominus h\right) = \sum_{s' \in S} \pi^{\mathcal{C}_\varphi^\Lambda[\neg(\bigwedge_{i=1}^n f^i_{1\ldots l_i})]}(s,h)(s') \cdot \mathbf{1}^{\Lambda'}_{s' \models \bigwedge_{i=1}^n f^i_{k_i \ldots l_i}} \cdot \Pr_{s'}^{\mathcal{C}_\varphi^{\Lambda'}}\left(\bigwedge_{i=1}^n f^i_{[k_i]} \ominus h\right).$$

Hence Eq. (1) holds by $\Pr_{\widehat{\alpha}}^{\mathcal{C}_\varphi^\Lambda}(\varphi) = \sum_{s \in \widehat{S}} \widehat{\alpha}(s) \Pr_s^{\mathcal{C}_\varphi^\Lambda}(\varphi)$.

At last, we prove Eq. (2). For $s' \in \widehat{S}$, define the event $Z(s') := \{\sigma \mid \sigma@h = s' \wedge \forall t \in [0,h).\ \sigma@t \models \bigwedge_{i=1}^n f^i_{1\ldots K_i}\}$. Again, in the stratified product it can be shown that $\{\sigma \mid \sigma \models \varphi\} \subseteq \bigcup_{s' \models \bigwedge_{i=1}^n f^i_{K_i}} Z(s')$. Fix first $\widehat{\alpha}_s$ as an initial distribution with $\widehat{\alpha}_s = 1$ and $s \models \bigwedge_{i=1}^n f^i_{1\ldots K_i}$. By the law of total probability, we have

$$\Pr_s^{\mathcal{C}_\varphi^\Lambda}(\varphi) = \sum_{s' \models \bigwedge_{i=1}^n f^i_{K_i}} \Pr_s^{\mathcal{C}_\varphi^\Lambda}\big(Z(s')\big) \cdot \Pr_s^{\mathcal{C}_\varphi^\Lambda}\big(\varphi \mid Z(s')\big).$$

By definition of $Z(s')$, $\Pr_s^{\mathcal{C}_\varphi^\Lambda}(Z(s')) = \pi^{\mathcal{C}_\varphi^\Lambda}(s,h)(s')$ holds. Thus,

$$\Pr_s^{\mathcal{C}_\varphi^\Lambda}(\varphi) = \sum_{s' \models \bigwedge_{i=1}^n f^i_{K_i}} \pi^{\mathcal{C}_\varphi^\Lambda}(s,h)(s') \cdot \Pr_s^{\mathcal{C}_\varphi^\Lambda}\big(\varphi \mid Z(s')\big).$$

Now let $\sigma \in Z(s')$. We consider the two following cases.

- If $h = \max_i\{b^i_{K_i-1}\} < \infty$, then $\sigma \models \varphi$ implies that at time h, σ has reached a state labeled with $f^1_{K_1}, \ldots, f^n_{K_n}$. This state is good in the product CTMC. So

  $$\Pr_s^{\mathcal{C}_\varphi^\Lambda}(\varphi) = \sum_{s' \models \bigwedge_{i=1}^n f^i_{K_i}} \pi^{\mathcal{C}_\varphi^\Lambda}(s,h)(s').$$

  It requires that the probability on the states labeled with $f^1_{K_1}, \ldots, f^n_{K_n}$ should be added. So

  $$\Pr_s^{\mathcal{C}_\varphi^\Lambda}(\varphi) = \sum_{s' \in S} \pi^{\mathcal{C}_\varphi^\Lambda}(s,h)(s') \cdot \mathbf{1}^{\Lambda}_{s' \models \bigwedge_{i=1}^n f^i_{K_i}} \cdot 1.$$

  Eq. (2) for general initial distribution $\widehat{\alpha}$ follows as the step 2 of this proof.

- Otherwise $h = \max_i\{b^i_{K_i-1}\} = \infty$, then $\Pr_s^{\mathcal{C}_\varphi^\Lambda}(\varphi)$ is the probability to reach the states labeled with $f^1_{K_1}, f^2_{K_2}, \ldots, f^n_{K_n}$ eventually. So we just need to pass the h to ∞ to obtain the probability. □

References

[1] A. Aziz, K. Sanwal, V. Singhal, R. Brayton, Model-checking continuous-time Markov chains, ACM Trans. Comput. Log. 1 (1) (2000) 162–170.

[2] C. Baier, B. Haverkort, H. Hermanns, J.P. Katoen, Model-checking algorithms for continuous-time Markov chains, IEEE Trans. Softw. Eng. 29 (6) (2003) 524–541.

[3] L. Zhang, D.N. Jansen, F. Nielson, H. Hermanns, Automata-based CSL model checking, in: 38th International Colloquium on International Colloquium on Automata, Languages and Programming (ICALP), Part II, in: LNCS, vol. 6756, Springer, 2011, pp. 271–282.

[4] C. Langmead, Generalized queries and bayesian statistical model checking in dynamic bayesian networks: Application to personalized medicine, in: 8th Annual International Conference on Computational Systems Bioinformatics (CSB), Life Sciences Society, 2009, pp. 201–212.

[5] S. Donatelli, S. Haddad, J. Sproston, Model checking timed and stochastic properties with CSL^{TA}, IEEE Trans. Softw. Eng. 35 (2) (2009) 224–240.

[6] T. Chen, T. Han, J.P. Katoen, A. Mereacre, Model checking of continuous-time Markov chains against timed automata specifications, Logical Methods in Computer Science 7 (1) (2011) 1–34, Paper No. 12.

[7] C. Baier, B.R. Haverkort, H. Hermanns, J.P. Katoen, Performance evaluation and model checking join forces, Commun. ACM 53 (9) (2010) 76–85.

[8] T. Chen, M. Diciolla, M. Kwiatkowska, A. Mereacre, Time-bounded verification of CTMCs against real-time specifications, in: 9th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS), in: LNCS, vol. 6919, Springer, 2011, pp. 26–42.

[9] L. Zhang, D.N. Jansen, F. Nielson, H. Hermanns, Efficient CSL model checking using stratification, Logical Methods in Computer Science 8 (2) (2012) 1–18, Paper No. 17.

[10] M.E. Andrés, P. van Rossum, Conditional probabilities over probabilistic and nondeterministic systems, in: 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), in: LNCS, vol. 4963, Springer, 2008, pp. 157–172.

[11] W.J. Stewart, Introduction to the Numerical Solution of Markov Chains, Princeton University Press, Princeton, NJ, 1994.