Model Checking CSL for Markov Population Models *

David Spieler
Modeling and Simulation Group
Computer Science Department
Saarland University
Saarbrücken, Germany
spieler@cs.uni-saarland.de

Ernst Moritz Hahn
State Key Laboratory of Computer Science
Institute of Software
Chinese Academy of Sciences
Beijing, China
hahn@ios.ac.cn

Lijun Zhang
zhanglj@ios.ac.cn

Markov population models (MPMs) are a widely used modelling formalism in the area of computational biology and related areas. The semantics of a MPM is an *infinite-state* continuous-time Markov chain. In this paper, we use the established *continuous stochastic logic* (CSL) to express properties of Markov population models. This allows us to express important measures of biological systems, such as probabilistic reachability, survivability, oscillations, switching times between attractor regions, and various others. Because of the infinite state space, available analysis techniques only apply to a very restricted subset of CSL properties. We present a full algorithm for model checking CSL for MPMs, and provide experimental evidence showing that our method is effective.

1 Introduction

In the context of continuous-time Markov chains (CTMCs), properties of interest can be specified using continuous stochastic logic (CSL) [2, 3]. CSL is a branching-time temporal logic inspired by CTL [9]. It allows to reason about properties of states (state labels), like the number of certain molecules given in this state, about what may happen in the next state (next operator), what may happen within a certain time (bounded until), what may finally happen (unbounded until) or about the long-run average behavior of a model (steady state). Because the underlying semantics of a Markov population model is given as a CTMC, we can also use CSL to reason about properties of such models, when interpreting CSL formulae on the CTMC semantics.

We consider the complete set of CSL formulae, including the steady-state operator and in certain cases also the unbounded until operator [3]. The resulting logic can express (nested) probabilistic properties such as *"the long-run probability is at least 0.4 that we reach* Ψ*-states along* Φ*-states within time interval [6.5,8.5] with a probability larger than* 0.98*"* via $\mathscr{S}_{\geq 0.4}(\mathscr{P}_{>0.98}(\Phi\, \mathscr{U}^{[6.5,8.5]}\, \Psi))$. Using CSL, we can express many measures important for biological models, including oscillation [4].

Previous works [11, 20, 21] have already considered techniques for the transient analysis of infinite-state CTMCs. These techniques are based on *truncation*. This means that only a finite relevant subset of the states of the infinite CTMC is taken into account. The extent to which these models are explored depends on the rates occurring there, as well as on the time bound of the transient analysis. Using truncation, techniques for the analysis of finite CTMCs [24] can be used for the analysis of properties of infinite-state models.

*This research has also been partially funded by the German Research Council (DFG) as part of the Cluster of Excellence on Multimodal Computing and Interaction at Saarland University and the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS), by the National Natural Science Foundation of China (NSFC) under grant No. 61361136002, 61350110518, and the Chinese Academy of Sciences Fellowship for International Young Scientists (Grant No. 2013Y1GB0006).

N. Bertand and L. Bortolussi (Eds.): QAPL 2014
EPTCS 154, 2014, pp. 93–107, doi:10.4204/EPTCS.154.7

$\sqcup$	$\bot$	?	$\top$
$\bot$	$\bot$	?	$\top$
?	?	?	$\top$
$\top$	$\top$	$\top$	$\top$

$\sqcap$	$\bot$	?	$\top$
$\bot$	$\bot$	$\bot$	$\bot$
?	$\bot$	?	?
$\top$	$\bot$	?	$\top$

	C
$\bot$	$\top$
?	?
$\top$	$\bot$

Table 1: Truth values of ternary logic operations.

In a previous publication [15], we have extended these results such that we were able to do approximate model checking for a subset of CSL. This subset excluded the steady-state operator as well as the unbounded until operator. We have implemented these techniques in the model checker INFAMY [14]. On the other hand, we recently developed means to find subsets of states of Markov population model CTMCs which contain the relevant steady-state probability mass [7]. For each of these states, we also obtain lower and upper bounds of the steady state probability. These techniques have been implemented in the tool GEOBOUND [23].

In this paper, we combine GEOBOUND and INFAMY, such that we can also handle the CSL steady-state operator. In addition, we introduce advanced truncation techniques which allow us to explore the model in a more advanced way, leading to a smaller number of states being necessary to check properties. By also taking into account not only the time bound of CSL properties, but also the atomic propositions, we can further restrict the state space to be explored. In certain cases, this also allows us to handle the unbounded until operator. Using a ternary logic [16, 17], in contrast to previous publications, we can compute safe lower and upper bounds for probabilities. In turn, we can decide exactly whether a certain formula holds, does not hold or whether this cannot be decided on the finite truncation of the current model. Apart from this, we have also made some technical improvements, applying for instance to the portability and robustness of INFAMY. We show the applicability of the approach on a number of biological models.

Organization of the paper We give background on Markov population models, CTMCs and CSL in Section 2, and also recall the established CSL model checking algorithm for finite CTMCs. In Section 3, we give the main contribution of the paper, CSL model checking for infinite CTMCs. Section 4 reports experimental results. Section 5 gives related work and Section 6 concludes.

2 Preliminaries

2.1 Ternary Logic

We consider a ternary logic [16, 17] with values $\mathbb{B}_3 := \{\top, \bot, ?\}$. With the ordering $\bot < ? < \top$, $\mathbb{B}_3$ forms a complete lattice. We interpret $\sqcap$ as the meet ("and" operator), and $\cdot^c$ as the complement ("not") operation, with the usual definitions. Other operators like $\sqcup$ ("or" operator) can be derived. Then, $\top$ and $\bot$ can be interpreted as values definitely true or false respectively, and ? is interpreted as an unknown value. We give an overview of the truth values in Table 1. Consider a formula over a number of values some of which are ?. If the value of this formula is different from ?, we know that when inserting $\bot$ or $\top$ instead of some of these values, the result would still be the same. This way, in some cases we can obtain a truth value even though we do not known the truth values of some formula parts.

2.2 Markov Chains

We define some basics of Markov chains and CSL. Let AP denote a set of atomic propositions. For a countable set S, a distribution μ over S is a function $\mu : S \to [0,1]$ with $\mu(S) = 1$, and we define $\mu(A) := \sum_{s \in A} \mu(s)$ for $A \subseteq S$.

Definition 1 *A* labelled continuous-time Markov chain (CTMC) *is a tuple* $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ *where* S *is a countable set of states,* s_{init} *is an initial state,* $L : (S \times AP) \to \mathbb{B}_3$ *is a labelling function, and* $\mathbf{R} : (S \times S) \to \mathbb{R}_{\geq 0}$ *is the* rate matrix.

In the stochastic process which the CTMC represents, s_{init} is the state we start in. The labelling function tells us for each state s and atomic proposition a whether a holds in s, does not hold in s, or whether this is not known or not specified. We say that the rate matrix is *rate-bounded* if the supremum $\sup_{s \in S} \mathbf{R}(s,S)$ is finite, otherwise, it is called *rate-unbounded*. If $\mathbf{R}(s,s') > 0$, we say that there is a transition from s to s'. For $s \in S$, let $\text{post}(s) := \{s' \mid \mathbf{R}(s,s') > 0\}$ denote the set of successor states of s. A state s is called *absorbing* if $\text{post}(s) = \emptyset$. A CTMC is *finitely branching* if for each state s the set of successors $\text{post}(s)$ is finite. In this paper, we consider rate-unbounded, finitely-branching CTMCs which do not explode [1, 8]. Roughly speaking, if the CTMC explodes, then there is a positive, non-zero probability of infinitely many jumps in finite time. On the contrary, the fact that a CTMC does not explode implies that in finite time with probability one only a finite number of states can be reached.

Transition probabilities in CTMCs are exponentially distributed over time. The probability that an arbitrary transition is triggered within time t is given by $1 - e^{-\mathbf{R}(s,S)t}$, where $\mathbf{R}(s,A) := \sum_{s' \in A} \mathbf{R}(s,s')$ for $A \subseteq S$. The probability of taking a particular transition from s to s' within time t is $\frac{\mathbf{R}(s,s')}{\mathbf{R}(s,S)} \left(1 - e^{-\mathbf{R}(s,S)t}\right)$. State s' is reachable from s if there exists a sequence of states $s_1, \ldots, s_n$ with $n \geq 1$, $s_1 = s$, $s_n = s'$, and for each $i = 1, \ldots, n-1$ it is $\mathbf{R}(s_i, s_{i+1}) > 0$.

For a set of states A, by $\mathscr{C}[A]$ we denote the CTMC in which states of A have been made absorbing. For $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ and $A \subseteq S$ we have $\mathscr{C}[A] := (S, s_{init}, \mathbf{R}', L)$ with $\mathbf{R}'(s,s') := \mathbf{R}(s,s')$ for $s \notin A$ and $\mathbf{R}'(s,s') := 0$ else, for all $s, s' \in S$. Given a rate matrix $\mathbf{R}$, we define the corresponding *infinitesimal generator matrix* $\mathbf{Q}$ such that for s, s' if $s \neq s'$ then $\mathbf{Q}(s,s') := \mathbf{R}(s,s')$ and we let $\mathbf{Q}(s,s) := -\sum_{s' \in S, s' \neq s} \mathbf{R}(s,s')$.

Paths and probabilistic measure Let $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ be a CTMC. A path is a sequence $\sigma = s_1 t_1 s_2 t_2 \ldots$ satisfying $\mathbf{R}(s_i, s_{i+1}) > 0$, and $t_i \in \mathbb{R}_{\geq 0}$. Paths are either infinite or have a last state s_n with $\mathbf{R}(s_n, S) = 0$. For the path σ and $i \in \mathbb{N}$, let $\sigma[i] = s_i$ denote the i-th state, and let $\delta(\sigma, i) = t_i$ denote the time spent in s_i. For $t \in \mathbb{R}_{\geq 0}$, let $\sigma@t$ denote $\sigma[i]$ such that i is the smallest index with $t \leq \sum_{j=1}^{i} t_j$. For $\mathscr{C}$, let $Path^{\mathscr{C}}$ denote the set of all paths, and $Path^{\mathscr{C}}(s)$ denote the set of all paths starting from s. A probability measure $Pr_s^{\mathscr{C}}$ on measurable subsets of $Path^{\mathscr{C}}(s)$ is uniquely defined. We omit the superscript $\mathscr{C}$ if it is clear from context.

2.3 Markov Population Models

In this paper, we will consider *Markov population models* (MPMs), a sub-class of CTMC, where states encode the number of individuals of certain population types represented by a vector over the natural numbers. Transitions between those states are defined by a change vector that characterizes the successor state and a rate determined by a propensity function that is evaluated in the predecessor state.

Formally, a MPM with d population types is a CTMC $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ with $S \subseteq \mathbb{N}^d$. The rate matrix R will be induced by *transition classes*.

$$[\![s,a]\!] := L(s,a),\ [\![s,\Phi_1\wedge\Phi_2]\!] := [\![s,\Phi_1]\!]\sqcap[\![s,\Phi_2]\!],\ [\![s,\neg\Phi]\!] := [\![s,\Phi]\!]^c,$$

$$[\![s,\mathscr{P}_{\bowtie p}(\phi)]\!] := \begin{cases} \top, & Pr_s\{\sigma|[\![\sigma,\phi]\!]=\top\}\bowtie p \wedge Pr_s\{\sigma|[\![\sigma,\phi]\!]\neq\bot\}\bowtie p,\\ \bot, & Pr_s\{\sigma|[\![\sigma,\phi]\!]=\top\}\not\bowtie p \wedge Pr_s\{\sigma|[\![\sigma,\phi]\!]\neq\bot\}\not\bowtie p,\\ ? & \text{else,}\end{cases}$$

$$[\![s,\mathscr{S}_{\bowtie p}(\Phi)]\!] := \begin{cases} \top, & S_L(\Phi)\bowtie p \wedge S_U(\Phi)\bowtie p,\\ \bot, & S_L(\Phi)\not\bowtie p \wedge S_U(\Phi)\not\bowtie p,\\ ? & \text{else,}\end{cases}$$

where

$$S_L(\Phi) := \lim_{t\to\infty} Pr_s\{\sigma \mid [\![\sigma@t,\Phi]\!]=\top\},$$

$$S_U(\Phi) := \lim_{t\to\infty} Pr_s\{\sigma \mid [\![\sigma@t,\Phi]\!]\neq\bot\},$$

$$[\![\sigma,\mathscr{X}^I\Phi]\!] := (\top \text{ iff } \delta(\sigma,0)\in I, \bot \text{ else})\sqcap[\![\sigma[1],\Phi]\!],$$

$$[\![\sigma,\Phi_1\,\mathscr{U}^I\,\Phi_2]\!] := \begin{cases} \top, & \exists t{\in}I.[\![\sigma@t,\Phi_2]\!]=\top \wedge \forall t'{<}t.[\![\sigma@t',\Phi_1]\!]=\top,\\ \bot, & \forall t{\in}I.[\![\sigma@t,\Phi_2]\!]=\bot \vee \exists t'{<}t.[\![\sigma@t',\Phi_1]\!]=\bot,\\ ? & \text{else.}\end{cases}$$

Table 2: CSL semantics.

Definition 2 (Transition Class) *A* transition class *is a tuple* (α,ν) *where* $\alpha:\mathbb{N}^d\to\mathbb{R}_{\geq 0}$ *is the* propensity function *and* $\nu\in\mathbb{Z}^d\setminus\{0\}$ *is the* change vector.

For the propensity functions, we will restrict to multivariate polynomials in $\mathbb{N}^d$. Given a set of transition classes $\{\tau_1,\ldots,\tau_m\}$, we define the entries of $\mathbf{R}$ via $\mathbf{R}(x,x+\nu)=\sum_{\{j\mid \nu_j=\nu\}}\alpha_j(x)$ for $\tau_j=(\alpha_j,\nu_j)$ and $1\leq j\leq m$. In order to ensure $\mathbf{R}$ is well defined, we demand that for all states $x\in S$ and all transition classes (α_j,ν_j) of our model $\alpha_j(x)>0$ only holds if $x+\nu_j\in S$. For the labelling L we demand that each state is labelled by those boolean expressions over population counts and constants that evaluate to true. We also refer to the populations by their name. For example, if we have a MPM with populations A (x_0) and B (x_1), then state $(3,4)$ would be labelled by e.g. $A\leq c$ for all $c\geq 3$, $A+B\leq 7$ and $A^2+B^2<30$ to name just a few.

2.4 CSL Model Checking

We consider the logic CSL [3] interpreted over a ternary logic [16, 17]. Let $I=[t,t']$ be an interval with $t,t'\in\mathbb{R}_{\geq 0}\cup\{\infty\}$ with $t'=\infty\Rightarrow t=0$ and $t\leq t'$. Let $p\in[0,1]$ and $\bowtie\in\{\leq,<,>,\geq\}$. The syntax of state formulae (Φ) and path formulae (ϕ) is:

$$\begin{aligned}\Phi &= a \mid \neg\Phi \mid \Phi\wedge\Phi \mid \mathscr{P}_{\bowtie p}(\phi) \mid \mathscr{S}_{\bowtie p}(\Phi),\\ \phi &= \mathscr{X}^I\Phi \mid \Phi\,\mathscr{U}^I\,\Phi.\end{aligned}$$

Let $\mathbb{F}$ be the set of all CSL formulae. The truth value $[\![\cdot]\!]_{\mathscr{C}}:((S\cup Path)\times\mathbb{F})\to\mathbb{B}_3$ of formulae is defined inductively in Table 2. If the model under consideration is clear from the context, we leave out the index $\mathscr{C}$. The model $\mathscr{C}$ satisfies a formula Φ if the initial state s_{init} does. We specify the *unbounded until* as $\Phi_1\,\mathscr{U}\,\Phi_2:=\Phi_1\,\mathscr{U}^{[0,\infty)}\,\Phi_2$ and the *eventually operator* as $\Diamond^I\Phi:=true\,\mathscr{U}^I\,\Phi$ where $true=a\vee\neg a$. Let $B\subseteq S$ be a set of states. For better readability, we use the name of the set B as an atomic proposition in formulae to characterize that the system is in a state contained in B.

Model checking CSL formulae without the operators $\mathscr{X}$ and $\mathscr{S}$ on a finite CTMC over a ternary logic has already been used before to handle a different abstraction technique for CTMCs [16, 17]. Adding routines to handle $\mathscr{X}$ and $\mathscr{S}$ is straightforward.

3 Model Checking based on Truncations

Now we discuss how to model check CSL formulae on infinite CTMCs. To this end, our goal is to operate on a finite truncation instead of the original infinite CTMC. In a nutshell, starting from the initial state, we explore the states of the infinite model until we assume that we have explored enough of them to obtain a useful result in the following steps. Then, we remove all transitions from the states at the border and set all of their atomic propositions to ?. In previous works [15] we already discussed some variants of such a model exploration. In Subsection 3.1 we give another such technique. It is more efficient than the previous methods, as it can explore model states in a more target-oriented way. Thus, it needs to explore less states than the previous methods. Afterwards, we discuss how to build a finite CTMC truncation for a nested CSL formula. Finally, in Subsection 3.3, we explain how to obtain results for the infinite-state CTMC only using the finite submodel.

3.1 Truncation-based Reachability Analysis

Given a set of states $\mathscr{W}_0$, a CTMC $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ and CSL formula of the form $\Phi = \mathscr{P}_{\bowtie p}(\Phi_1 \,\mathscr{U}\, \Phi_2)$, we want to compute a finite submodel $\mathscr{C}|_{\mathscr{W}} = (\mathscr{W} \cup \overline{\mathscr{W}}, \mathbf{R}_{\mathscr{W}}, L_{\mathscr{W}})$ sufficient to decide Φ on all states of $\mathscr{W}_0$. We define *finite truncations* of a CTMC.

Definition 3 *Let $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ be a CTMC. Let $\mathscr{W} \subseteq S$ be a finite subset of S, and let $\overline{\mathscr{W}} := \text{post}(\mathscr{W}) \setminus \mathscr{W}$. The* finite truncation *of $\mathscr{C}$ is the finite CTMC $\mathscr{C}|_{\mathscr{W}} := (\mathscr{W} \cup \overline{\mathscr{W}}, \mathbf{R}_{\mathscr{W}}, L_{\mathscr{W}})$ where $L_{\mathscr{W}}(s,\cdot) := L(s,\cdot)$ if $s \in \mathscr{W}$, and $L_{\mathscr{W}}(s,\cdot) = ?$ else. The rate matrix is defined by $\mathbf{R}_{\mathscr{W}}(s,s') := \mathbf{R}(s,s')$ if $s \in \mathscr{W}$, and $\mathbf{R}_{\mathscr{W}}(s,\cdot) := 0$ else.*

We build the truncation of the model iteratively, using the high-level (transition class) description of the model. Starting from states A, we explore the model until for all $s \in \mathscr{W}_0$ the probability to reach states in $\overline{\mathscr{W}}$ is below an accuracy ε, which we may choose as a fixed value or due to the probability bound p.

Algorithm 1 describes how we can obtain a sufficiently large state set $\mathscr{W}$. For $s \in S$, $s' \in \overline{\mathscr{W}}$ and $t \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ we use $\pi_{\mathscr{C}}(s,t,s')$ to denote the probability that at time $t \in \mathbb{R}_{\geq 0}$, the CTMC $\mathscr{C}$ is in state s', under the condition that it was in s initially. For $t = \infty$, we let π denote the limit for $t \to \infty$ if this value exists. If s will reach absorbing states with probability one, this is the case. Further, for a set of absorbing states B, we let $\xi_{\mathscr{C}}(s,t,B) := \sum_{s' \in B} \pi_{\mathscr{C}}(s,t,s')$ denote the probability to reach B within time $t \in \mathbb{R}_{\geq 0} \cup \{\infty\}$. Given a fixed s and t, we can compute $\pi_{\mathscr{C}}(s,t,s')$ for all s' at once effectively, and given B and t we can compute $\xi(s,t,B)$ for all s at once effectively [3].

The algorithm is started on a CTMC $\mathscr{C}$ and a set of states $\mathscr{W}_0$, for which we want to decide the property. We also provide the time bound t as well as the accuracy ε. With $\hat{\mathscr{W}}$ we denote a set of states for which the exploration algorithm may stop immediately, as further exploration is not needed to decide the given property. For Φ above and $I = [0,a], a \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, we could specify $\hat{\mathscr{W}}$ as the states which fulfill $\Phi_2 \vee (\neg\Phi_1 \wedge \neg\Phi_2)$. For all paths of the model, the truth value is independent of the states after such a state.

Algorithm 1: TRANSIENTTRUNC$(\mathscr{C}, \mathscr{W}_0, \hat{\mathscr{W}}, t, \varepsilon)$.

$\mathscr{W} := \mathscr{W}_0$
$\overline{\mathscr{W}} := \text{post}(\mathscr{W}) \setminus \mathscr{W}$
while $\max_{s \in \mathscr{W}_0} \xi_{\mathscr{C}[\overline{\mathscr{W}}]}(s, t, \overline{\mathscr{W}} \setminus \hat{\mathscr{W}}) \geq \varepsilon$ **do**
  choose s from $\arg\max_{s \in \mathscr{W}_0} \xi_{\mathscr{C}[\overline{\mathscr{W}}]}(s, t, \overline{\mathscr{W}} \setminus \hat{\mathscr{W}})$
  while $\xi_{\mathscr{C}[\overline{\mathscr{W}}]}(s, t, \overline{\mathscr{W}} \setminus \hat{\mathscr{W}}) \geq \varepsilon$ **do**
    choose $A \subseteq (\overline{\mathscr{W}} \setminus \hat{\mathscr{W}})$ such that $\pi_{\mathscr{C}[\overline{\mathscr{W}}]}(s, t, A) \geq \varepsilon$
    $\mathscr{W} := \mathscr{W} \cup A$
    $\overline{\mathscr{W}} := \text{post}(\mathscr{W}) \setminus \mathscr{W}$
return $\mathscr{W} \cup (\text{post}(\mathscr{W}) \cap \hat{\mathscr{W}})$

3.2 Truncation-based Steady-state Analysis

In the following, we will develop a technique to retrieve a finite subset of states $\mathscr{W} \subseteq S$ that contains most of the total steady-state probability mass, i.e. $\sum_{c \in \mathscr{W}} \pi(c) > 1 - \varepsilon$ for a given $\varepsilon < 1$. The next step will be to derive lower and upper bounds on the state-wise steady-state probabilities inside that window $\mathscr{W}$.

Geometric Bounds For the presented methodology to be applicable, we have to restrict our models to *ergodic* MPMs, since for steady-state analysis the equilibrium distribution has to exist uniquely. Ergodicity can be verified by the means of Lyapunov functions and the following theorem. In the following, by $X(t)$ we refer to the stochastic process underlying the MPM, and by E to the expectation of a random variable.

Definition 4 (Lyapunov Function) *A* Lyapunov function *is a function* $g : \mathbb{N}^d \to \mathbb{R}_{\geq 0}$.

A suitable Lyapunov function which also we used for our experimental results is the squared Euclidean norm $||\cdot||_2^2$ defined by $||x||_2^2 = x_1^2 + x_2^2 + \cdots + x_d^2$ where d is the number of population types.

Theorem 1 (Tweedie [26]) *Assuming that* $X(t)$ *is* irreducible*, it is also* ergodic *and uniquely determined by its infinitesimal generator iff there exists a Lyapunov function* g^**, a finite subset* $\mathscr{W} \subseteq S$*, and a constant* $\lambda > 0$ *such that*

1. $\frac{\mathrm{d}}{\mathrm{d}t} E[g^*(X(t)) \mid X(t) = x] \leq -\lambda$ *for all* $x \in \mathbb{N}^d \setminus \mathscr{W}$*,*
2. $\frac{\mathrm{d}}{\mathrm{d}t} E[g^*(X(t)) \mid X(t) = x] < \infty$ *for all* $x \in \mathscr{W}$*,*
3. *the set* $\{x \in \mathbb{N}^d \mid g^*(x) \leq l\}$ *is finite for all* $l < \infty$*.*

Therefore, given a MPM $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ induced by a set of transition classes $\{\tau_1, \ldots, \tau_m\}$, we can use Theorem 1 for a semi-decision procedure to check ergodicity by choosing candidates for Lyapunov functions. Our experience has shown that in most cases rather simple functions, i.e. multivariate polynomials of degree two, already suffice for usual models from systems biology and queuing theory. Consequently, we restrict the choice of Lyapunov functions to that class.

We can exploit Theorem 1 also to retrieve the aforementioned window $\mathscr{W}$ that encloses most of the steady-state probability mass. At first, let the *drift* $d^*(x)$ in state $x \in \mathbb{N}^d$ be defined as

$$d^*(x) := \frac{\mathrm{d}}{\mathrm{d}t} E[g^*(X(t)) \mid X(t) = x] = (\mathbf{Q} g^*)(x).$$

Due to the transition class induced structure of $\mathbf{Q}$, the drift is given as

$$d^*(x) = \sum_{j=1}^{m} \alpha_j(x)(g^*(x+v_j) - g^*(x))$$

and can easily be represented symbolically in a state x. Since the propensity functions of our models as well as the Lyapunov function are multivariate polynomials, so is the drift. The next step is to retrieve a positive real number $c \geq \max_{x\in\mathbb{N}^d} d^*(x)$. In order to find that global maximal drift, common global optimization techniques like gradient based methods and simulated annealing can not be used, since there is no guarantee to get the real global maximum. What we propose instead is to solve $\nabla d^*(x)|_p = 0$ for all $p \in 2^{\{x_1,\ldots,x_m\}}$, where $f(x)|_p$ denotes the projection of the multivariate function f onto the subspace spanned by $x_i \in p$ to retrieve all K possible candidates m_k. In order to solve the equation systems $\nabla d^*(x)|_p = 0$, we suggest the use of the polyhedral homotopy continuation method, which is guaranteed to find all roots. For implementations, we refer to [13] and [18]. Finally, after restricting the candidate set to $M = \{m_k \mid m_k \in \mathbb{R}^d_{\geq 0}\}$, we set $c = \max_{m\in M} d^*(m)$. Please note, that due to the existence of a maximal value c, the chosen Lyapunov function serves as a witness for ergodicity assuming an irreducible MPM. By scaling the Lyapunov function by $\frac{1}{c+\gamma}$ we retrieve the normalized drift

$$d(x) = \frac{\mathrm{d}}{\mathrm{d}t} E[g(X(t)) \mid X(t) = x]$$

and using conditions 1 and 2 from Theorem 1 we get

$$d(x) \leq \frac{c}{c+\gamma} - \bar{\chi}_{\mathscr{W}}(x), \tag{1}$$

where $\bar{\chi}_{\mathscr{W}}(x) = 1$ if $c \notin \mathscr{W}$ and 0 else. Multiplying Inequality 1 with $\pi(x)$ and summing over x leads us to

$$\pi(\bar{\mathscr{W}}) = \sum_{x\notin\mathscr{W}} \pi(x) \leq \frac{c}{c+\gamma}.$$

Convergence of this sum is guaranteed for (infinite) ergodic MPMs as stated in [10]. Consequently, we can exploit this inequality by directly choosing $d(x) = \frac{\varepsilon}{c} d^*(x)$, i.e. setting $\lambda > 0$ such that $\varepsilon = \frac{c}{c+\gamma}$, to get $\pi(\mathscr{W}) > 1-\varepsilon$ for

$$\mathscr{W} = \{x \in \mathbb{N}^d \mid d(x) > \varepsilon - 1\}.$$

State-wise Bounds Given our state space window $\mathscr{W}$, our next goal is to get lower ($l(x)$) and upper ($u(x)$) bounds on the steady-state probabilities inside $\mathscr{W}$, i.e. probability vectors l and u such that $l(x) \leq \pi(x) \leq u(x)$ for all $c \in \mathscr{W}$. For this, we will employ the methodology developed by Courtois and Semal [5] [6] which we have extended to infinite state MPMs in [7]:

Theorem 2 ([7]) *Let $\mathbf{Q}$ be the infinitesimal generator of an ergodic CTMC $X(t)$ with countably infinite state space S and let $\mathscr{W} \subseteq S$ be a finite subset of the state space. Further, we let the matrix $\mathbf{C}$ be the finite submatrix of $\mathbf{Q}$ containing exactly the states in $\mathscr{W}$. If by $\mathbf{U}$ we refer to the uniformized CTMC of $\mathbf{C}$, i.e.*

$$\mathbf{U} = \mathbf{I} + \alpha^{-1}\mathbf{C} \text{ with } \alpha > \max_i -\mathbf{C}(i,i),$$

then for all $x \in \mathscr{W}$ we have

$$\min_j \pi^{\mathbf{U_j}}(x) \leq \frac{\pi(x)}{\sum_{x\in\mathscr{W}} \pi(x)} \leq \max_j \pi^{\mathbf{U_j}}(x)$$

where $\pi^{\mathbf{U_j}}$ is the steady-state distribution of matrix $\mathbf{U_j}$ which is matrix $\mathbf{U}$ made stochastic by increasing column j.

As presented in the previous paragraph, we have $1-\varepsilon < \sum_{x\in\mathscr{W}} \pi(x) \leq 1$. Consequently, we can use the geometric bounding technique to obtain the unconditional state-wise bounds from Theorem 2, that is to retrieve for all states $x \in \mathscr{W}$

$$l(x) = (1-\varepsilon)\min_j \pi^{\mathbf{U}_j}(x) \leq \pi(x) \leq \max_j \pi^{\mathbf{U}_j}(x) = u(x).$$

Geobound All these presented techniques, i.e., the retrieval of geometric bound via Lyapunov functions and the computation of state-wise steady-state bounds via the methodology of the previous paragraph have been implemented in a prototypical tool called GEOBOUND [23].

3.3 Truncation-based CSL Model Checking

Given a CTMC $\mathscr{C}$, we want to check whether $\mathscr{C}$ satisfies Φ. This is done in two phases. At first, we construct a finite truncation that is sufficient to check the formula. To this end, we employ an algorithm to determine suitable truncations. The states explored depend on the specific CSL formula analyzed. The computation works by recursive descent into sub-formulae. The most intricate formulae are the probabilistic operators, for which we use the technique from Section 3.1. After the exploration, we can compute $[\![s_{init}, \Phi]\!]$ on the finite truncation.

Algorithm 2: TRUNCATE$(\mathscr{C} = (S, s_{init}, \mathbf{R}, L), \Phi)$. CSL state space exploration.

function TRUNC$(\mathscr{C}, \mathscr{W}, \Phi)$
  switch Φ **do**
    case a **return** $\mathscr{W}$
    case $\neg\Psi$ **return** TRUNC$(\mathscr{C}, \mathscr{W}, \Psi)$
    case $\Phi_1 \wedge \Phi_2$ **return** TRUNC$(\mathscr{C}, \mathscr{W}, \Phi_1) \cup$ TRUNC$(\mathscr{C}, \mathscr{W}, \Phi_2)$
    case $\mathscr{P}_{\bowtie p}(\mathscr{X}^I \Psi)$ **return** TRUNC$(\mathscr{C}, \mathscr{W} \cup \overline{\mathscr{W}}, \Psi)$
    case $\mathscr{P}_{\bowtie p}(\Phi_1 \,\mathscr{U}^{[t,t']}\, \Phi_2)$
      $\mathscr{W}_t =$ TRANSIENTTRUNC$(\mathscr{C}, \mathscr{W}, \text{stop}(\Phi), t, \varepsilon)$
      $\mathscr{W}_{t'} =$ TRANSIENTTRUNC$(\mathscr{C}, \mathscr{W}_t, \text{stop}(\Phi), t'-t, \varepsilon)$
      return TRUNC$(\mathscr{C}, \mathscr{W}_{t'}, \Phi_1) \cup$ TRUNC$(\mathscr{C}, \mathscr{W}_{t'}, \Phi_2)$
    case $\mathscr{S}_{\bowtie p}(\Psi)$ **return** TRUNC$(\mathscr{C}, \text{post}(\mathscr{W}) \setminus \mathscr{W}, \Psi)$
return $\mathscr{C}|_{\text{TRUNC}(\mathscr{C}, \{s_{init}\}, \Phi)}$

Algorithm 2 describes the exploration component. Given a CTMC $\mathscr{C} = (S, s_{init}, \mathbf{R}, L)$ be a CTMC and a state formula Φ, we call TRUNC$(\mathscr{C}, \{s_{init}\}, \Phi)$. Afterwards, we can use the CSL model checking algorithm for a ternary logic on the model obtained this way. With stop(Φ) we denote a set of states for which we can stop the exploration immediately, as exemplified in Section 3.1. For nested formulae, this value is computed by a simple precomputation in a recursive manner.

We employ ternary CSL model checking on the finite model obtained. However, we have already obtained state-wise bounds on the steady-state probabilities beforehand using the approach presented in Section 3.2 and implemented in the tool GEOBOUND. Thus, to obtain the lower bound probabilities of $\mathscr{S}_{\bowtie p}(\Psi)$, we sum up the lower bound steady-state probabilities of states $s \in \mathscr{W}$ with $[\![x, \Psi]\!] = \top$. For the upper probability bound, we sum upper steady-state probabilities of states s with $[\![x, \Psi]\!] \neq \bot$ and add the probability ε that limits the steady-state probability outside $\mathscr{W}$. The probabilities computed are the probabilities for all states, because the model is ergodic.

Correctness Consider a truncation $\mathscr{C}|_{\mathscr{W}} = (\mathscr{W} \cup \overline{\mathscr{W}}, \mathbf{R}_{\mathscr{W}}, L_{\mathscr{W}})$ constructed for the input CTMC $\mathscr{C}$ and a state formula Φ. If we obtain the truth value $[\![s, \Phi]\!] \neq ?$ in $\mathscr{C}_{\mathscr{W}}$, then this is also the truth value in $\mathscr{C}$: The correctness is independent of the exploration algorithm, which plays a role for performance and applicability of the approach. If too many states are explored, we may run out of memory, whereas if too few are explored, we are unable to decide the value in the original model. As a result, the correctness of the algorithm for CSL without steady state follows by giving a simulation relation [17, Definition 3.4.2] between $\mathscr{C}$ and $\mathscr{C}|_{\mathscr{W}}$ and [17, Theorem 4.5.2]. The correctness of the steady-state extension follows as we give safe upper and lower bounds in exactly the same way as it is done in [17, Theorem 4.5.2].

4 Experimental Results

Using several case studies, we assess the effectiveness of our technique. For that, we have combined the tool GEOBOUND [23] to compute bounds on steady-state probabilities for Markov population models with the infinite-state model checker INFAMY [14]. This way, we can effectively handle the combination of models and properties described in this paper. To show the efficiency of the approach, we applied our tool chain on a number of models from different areas. The results were obtained on an Ubuntu 10.04 machine with an Intel dual-core processor at 2.66 GHz equipped with 3 GB of RAM. The tools used are available[1]. Instead of the truth value for the formula under considerations, in the result tables we give intervals of the probability measure of the outer formula. Note that in case of a single (ergodic) strongly connected component, there is a unique steady state distribution and it suffices to state a single pair of lower and upper probability bounds for a CSL formula with outer steady state operator since the validity is the same for each state. We make use of a derived operator for conditional steady-state measures defined as

$$[\![s, \mathscr{S}_{\bowtie p}(\Phi_1 \mid \Phi_2)]\!] = \begin{cases} \top, & S_L(\Phi_1|\Phi_2) \bowtie p \wedge S_U(\Phi_1|\Phi_2) \bowtie p, \\ \bot, & S_L(\Phi_1|\Phi_2) \not\bowtie p \wedge S_U(\Phi_1|\Phi_2) \not\bowtie p, \\ ? & \text{else}, \end{cases}$$

where

$$\begin{aligned} S_L(\Phi_1 \mid \Phi_2) &:= \frac{S_L(\Phi_1 \wedge \Phi_2)}{S_U(\Phi_2)}, \\ S_U(\Phi_1 \mid \Phi_2) &:= \frac{S_U(\Phi_1 \wedge \Phi_2)}{S_L(\Phi_2)}. \end{aligned}$$

Protein Synthesis [12] We analyze the MPM encoding protein synthesis, as depicted in Table 3. In biological cells, each protein (P, x_2) is encoded by a gene (G, x_1). If the gene is active ($G = 1$), the corresponding protein will be synthesized with rate $\nu = 1.0$. Proteins may degenerate with rate $\delta = 0.02$ and thus disappear after a time. The gene switches from active state to inactive ($G = 0$) with rate $\mu = 5.0$ and vice versa with rate $\lambda = 1.0$. Note that in a previous paper, this model has been presented as a stochastic Petri net (SPN). Often, transition class models and SPN (without zero-arcs) can trivially be encoded within each other.

We consider the property that on the long run, given that there are more than 20 proteins, a state with 20 or less proteins is most likely (with a probability of at least 0.9) reached within t time units:

$$\mathscr{S}_{>p}(\mathscr{P}_{>0.9}(\Diamond^{[0,t]} P \leq 20) \mid P > 20).$$

j	τ_j	$\alpha_j(x)$	$\nu^{(j)}$
1	(α_1, ν_1)	$\lambda(1-x_1)$	e_1^T
2	(α_2, ν_2)	μx_1	$-e_1^T$
3	(α_3, ν_3)	νx_1	e_2^T
4	(α_4, ν_4)	δx_2	$-e_2^T$

Table 3: Transition classes of the protein synthesis model.

Table 4: Protein synthesis results.

ε	t	depth	time		n	probability bounds
			GEOBOUND	INFAMY		$[S_L, S_U]$
10^{-1}	10	8	0.9	0.1	217	[0.002, 1.0]
	20	8		0.1	217	[0.003, 1.0]
	60	8		0.1	217	[0.004, 1.0]
10^{-3}	10	5	3.2	0.3	1531	[0.144, 1.0]
	20	5		0.4	1531	[0.259, 0.816]
	60	5		0.7	1531	[0.317, 1.0]
10^{-6}	10	3	7.8	34.3	46431	[0.451350, 0.454779]
	20	3		67.0	46431	[0.813116, 0.817401]
	60	3		257.5	46431	[0.997642, 1.0]

We give results in Table 4. The shortcut stop of Algorithm 2 was not used for the analysis. With "depth" we specify the number of states of the shortest path from the initial state to any other state of the truncation. The runtimes of GEOBOUND and INFAMY is given in seconds. The rate of decay of proteins depends on the number of proteins existing. For the states on the border of $\mathscr{W}$, we have large rates back to existing states. Because of this, for the given parameters the state space exploration algorithm does not need to explore further states, and the total number of states explored n does not increase with the time bound. To obtain different n, we would have needed to choose extremely large time bounds, for which analysis would be on the one hand infeasible and on the other hand would lead to results extremely close to 1. The lower and upper bounds are further away than ε. This results, because we have to divide by $S_U(\Phi_2)$ for the lower and by $S_L(\Phi_2)$ for the lower bound. In turn, this may lead to a much larger error than ε.

Gene Expression [25] Next, we analyze a network of chemical reactions where a gene is transcribed into mRNA (M) with rate $\lambda = 25.0$ and the mRNA is translated into proteins (P) with rate $\mu = 1.0$. Both populations can degrade at rates $\delta_M = 2.0$ and $\delta_P = 1.0$, respectively. The corresponding transition classes are listed in Table 5. The property of interest is the steady-state probability of leaving a certain set of states W enclosing more than 80% of the steady-state probability mass most likely within t time units, i.e.,

$$\mathscr{S}_{>p}(\mathscr{P}_{>0.9}(\Diamond^{[0,t]}\neg W) \mid W),$$

[1] `http://alma.cs.uni-saarland.de/?page_id=74`

j	τ_j	$\alpha_j(x)$	$\nu^{(j)}$
1	(α_1, ν_1)	λ	e_1^T
2	(α_2, ν_2)	μx_1	e_2^T
3	(α_3, ν_3)	$\delta_M x_1$	$-e_1^T$
4	(α_4, ν_4)	$\delta_P x_2$	$-e_2^T$

Table 5: Transition classes of the gene expression model.

where

$$W := M > 5 \wedge M < 20 \wedge P > 5 \wedge P < 20$$

with $\mathscr{S}_{>0.8}(W) = \top$.

Table 6: Gene expression results.

ε	t	depth	time		n	probability bounds
			GEOBOUND	INFAMY		$[S_L, S_U]$
10^{-1}	2	24	3.4	5.2	2558	[0.01, 0.2]
	4	24		6.0	2558	[0.3, 0.6]
	8	24		8.5	2558	[0.8, 1.0]
$5 \cdot 10^{-2}$	2	20	6.1	11.9	3663	[0.015, 0.078]
	4	20		15.4	3663	[0.34, 0.46]
	8	20		22.1	3663	[0.90, 1.0]
10^{-2}	2	15	8.5	99.3	11736	[0.015, 0.029]
	4	15		139.9	11736	[0.37, 0.40]
	8	15		219.5	11736	[0.97, 1.0]

The results are stated in Table 6. Similar to the protein synthesis case study, we see that there is no increase in the number of states, because the window size already comprises enough states for the transient analysis. In Table 7 we consider results for the subformula $\mathscr{P}_{>0.9}(\Diamond^{[0,t]}\neg W)$. We compare the methods to explore states for the transient until described in this paper (Advanced) with the finite state projection [21] (FSP) previously used in INFAMY. We see that the time needed is comparable, but the new algorithm needs to explore less states. This is the case because with the method introduced here when building the finite truncation we have more control into which direction we explore. In contrast, the FSP explores the model into all directions at the same time, until enough precision is reached. When we use the shortcut stop (Advanced+AP), we will not explore states further in which $\neg W$ holds. When exploring the model, for larger time bounds there is some point at which there are almost only states of which all successors have been completely explored and states for which $\neg W$ holds. Thus, the maximal number of states to be explored is constant with this optimization, except for very large time bounds.

For the protein synthesis, there is almost no difference in the number of states needed by the new method and FSP. Because it has only one infinite variable, there is just one direction to be explored. Thus, the new method performs worse, as it needs more effort to explore into this direction.

Using the shortcut method also allows us to handle the formula $\mathscr{P}_{>0.9}(\Diamond\neg W)$, involving the unbounded until operator. Using a precision of 10^{-6}, we needed a total time of 1.5 seconds, reached 974 states and obtained a reachability probability almost 1.0. Using a precision of 10^{-10}, we needed

Table 7: Gene expression - comparison of methods.

T	FSP		Advanced		Advanced+AP	
	time	n	time	n	time	n
1	0.3	1223	0.5	803	0.3	495
2	0.8	1889	1.1	1257	0.6	838
3	1.2	2209	1.5	1460	0.8	945
4	1.5	2344	1.9	1557	0.9	971
5	1.8	2483	2.1	1610	0.9	974
6	2.1	2483	2.4	1647	1.0	974
7	2.4	2554	2.7	1674	1.1	974
8	2.7	2554	2.9	1690	1.2	974
9	3.0	2626	3.2	1707	1.2	974
10	3.2	2626	3.3	1720	1.3	974

3.7 seconds and explored 1823 states. For the computation of unbounded until probabilities, efficient specialized algorithms were used, which explains that we needed less time than for some of the time bounded experiments.

Exclusive Switch [19] The exclusive switch is a gene regulatory network with one promotor region shared by two genes. That promotor region can either be unbound ($G = 1, G.P_1 = 0, G.P_2 = 0$) or bound by a protein expressed by gene 1 ($G = 0, G.P_1 = 1, G.P_2 = 0$) or gene 2 ($G = 0, G.P_1 = 0, G.P_2 = 1$). If the promotor is unbound, both proteins are expressed, each with rate $\rho = 0.05$, otherwise only the protein that is currently bound to the promotor is produced at rate ρ. Proteins degrade at rate $\delta = 0.005$. Binding happens at rate $\lambda = 0.01$ and unbinding at rate $\mu = 0.008$. The transition class structure is given in Table 8.

j	τ_j	$\alpha_j(x)$	$v^{(j)}$
1	(α_1, v_1)	ρx_3	e_1^T
2	(α_2, v_2)	ρx_3	e_2^T
3	(α_3, v_3)	δx_1	$-e_1^T$
4	(α_4, v_4)	δx_2	$-e_2^T$
5	(α_5, v_5)	$\lambda x_1 x_3$	$(-e_1 - e_3 + e_4)^T$
6	(α_6, v_6)	$\lambda x_2 x_3$	$(-e_2 - e_3 + e_5)^T$
7	(α_7, v_7)	μx_4	$(-e_4 + e_1 + e_3)^T$
8	(α_8, v_8)	μx_5	$(-e_5 + e_2 + e_3)^T$
9	(α_9, v_9)	ρx_4	e_1^T
10	(α_{10}, v_{10})	ρx_5	e_2^T

Table 8: Transition classes of the exclusive switch.

This system has two attractor regions, i.e., two spatial maxima in the steady-state probability dis-

tribution over the protein levels, one at $P_1 = 10, P_2 = 0$ and the other one at $P_1 = 0, P_2 = 10$. We are interested in the switching time between these two regions. For this, we estimate the time needed for a 90%-quantile of the steady-state probability mass of one of the two attractors to reach the other attractor region. More precisely, let

$$\begin{aligned} \mathit{start} &:= ||(P_1,P_2)-(10,0)||_2^2 \leq 4 \\ \mathit{end} &:= ||(P_1,P_2)-(0,10)||_2^2 \leq 4. \end{aligned}$$

Then, the formula to check is

$$\mathscr{S}_{>p}(\mathscr{P}_{>0.9}(\Diamond^{[0,t]}\mathit{end}) \mid \mathit{start}).$$

Note that since the model is symmetric we only have to check one formula from one attractor to the other. The corresponding results are depicted in Table 9.

Table 9: Exclusive switch results.

ε	t	depth	time		n	probability bounds
			GEOBOUND	INFAMY		$[S_L, S_U]$
10^{-1}	7700	14	5.8	47.9	3414	[0.2, 0.7]
	7800	14		48.0	3414	[0.3, 0.9]
	7900	14		47.6	3414	[0.5, 1.0]
	8000	14		48.1	3414	[0.6, 1.0]
$5 \cdot 10^{-2}$	7700	12	6.9	128.3	4848	[0.26, 0.49]
	7800	12		129.7	4848	[0.43, 0.70]
	7900	12		130.5	4848	[0.64, 0.98]
	8000	12		131.0	4848	[0.83, 1.0]
10^{-2}	7700	8	86.2	1881.6	14806	[0.30, 0.35]
	7800	8		1904.5	14806	[0.50, 0.56]
	7900	8		1930.1	14806	[0.75, 0.82]
	8000	8		1942.9	14806	[0.96, 1.0]

From these results we may conclude that in half of the cases, most likely the switching time between the attractor regions is at most 7800 time units, while in almost all cases the switching time is most likely below 8000 time units, assuming the system has stabilized to a steady state.

5 Related Work

The techniques of this paper are derived from combinations of our previous works [14, 15] and [7, 23]. This work has been inspired and is related by a number of other works.

Finite state projection (FSP) by Munsky and Khammash [21] is closely related. The method also works on building a finite truncation of the original model. The proofs given work for general truncations, but in their publications they always use an algorithm which explores the model in a breadth-first way. They consider time-bounded reachability and no logic like CSL. Adaptive uniformization for CTMCs was introduced by van Moorsel and Sanders [20]. In this approach uniformization is recalibrated to perform well when exploring the state space on the fly. Remke et al. [22] have developed algorithms for model checking CSL against infinite-state CTMCs of Quasi-birth-death processes (QBDs) and Jackson queuing networks (JQNs). The systems to which the method is applicable are less general, but the

approach is less expensive than our method. To the best of the authors knowledge, the method to bound the steady-state probabilities using Lyapunov functions is used here for the first time for general MPMs.

6 Conclusion

In this paper, we have shown how to model check CSL on Markov population models of infinite size. Without the steady-state operator, the method is also applicable for general CTMC derived from a high-level specification. We have evaluated our method on models from the biological domain. The method extends previous related publications by means to check the steady-state operator and gives guarantees for the truth value obtained.

As future work we plan to integrate these methods into our probabilistic online model checker: IS-CASMC:

`http://iscasmc.ios.ac.cn/IscasMC/`

References

[1] William J. Anderson (1991): *Continuous-Time Markov Chains: An Applications-Oriented Approach*. Springer Verlag, doi:10.1007/978-1-4612-3038-0.

[2] Adnan Aziz, Kumud Sanwal, Vigyan Singhal & Robert K. Brayton (2000): *Model-Checking Continuous-Time Markov Chains*. *ACM Trans. Comput. Log* 1(1), pp. 162–170, doi:10.1145/343369.343402.

[3] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns & Joost-Pieter Katoen (2003): *Model-Checking Algorithms for Continuous-Time Markov Chains*. *IEEE Trans. Software Eng* 29(6), pp. 524–541, doi:10.1109/TSE.2003.1205180.

[4] P. Ballarini, R. Mardare & I. Mura (2009): *Analysing Biochemical Oscillation through Probabilistic Model Checking*. In: *Proc. 2nd Workshop From Biology to Concurrency and Back (FBTC'08), Electronic Notes in Theoretical Computer Science* 229 (issue 1), Elsevier, pp. 3–19, doi:10.1016/j.entcs.2009.02.002.

[5] P.-J. Courtois & P Semal (1984): *Bounds for the Positive Eigenvectors of Nonnegative Matrices and for their Approximations by Decomposition*. *J. ACM* 31(4), pp. 804–825, doi:10.1145/1634.1637.

[6] Pierre-Jacques Courtois (1985): *Analysis of Large Markovian Models by Parts. Applications to Queueing Network Models*. In: *Messung, Modellierung und Bewertung von Rechensystemen, 3. GI/NTG-Fachtagung*, pp. 1–10, doi:10.1007/978-3-642-87472-7_1.

[7] Tuğrul Dayar, Holger Hermanns, David Spieler & Verena Wolf (2011): *Bounding the equilibrium distribution of Markov population models*. *Numerical Linear Algebra with Applications* 18(6), pp. 931–946, doi:10.1002/nla.795.

[8] Nico M. van Dijk (1988): *On the finite horizon Bellman equation for controlled Markov jump models with unbounded characteristics*. *Stochastic Proc. Appl* 28, pp. 141–157, doi:10.1016/0304-4149(88)90071-3.

[9] E. Allen Emerson & Edmund M. Clarke (1982): *Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons*. *Sci. Comput. Program* 2(3), pp. 241–266, doi:10.1016/0167-6423(83)90017-5.

[10] P. Glynn & A. Zeevi (2008): *Bounding stationary expectations of Markov processes*. *IMS Collections: Markov Processes and Related Topics* 4, pp. 195–214, doi:10.1214/074921708000000381.

[11] Winfried K. Grassmann (1991): *Finding Transient Solutions in Markovian Event Systems Through Randomization*. In: *Numerical Solution of Markov Chains*, pp. 357–371.

[12] Peter J. E. Gross & Jean Peccoud (1998): *Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets*. *Proc. Natl. Acad. Sci.* 95, pp. 6750–6755, doi:10.1073/pnas.95.12.6750.

[13] Takayuki Gunji, Sunyoung Kim, Masakazu Kojima, Akiko Takeda, Katsuki Fujisawa & Tomohiko Mizutani (2004): *PHoM – a Polyhedral Homotopy Continuation Method for Polynomial Systems*. *Computing* 73(1), pp. 57–77, doi:10.1007/s00607-003-0032-4.

[14] E. Moritz Hahn, Holger Hermanns, Björn Wachter & Lijun Zhang (2009): *INFAMY: An Infinite-State Markov Model Checker*. In: *CAV*, Springer, pp. 641–647, doi:10.1007/978-3-642-02658-4_49.

[15] E. Moritz Hahn, Holger Hermanns, Björn Wachter & Lijun Zhang (2009): *Time-Bounded Model Checking of Infinite-State Continuous-Time Markov Chains*. *Fundamenta Informaticae* 95, pp. 129–155, doi:10.3233/FI-2009-145.

[16] Joost-Pieter Katoen, Daniel Klink, Martin Leucker & Verena Wolf (2007): *Three-Valued Abstraction for Continuous-Time Markov Chains*. In: *CAV*, *Lecture Notes in Computer Science* 4590, Springer, pp. 311–324, doi:10.1007/978-3-540-73368-3_37.

[17] Daniel Klink (2010): *Three-Valued Abstraction for Stochastic Systems*. Ph.D. thesis, RWTH Aachen.

[18] T. L. Lee, T. Y. Li & C. H. Tsai (2008): *HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method*. *Computing* 83(2-3), pp. 109–133, doi:10.1007/s00607-008-0015-6.

[19] A. Loinger, A. Lipshtat, N. Q. Balaban & O. Biham (2007): *Stochastic simulations of genetic switch systems*. *Physical Review E* 75(2), p. 021904, doi:10.1103/PhysRevE.75.021904.

[20] Aad P. A. van Moorsel & William H. Sanders (1994): *Adaptive Uniformization*. *Communications in Statistics - Stochastic Models* 10(3), pp. 619–647, doi:10.1080/15326349408807313.

[21] Brian Munsky & Mustafa Khammash (2006): *The Finite State Projection Algorithm for the Solution of the Chemical Master Equation*. *Journal of Chemical Physics* 124(044104), doi:10.1063/1.2145882.

[22] Anne Remke, Boudewijn R. Haverkort & Lucia Cloth (2007): *CSL Model Checking Algorithms for QBDs*. *Theor. Comput. Sci* 382(1), pp. 24–41. Available at http://dx.doi.org/10.1016/j.tcs.2007.05.007.

[23] D. Spieler (2011): *Geobound*. *http://mosi.cs.uni-saarland.de/?page_id=74*.

[24] William J. Stewart (1994): *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press.

[25] M. Thattai & A. van Oudenaarden (2001): *Intrinsic noise in gene regulatory networks. Proceedings of the National Academy of Science, USA* 98(15), pp. 8614–8619, doi:10.1073/pnas.151588598.

[26] R. Tweedie (1975): *Sufficient conditions for regularity, recurrence and ergodicity of Markov processes. Math. Proc. Camb. Phil. Soc.* 78, pp. 125–130, doi:10.1017/S0305004100051562.