Model Checking Infinite-State Markov Chains[*]

Anne Remke, Boudewijn R. Haverkort, and Lucia Cloth

University of Twente, Faculty for Electrical Engineering,
Mathematics and Computer Science
{anne, brh, lucia}@cs.utwente.nl

Abstract. In this paper algorithms for model checking CSL (continuous stochastic logic) against infinite-state continuous-time Markov chains of so-called quasi birth-death type are developed. In doing so we extend the applicability of CSL model checking beyond the recently proposed case for finite-state continuous-time Markov chains, to an important class of infinite-state Markov chains. We present syntax and semantics for CSL and develop efficient model checking algorithms for the steady-state operator and the time-bounded next and until operator. For the former, we rely on the so-called matrix-geometric solution of the steady-state probabilities of the infinite-state Markov chain. For the time-bounded until operator we develop a new algorithm for the transient analysis of infinite-state Markov chains, thereby exploiting the quasi birth-death structure. A case study shows the feasibility of our approach.

1 Introduction

Continuous-time Markov chains are a widely spread modeling formalism for performance and dependability evaluation of computer and communication systems. Recently, various researchers have adopted CTMCs as "stochastic extension" of finite-state automata and have proposed new logics to express quantitative properties for them. Most notably is the work on CSL for CTMCs [2, 4] as stochastic extension of CTL, and the work on CSRL for Markov reward models (CTMCs enhanced with a state reward) [3]. Efficient computational algorithms have been developed for checking these models against formally specified properties expressed in these logics, cf. [3, 4], as well as supporting tools, cf. PRISM [13] and ETMC^2 [11].

All of the above work, however, has focused on *finite*-state models. In this paper we will extend model checking CSL towards *infinite*-state CTMCs. It is then possible to assess infinite-state systems, or to approximate the behavior of very large-but-finite systems. The analysis of general infinite-state CTMCs is, however, beyond reach. Therefore, we restrict the class of infinite-state CTMCs

[*] The work presented in this paper has been performed in the context of the MC=MC project (612.000.311), financed by the Netherlands Organization for Scientific Research (NWO) and is based on the diploma thesis [18], supported by the German Academic Exchange Service (DAAD).

N. Halbwachs and L. Zuck (Eds.): TACAS 2005, LNCS 3440, pp. 237–252, 2005.
© Springer-Verlag Berlin Heidelberg 2005

to the class of so-called *quasi birth-death models* (QBDs) [16], for which, despite their infinite state space, efficient algorithms exist to compute steady-state probabilities. As we will see in the course of the paper, we also require the transient, i.e., time-dependent, analysis of the infinite-state QBDs; we develop new algorithms for that purpose in this paper as well.

The paper is further organized as follows. We introduce labeled infinite-state CTMCs, and QBDs in particular, in Section 2. We then describe syntax and semantics of CSL in Section 3. Section 4 addresses in detail the model checking algorithms for the CSL operators. The feasibility of the approach is illustrated in Section 5 with a small case study, and the paper is concluded in Section 6.

2 Infinite-State CTMCs

For a fixed set of AP of atomic propositions a labeled infinite-state CTMC is defined as follows:

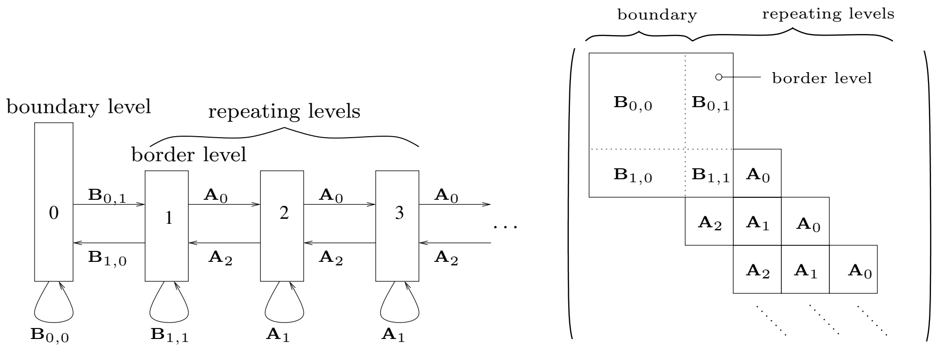
Definition 1. *A **labeled infinite-state CTMC** $\mathcal{M}$ is a tuple $(S, \mathbf{Q}, L)$ with an infinite countable set of states S, a square generator matrix*[1] $\mathbf{Q} : S \times S \to \mathbb{R}$, *and labeling function $L : S \to 2^{AP}$.*

The value $\mathbf{Q}(i, j) \geq 0$, for $i \neq j$, equals the rate at which a transition from state i to state j occurs in the CTMC, whereas $\mathbf{Q}(i, i)$ denotes the negative sum of the off-diagonal entries in the same row of $\mathbf{Q}$; its value represents the rate of leaving state i (in the sense of an exponentially distributed residence time). The labeling function L assigns to each state the set of valid atomic propositions in that state.

A special case of infinite-state CTMCs are CTMCs with quasi birth-death structure [16]. Informally speaking, the infinite state space of a QBD can be viewed as a two-dimensional strip, which is finite in one dimension and infinite in the other. Furthermore, the states in this strip can be grouped in so-called levels, according to their value or identity in the infinite dimension. Thus, the state space of a QBD consist of neighboring levels, which are all alike, except for the first one (level 0). The first level is called *boundary level* and all the others *repeating levels*. The first repeating level is sometimes called the *border level* as it separates the boundary level from the repeating levels.

Transitions, represented by positive entries in the matrix $\mathbf{Q}$, can only occur between states of the same level or between states of neighboring levels. All repeating levels have the same inter- and intra-level transition structure. The state space of a QBD can be partitioned into an infinite number of finite sets $S^j, j = \{0, 1, \ldots\}$, each containing the states of one level, such that $S = \bigcup_{j=0}^{\infty} S^j$. Figure 1(a) gives a graphical representation of a QBD, where level 0 is the boundary level and the levels from level 1 onwards are repeating levels. The inter-level transitions can be represented through matrices $\mathbf{B}_{0,1}, \mathbf{B}_{1,0}, \mathbf{A}_0, \mathbf{A}_2$, whereas

[1] Note that $\mathcal{M}$ does not contain self loops. Residence times in a CTMC obey a memoryless distribution, hence, self loops can be eliminated.



(a)
levels in a QBD

(b)
block-tridiagonal structure
of the generator matrix $\mathbf{Q}$

Fig. 1. Regular structure of QBDs

the intra-level transitions can be represented through the matrices $\mathbf{B}_{0,0}, \mathbf{B}_{1,1}$ and $\mathbf{A}_1$ (cf. Figure 1(b)).

Although QBDs are introduced here at the state level, high-level formalisms, e.g., based on stochastic Petri nets [17] or stochastic process algebras [7, 15], do exist.

Definition 2. *A* ***labeled QBD*** $\mathcal{Q}$ *of order* (N_0, N) *(with* $N, N_0 \in \mathbb{N}^+$*) is a labeled infinite-state CTMC, cf. Def. 1. The set of states is composed as* $S = \{0, \ldots, N_0 - 1\} \times \{0\} \cup \{0, \ldots, N-1\} \times \mathbb{N}^+$*, where the first part represents the boundary level with* N_0 *states, and the second part the infinite number of repeating levels, each with* N *states. The block-tridiagonal generator matrix* $\mathbf{Q}$ *consists of the following finite matrices describing the inter- and intra-level transitions:*

$\mathbf{B}_{0,0} \in \mathbb{R}^{N_0 \times N_0}$*: intra-level transition structure of the boundary level,*
$\mathbf{B}_{0,1} \in \mathbb{R}^{N_0 \times N}$*: inter-level transitions from the boundary level to the border level,*
$\mathbf{B}_{1,0} \in \mathbb{R}^{N \times N_0}$*: inter-level transitions from the border level to the boundary level,*
$\mathbf{B}_{1,1} \in \mathbb{R}^{N \times N}$*: intra-level transition structure of the border level.*
$\mathbf{A}_0 \in \mathbb{R}^{N \times N}$*: inter-level transitions from one repeating level to the next higher repeating level,*
$\mathbf{A}_1 \in \mathbb{R}^{N \times N}$*: intra-level transitions for the repeating levels*[2]*, and*
$\mathbf{A}_2 \in \mathbb{R}^{N \times N}$*: inter-level transitions from one repeating level to the next lower repeating level.*

In the following we limit ourselves to strongly connected CTMCs and to so-called *level-independent* atomic propositions. That is, if an atomic proposition $ap \in AP$ is valid in a certain state of an arbitrary repeating level, it has to be

[2] Note that $\mathbf{B}_{1,1}$ differs from $\mathbf{A}_1$ only in the diagonal entries.

valid in the corresponding states of all repeating levels. This limitation poses a true restriction on the set of formulas we are able to check. In practice, this means that a CSL formula must not refer to the level index in order to be level-independent.

Definition 3. *Let $i \in \{0, \dots, N-1\}$. An atomic proposition $ap \in AP$ is* ***level-independent*** *if and only if for all $l, k \geq 1, L(i,k) = L(i,l)$.*

An *infinite path* σ is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \dots$ with, for $i \in \mathbb{N}$, $s_i \in S$ and $t_i \in \mathbb{R}_{>0}$ such that $\mathbf{Q}(s_i, s_{i+1}) > 0$ for all i. A *finite path* σ is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots s_{l-1} \xrightarrow{t_{l-1}} s_l$ such that s_l is absorbing[3], and $\mathbf{Q}(s_i, s_{i+1}) > 0$ for all $i < l$. For an infinite path σ, $\sigma[i] = s_i$ denotes for $i \in \mathbb{N}$ the $(i+1)$st state of path σ. The time spent in state s_i is denoted by $\delta(\sigma, i) = t_i$. Moreover, with i the smallest index with $t \leq \sum_{j=0}^{i} t_j$, let $\sigma@t = \sigma[i]$ be the state occupied at time t. For finite paths σ with length $l+1$, $\sigma[i]$ and $\delta(\sigma, i)$ are defined in the way described above for $i < l$ only and $\delta(\sigma, l) = \infty$ and $\delta@t = s_l$ for $t > \sum_{j=0}^{l-1} t_j$.

$Path^{\mathcal{Q}}(s_i)$ is the set of all finite and infinite paths of the QBD $\mathcal{Q}$ that start in state s_i and $Path^{\mathcal{Q}}$ includes all (finite and infinite) paths of the QBD $\mathcal{Q}$. The superscript $\mathcal{Q}$ will be omitted whenever it is clear to which QBD the paths refer.

As for finite CTMCs, a probability measure Pr on paths can be defined [4]. Starting from there, two different types of state probabilities can be distinguished for QBDs.

The **transient state probability** is a time-dependent measure that considers the QBD at an exact time instant t. The probability to be in state s' at time instant t, given the initial state s is denoted as $\pi^{\mathcal{Q}}(s, s', t) = \Pr(\sigma \in Path(s) \mid \sigma@t = s')$. The transient probabilities are characterized by a linear system of differential equations of infinite size. Let $\underline{\pi}(t)$ be the vector of transient state probabilities at time t for all possible states (we omit the superscript $\mathcal{Q}$ as well as the starting state s for brevity here), we have $\underline{\pi}'(t) = \underline{\pi}(t) \cdot \mathbf{Q}$, given starting state s. Using a standard differential equation solver is difficult since we deal with an infinite number of differential equations. An approach using Laplace transforms and exploiting the tri-diagonal structure of the matrix $\mathbf{Q}$ has been presented in [20], however, this approach does not lead to practically feasible algorithms. Instead, it is better to resort to a technique known as uniformization, cf. [8, 9]. This will be elaborated upon in Section 4.

The **steady-state probabilities** to be in state s', given initial state s, are then defined as $\pi^{\mathcal{Q}}(s, s') = \lim_{t \to \infty} \pi^{\mathcal{Q}}(s, s', t)$, and indicate the probabilities to be in some state s' "in the long run". If steady-state is reached, the above mentioned derivatives will approach zero. Furthermore, if the QBD is ergodic, the initial state does not influence the steady-state probabilities (we therefore often write $\pi(s')$ instead of $\pi(s, s')$ for brevity). Thus, the steady-state probability vector $\underline{\pi}$ then follows from the infinite system of linear equations $\underline{\pi} \cdot \mathbf{Q} = \underline{0}$, and $\sum_s \pi_s = 1$ (normalization). This system of equations can be solved using so-called matrix-geometric methods which exploit the repetitive structure in the

[3] A state s is called absorbing if for all s' the rate $\mathbf{Q}(s, s') = 0$.

matrix $\mathbf{Q}$ [9, 16]. The idea is that the steady-state probabilities are found in a level-wise fashion, starting from the boundary and the border level. In order to do so, one first has to find the smallest square matrix $\mathbf{R}$ that satisfies the matrix-quadratic equation $\mathbf{A}_0\mathbf{R}^2 + \mathbf{A}_1\mathbf{R} + \mathbf{A}_2 = \mathbf{0}$. Efficient algorithms to do so exist, cf. [14]. Then, a system of linear equations can be set up that involves only the steady-state probabilities of the boundary and the border level, as well as a normalization equation with respect to these two levels. This system of linear equations can be solved with known iterative techniques like the Gauss-Seidel iterative method. Let $\underline{v}_0$ and $\underline{v}_1$ denote the steady-state probabilities of the first two levels, then, the matrix-geometric result [16] states that for $i = 1, 2, \cdots$, we have $\underline{v}_{i+1} = \underline{v}_i \cdot \mathbf{R}$.

A final remark should be made about stability. Since a QBD has an infinite state space, the transition rates can be such that all probability mass in steady state resides in levels that are "infinitely far away" from level 0. This, often undesirable situation, can be detected solely on the basis of the matrices $\mathbf{A}_0, \mathbf{A}_1$ and $\mathbf{A}_2$, hence, before any (expensive) computations on $\mathbf{R}$ start. Notice that in such cases, computing steady-state probabilities does not make sense; transient probabilities can still be computed.

3 The Logic CSL

We apply the logic CSL [4] on QBDs. Syntax and semantics are the same for the only difference that we now interpret the formulas over QBDs.

Syntax. Let $p \in [0, 1]$ be a real number, $\bowtie \in \{\leq, <, >, \geq\}$ a comparison operator, $I \subseteq \mathbb{R}_{\geq 0}$ a nonempty interval and AP a set of atomic propositions with $ap \in AP$.

Definition 4. *CSL state formulas Φ are defined by*

$$\Phi ::= \mathtt{tt} \mid ap \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathcal{S}_{\bowtie p}(\Phi) \mid \mathcal{P}_{\bowtie p}(\phi),$$

where ϕ is a path formula constructed by

$$\phi ::= \mathcal{X}^I \Phi \mid \Phi\, \mathcal{U}^I \Phi.$$

The steady-state operator $\mathcal{S}_{\bowtie p}(\Phi)$ denotes that the steady-state probability for a Φ-state meets the bound p. $\mathcal{P}_{\bowtie p}(\phi)$ asserts that the probability measure of the paths satisfying ϕ meets the bound p. The next operator $\mathcal{X}^I\Phi$ states that a transition to a Φ-state is made at some time instant $t \in I$. The until operator $\Phi\, \mathcal{U}^I \Psi$ asserts that Ψ is satisfied at some time instant in the interval I and that at all preceding time instants Φ holds.

Semantics. For a CSL state formula Φ on a QBD $\mathcal{Q}$, the satisfaction set contains all states of $\mathcal{Q}$ that fulfill Φ. The satisfaction set can be considered as the infinite union of finite *level satisfaction sets:* $Sat(\Phi) = Sat^0(\Phi) \cup \bigcup_{j=1}^{\infty} Sat^j(\Phi)$. $Sat^j(\Phi)$ contains only those Φ-states that are situated in level j.Satisfaction is stated in terms of a satisfaction relation $\models$, which is defined as follows.

Definition 5. *The relation* $\models$ *for states and CSL state formulas is defined as:*

$$\begin{array}{ll} s \models \mathtt{tt} \ \textit{for all } s \in S, & s \models \Phi \wedge \Psi \quad \textit{iff } s \models \Phi \textit{ and } s \models \Psi, \\ s \models ap \ \textit{ iff } ap \in L(s) & s \models \mathcal{S}_{\bowtie p}(\Phi) \ \textit{iff } \pi^{\mathcal{Q}}(s, Sat(\Phi)) \bowtie p, \\ s \models \neg\Phi \ \textit{iff } s \not\models \Phi & s \models \mathcal{P}_{\bowtie p}(\phi) \ \textit{iff } Prob^{\mathcal{Q}}(s, \phi) \bowtie p. \end{array}$$

where $\pi^{\mathcal{Q}}(s, Sat(\Phi)) = \sum_{s' \in Sat(\Phi)} \pi^{\mathcal{Q}}(s, s')$*, and* $Prob^{\mathcal{Q}}(s, \phi)$ *describes the probability measure of all paths* $\sigma \in Path(s)$ *that satisfy* ϕ *when the system is starting in state* s*, that is,* $Prob^{\mathcal{Q}}(s, \phi) = Pr\{\sigma \in Path^{\mathcal{Q}}(s) \mid \sigma \models \phi\}$.

Definition 6. *The relation* $\models$ *for paths and* CSL^{∞} *path formulas is defined as:*

$$\begin{array}{ll} \sigma \models \mathcal{X}^I \Phi & \textit{iff } \sigma[1] \textit{ is defined and } \sigma[1] \models \Phi \textit{ and } \delta(\sigma, 0) \in I, \\ \sigma \models \Phi\, \mathcal{U}^I \Psi & \textit{iff } \exists t \in I\ (\sigma@t \models \Psi \wedge (\forall t' \in [0, t)(\sigma@t' \models \Phi))). \end{array}$$

4 Model Checking Algorithms

In order to develop a model checking algorithm for QBDs, we need to focus on the connection between the validity of state formulas and the special birth-death structure of QBDs. At first glance, one could think that in corresponding states of all repeating levels the same CSL formulas hold. Model checking a QBD would then be reducible to model checking the boundary level and one repeating level representative for all others. Unfortunately this is not the case, as can be explained considering the time-bounded next and until operator. In order to check CSL properties that contain these path formulas, we need to examine all possible paths in a level-wise fashion. Considering time-bounded next, note that in the border level other next-formulas might be satisfied than in the other repeating levels, because the boundary level is still reachable from the border level but not from any other repeating level. Thus, if we want to check for example the formula $\phi = \mathcal{X}^I$ *red* and the property *red* is only valid in the boundary level, this property ϕ can be fulfilled by a path starting in the border level, but not when starting in any other repeating level. A similar reasoning holds for the until operator, where not only the border level is concerned but even more repeating levels, because with the until operator not just one step is considered, but potentially infinitely many. Thus, for path-formulas no two repeating levels can a priori be considered the same.

4.1 Level Independence of CSL Formulas

Even though CSL formulas are not level independent in general, their validity does not change arbitrarily between levels. Remember that we assume level independence of atomic propositions for the QBDs we consider. For CSL formulas, we generalize the idea of level independence: we only require that the validity in a state is level independent for repeating levels with an index of at least k. Thus, we allow the validity of a CSL formula to change between corresponding states of repeating levels, but only up to repeating level $k - 1$. From level k onwards, the validity must remain unchanged.

Definition 7. *Let $\mathcal{Q}$ be a QBD of order (N_0, N). A CSL state formula Φ is* level independent as of level $k \geq 1$ *(in QBD $\mathcal{Q}$) if and only if for levels above and including k, the validity of Φ in a state does not depend on the level, that is,*

$$\textit{for all } i \in \{0, \dots, N-1\} \textit{ and for all } l \geq k : \quad (i,l) \models \Phi \iff (i,k) \models \Phi.$$

The following proposition states, under the assumption of level independent atomic propositions, that such a k exists for any CSL state formula.

Proposition 1 Let $\mathcal{Q}$ be a QBD with level independent atomic properties and let Φ be a CSL state formula. Then there exists a $k \in \mathbb{N}$, such that Φ is level independent as of level k in $\mathcal{Q}$.

We will justify this proposition inductively in the sections that discuss the model checking of the different types of CSL state formulas.

For model checking a property Φ, we will compute the set $Sat(\Phi)$ with a recursive descent procedure over the parse tree of Φ. To do so, the CSL formula Φ is split into its sub-formulas and for every sub-formula the model checker is invoked recursively. For a state formula Φ that is level independent as of level k, cf. Definition 7, only the first k level satisfaction sets have to be computed. $Sat^k(\Phi)$ then acts as a representative for all following levels. In what follows we discuss the required computations for one such invocation, for each of the operators in the logic CSL.

4.2 Atomic Propositions and Logical Operators

Computing the satisfaction set for an atomic proposition ap proceeds as follows. $Sat^0(ap)$ consists of those states of the boundary level where ap is contained in the labeling. We test all states in the border level in order to obtain $Sat^1(ap)$, and, hence, $Sat^j(ap)$ for $j \geq 1$ (as per Definition 3).

Let Φ be a CSL state formula that is level independent as of level k. Its negation $\neg\Phi$ is clearly also level independent as of level k. The level satisfaction sets of $\neg\Phi$ are computed by complementing the corresponding satisfaction set of Φ:

$$Sat^j(\neg\Phi) = S^j \backslash Sat^j(\Phi), \text{for all } j \geq 0.$$

Let Φ and Ψ be two CSL state formulas, level independent as of level k_Φ and k_Ψ, respectively. The conjunction $\Phi \wedge \Psi$ is level independent as of level $\max(k_\Phi, k_\Psi)$. The level satisfaction sets are computed by intersecting the corresponding satisfaction sets of Φ and Ψ:

$$Sat^j(\Phi \wedge \Psi) = Sat^j(\Phi) \cap Sat^j(\Psi), \text{for all } j \geq 0.$$

4.3 Steady-State Operator

A state s satisfies $\mathcal{S}_{\bowtie p}(\Phi)$ if the accumulated steady state probability of all Φ-states reachable from s meets the bound p. Since we assume a strongly connected QBD, the steady-state probabilities are independent of the starting state. It follows that either all states satisfy a steady-state formula or none of the states does, which implies that a steady-state formula is always level independent as of level 1. We first determine the satisfaction set $Sat(\Phi)$ and then compute the accumulated steady-state probability. If the accumulated steady-state probability meets the bound p, we have $Sat(\mathcal{S}_{\bowtie p}(\Phi)) = S$, otherwise, $Sat(\mathcal{S}_{\bowtie p}(\Phi)) = \varnothing$. Exploiting the special structure of QBDs, the accumulated probability is given by

$$\pi(Sat(\Phi)) = \sum_{s \in Sat(\Phi)} \pi(s) = \sum_{j=0}^{\infty} \sum_{s \in Sat^j(\Phi)} \underline{v}_j(s),$$

where the vectors $\underline{v}_j = (\cdots, \underline{v}_j(s), \cdots)$ can be computed one after the other, using the matrix-geometric method, as discussed in Section 2.

In order to deal with the infinite sum we iterate through the repeating levels and accumulate the steady-state probabilities in a level-wise fashion. Denote with $\tilde{\pi}^l(Sat(\Phi))$ the accumulated steady-state probabilities of all Φ-states up to level l, that is,

$$\tilde{\pi}^l(Sat(\Phi)) = \sum_{j=0}^{l} \sum_{s \in Sat^j(\Phi)} \underline{v}_j(s).$$

Starting with $l = 0$, we iterate through the levels and compute $\tilde{\pi}^l(Sat(\Phi))$ and $\tilde{\pi}^l(Sat(\neg\Phi))$, respectively. The computation of the steady-state probabilities of $\neg\Phi$-states introduces no additional cost, since they are computed anyway. In every step we have to check whether we can already decide on the validity of the steady-state formula $\mathcal{S}_{\bowtie p}(\Phi)$. The following implications hold:

$$\begin{aligned} &\text{(a)} \qquad \tilde{\pi}^j(Sat(\Phi)) > p \Rightarrow \pi(Sat(\Phi)) > p, \\ &\text{(b)}\ \tilde{\pi}^j(Sat(\neg\Phi)) > 1 - p \Rightarrow \pi(Sat(\Phi)) < p. \end{aligned}$$

As soon as one of the left hand side inequalities becomes true, the iteration stops. For the interpretation we distinguish the cases $\mathcal{S}_{<p}(\Phi)$ and $\mathcal{S}_{>p}(\Phi)$. For $\mathcal{S}_{\geq p}(\Phi)$ or $\mathcal{S}_{\leq p}(\Phi)$ the equations need to be modified accordingly. For $\mathcal{S}_{<p}(\Phi)$ the interpretation is as follows. If inequality (a) holds, the condition $\pi(Sat(\Phi)) < p$ is clearly not accomplished and $Sat(\mathcal{S}_{<p}(\Phi)) = \varnothing$. If inequality (b) holds, the condition $\pi(Sat(\Phi)) < p$ is accomplished and $Sat(\mathcal{S}_{<p}(\Phi)) = S$. For $\mathcal{S}_{>p}(\Phi)$ the same conditions need to be checked in every iteration step j, but they need to be interpreted differently. If inequality (a) holds, the probability bound is met and $Sat(\mathcal{S}_{>p}(\Phi)) = S$. If inequality (b) holds, the bound is not met and $Sat(\mathcal{S}_{>p}(\Phi)) = \varnothing$.

The satisfaction set of Φ might be finite. For a CSL formula Φ that is level independent as of level k, this is the case when no state in level k satisfies Φ. The iteration then ends at level $k - 1$ and $\pi(Sat(\Phi)) = \tilde{\pi}^{k-1}(Sat(\Phi))$. In case $Sat(\Phi)$ is infinite, the iterations stop as soon as one of the inequalities is satisfied. Unfortunately, if the probability p is exactly equal to the steady-state probability $\pi(Sat(\Phi))$, the approximations $\tilde{\pi}^l(Sat(\Phi))$ and $\tilde{\pi}^l(Sat(\neg\Phi))$ will never fulfill one of the inequalities. In an implementation of this algorithm some care must be taken to detect this case in order to avoid a non-stopping iteration.

Instead of the just-sketched iterative process, we can also develop a closed-form matrix expression for the probability $\pi(Sat(\Phi))$ by exploiting properties of the matrix-geometric solution, i.e., by using the fact that $\sum_i \mathbf{R}^i = (\mathbf{I} - \mathbf{R})^{-1}$. In doing so, the infinite summation disappears, however, it comes at the cost of a required matrix inversion. In practice, this is therefore not always a more efficient approach, but it avoids the stopping problem.

4.4 Time-Bounded Next Operator

Recall that a state s satisfies $\mathcal{P}_{\bowtie p}(\mathcal{X}^I \Phi)$ if the one-step probability to reach a state that fulfills Φ within a time in $I = [a, b]$, outgoing from s meets the bound p, that is,

$$\begin{aligned} s \models \mathcal{P}_{\bowtie p}(\mathcal{X}^I \Phi) \;&\Leftrightarrow\; \Pr\{\sigma \in \mathit{Path}(s) \mid \sigma \models \mathcal{X}^I \Phi\} \;\bowtie p \\ &\Leftrightarrow \left(\left(e^{Q(s,s)\cdot a} - e^{Q(s,s)\cdot b} \right) \cdot \sum_{\substack{s' \in \mathit{Sat}(\Phi) \\ s' \neq s}} \frac{Q(s,s')}{-Q(s,s)} \right) \bowtie p, \end{aligned} \tag{1}$$

where $e^{Q(s,s)\cdot a} - e^{Q(s,s)\cdot b}$ is the probability of residing at s for a time in I, and $\frac{Q(s,s')}{-Q(s,s)}$ specifies the probability to step from state s to state s'. Note that the above inequality contains a (possibly infinite) summation over all Φ-states. However, we only need to sum over the states of $\mathit{Sat}(\Phi)$ that are reachable from s in one step. That is, for $s = (i, j)$, we only have to consider the Φ-states from levels $j-1, j$, and $j+1$. For all states of all other levels the one-step probabilities are zero anyway. The infinite set $\mathit{Sat}(\Phi)$ ruling the summation in (1) can thus be replaced by the finite set $\mathit{Sat}_{\mathcal{X},(i,j)}(\Phi)$ containing only the states from level $j-1, j, j+1$ that fulfill Φ, that is,

$$\mathit{Sat}_{\mathcal{X},(i,j)}(\Phi) = \begin{cases} \mathit{Sat}^0(\Phi) \cup \mathit{Sat}^1(\Phi), & j = 0, \\ \mathit{Sat}^{j-1}(\Phi) \cup \mathit{Sat}^j(\Phi) \cup \mathit{Sat}^{j+1}(\Phi), & \text{otherwise}. \end{cases}$$

Now, let the inner formula Φ of the next-formula be level independent as of level k. Hence, the validity of the state formula $\mathcal{P}_{\bowtie p}(\mathcal{X}^I \Phi)$ might be different in corresponding states for all levels up to $k-1$. Therefore, unfortunately, level k can still have different states satisfying $\mathcal{P}_{\bowtie p}(\mathcal{X}^I \Phi)$ since level $k-1$ is reachable in one step. But, as of level $k+1$, only levels can be reached where the validity of state formula Φ is equal for corresponding states. Hence, if Φ is level independent as of level k, $\mathcal{P}_{\bowtie p}(\mathcal{X}^I \Phi)$ is level independent as of level $k+1$. For the construction of the satisfaction set of such a formula, we therefore have to compute explicitely the satisfying states up to level $k+1$. Subsequently, $\mathit{Sat}^{k+1}(\mathcal{P}_{\bowtie p}(\mathcal{X}^I \Phi))$ can be seen as a representative for all following repeating levels.

4.5 Time-Bounded Until Operator

For model checking $\mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^I \Psi)$ we adopt the general approach for finite CTMCs [4]. The idea is to use a transformed QBD where several states are made absorbing. We focus on the case where $I = [0, t]$. The CSL path formula $\varphi = \Phi\, \mathcal{U}^{[0,t]} \Psi$ is valid if a Ψ-state is reached on a path, before time t via only Φ-states. As

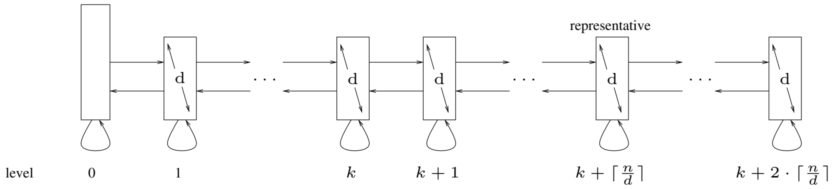

representative
d
d
d
d
d
level
0
1
k
$k+1$
$k+\lceil \frac{n}{d} \rceil$
$k+2\cdot\lceil \frac{n}{d} \rceil$

Fig. 2. Finite fraction of the QBD needed for the transient solution

soon as a Ψ-state is reached, the future behavior of the QBD is irrelevant for the validity of φ. Thus all Ψ-states can be made absorbing without affecting the satisfaction set of formula φ. On the other hand, as soon as a $(\neg\Phi \wedge \neg\Psi)$ state is reached, φ will be invalid, regardless of the future evolution of the system. As a result we may switch from checking the Markov chain $\mathcal{Q}$ to the Markov chain $\mathcal{Q}[\Psi][\neg\Phi \wedge \neg\Psi] = \mathcal{Q}[\neg\Phi \vee \Psi]$, where the states satisfying the formula in $[\cdot]$ are made absorbing. Model checking a formula involving the until operator then reduces to calculating the transient probabilities $\pi^{Q[\neg\Phi\vee\Psi]}(s, s', t)$ for all Ψ-states s'. Exploiting the special structure of QBDs yields

$$\begin{aligned} s \models \mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{[0,t]}\Psi) &\Leftrightarrow \mathit{Prob}^{Q}(s, \Phi\, \mathcal{U}^{[0,t]}\Psi) \ \bowtie\ p \\ &\Leftrightarrow \left(\sum_{i=0}^{\infty} \sum_{s' \in Sat^i(\Psi)} \pi^{Q[\neg\Phi\vee\Psi]}(s, s', t) \right) \bowtie p. \end{aligned}$$

Making the QBD Finite. Uniformization [8] is an often used method to compute transient probabilities in finite CTMCs. The continuous-time model is transformed into a discrete-time model together with a Poisson process with rate λ. The uniformization constant λ must be at least equal to the maximum absolute value of the diagonal entries of the generator $\mathbf{Q}$. Since for a QBD the matrix $\mathbf{Q}$ has only finitely many different diagonal entries (originating from the matrices $\mathbf{B}_{0,0}, \mathbf{B}_{1,1}$, and $\mathbf{A}_1$), λ can be determined even though $\mathbf{Q}$ has an infinite number of entries. For an allowed numerical error ε_t, uniformization requires a finite number n of steps (state changes) to be taken into account in order to compute the transient probabilities. Note that n can be computed *a priori*, given ε_t, λ and t.

Let $d \geq 1$ be the so-called *level diameter*, that is, the minimum number of state transitions that is needed to cross a complete repeating level. If n steps are to be taken into account, only $\lceil \frac{n}{d} \rceil$ levels can be reached from a state in level l in either direction.

Thus, for model checking the formula $\mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{[0,t]}\Psi)$, first all $\neg\Phi \vee \Psi$-states have to be made absorbing. If $\neg\Phi \vee \Psi$ is level-independent as of level k, then, using uniformization with n steps, we obtain the same transient probabilities for corresponding states as of level $k + \lceil \frac{n}{d} \rceil$, since only equivalent repeating levels are seen when stepping through the QBD.

In order to compute the transient probabilities for all states of the QBD, it suffices to compute them for the first $k + \lceil \frac{n}{d} \rceil$ levels only. Hence, only a finite part of the infinite QBD is needed. Outgoing from level $k + \lceil \frac{n}{d} \rceil$ there must still be the possibility to undertake $\lceil \frac{n}{d} \rceil$ steps "to the right". The total number of levels we have to consider therefore is $k + 2 \cdot \lceil \frac{n}{d} \rceil$ (cf. Figure 2). Thus, we reduced the task of computing transient probabilities for an infinite QBD to the computation of transient probabilities in a finite CTMC.

Interpretation of the Transient Probabilities. For all states in the first $k + \lceil \frac{n}{d} \rceil$ levels, we add the computed transient probabilities to reach any Ψ-state and check whether the accumulated probability meets the bound p. When using uniformization, the computed accumulated probability

$$\tilde{\pi}^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) = \sum_{s' \in Sat(\Psi)} \tilde{\pi}^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, s', t)$$

is always an underestimation of the actual probability. Fortunately, we are able to indicate a maximum error ε_m (depending on ε_t) such that

$$\pi^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) \leq \tilde{\pi}^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) + \varepsilon_m.$$

The value of ε_m decreases as n increases. Applying the above inequality, we obtain the following implications:

$$\begin{array}{ll} \text{(a)} & \tilde{\pi}^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) > p \Rightarrow \pi^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) > p \\ \text{(b)} & \tilde{\pi}^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) < p - \varepsilon_m \Rightarrow \pi^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) < p \end{array}$$

If one of these inequalities (a) or (b) holds, we can decide whether the bound $< p$ or $> p$ is met. For the bounds $\leq p$ and $\geq p$, similar implications can be derived. If $\tilde{\pi}^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t) \in [p, p - \varepsilon_m]$, then we cannot decide whether $\pi^{\mathcal{Q}[\neg\Phi\vee\Psi]}(s, Sat(\Psi), t)$ meets the bound p. The number of steps n considered when computing the transient probabilities via uniformization has been too small in that case. Decreasing ε_t, hence, increasing n, might resolve this problem.

As already mentioned, for all levels $\geq k + \lceil \frac{n}{d} \rceil$, the transient probabilities computed with n steps will be the same. If we can decide whether the bound p is met (case (a) or (b) above), we can be sure that $\mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{[0,t]} \Psi)$ is level independent as of level $k + \lceil \frac{n}{d} \rceil$. It might actually be the case that level independence starts at a smaller level.

If n is large enough we check for all states in levels up to $k + \lceil \frac{n}{d} \rceil$ whether the accumulated transient probability of reaching a Ψ-state meets the bound p. These states form the subsequent level satisfaction sets $Sat^j(\mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{[0,t]} \Psi))$. The satisfaction set for level $k + \lceil \frac{n}{d} \rceil$ is representative for all following levels.

The more general case where $I = [t_1, t_2]$ for $0 < t_1 < t_2$ can be treated by following the procedure given in [4]. It requires the computation of transient probabilities in two "versions" of the QBD, where different states are made absorbing. The number of levels to be considered must be adapted accordingly. Details of this procedure are omitted for brevity here, but can be found in [18].

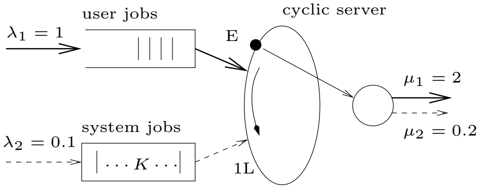


Fig. 3. Dual-Job-Class Cyclic-Server System

The case where $I = [0, \infty]$ can be addressed similarly as in the finite-state case, cf. [4–Corollary 1], except for the fact that it leads to a system of linear equations of infinite size. Given the special (QBD) structure of this system of linear equations, it appears that also in this case a matrix-geometric solution approach might be applicable, but this remains to be investigated.

Complexity. For model checking the until operator we need to consider $k + \frac{2n}{d}$ levels with N states, respectively N_0 states for the boundary level. ν denotes the average number of transitions originating from a single state of the QBD. To compute the transient probabilities we require the sum of $\mathcal{O}(\lambda t)$ matrices, each of which is the result of a matrix-matrix multiplication. This results in an overall computational complexity of $\mathcal{O}(\lambda t \cdot \nu(N_0 + kN + nN)^2)$. Regarding storage complexity, we require $\mathcal{O}(3(N_0 + kN + nN))$ storage for the probability matrices and $\mathcal{O}(\nu(N_0^2 + NN_0 + N^2))$ for the transition matrix of the underlying DTMC.

5 Case Study: A Dual-Job-Class Cyclic-Server System

System Description. We analyze a system with two sorts of jobs, as depicted in Figure 4. User jobs, having high priority, are served according to an exhaustive scheduling strategy. System jobs, having low priority, are served with a 1-limited scheduling strategy. In the beginning, the server always starts serving user jobs and a system job can only be served after at least one user job has been served. As long as there are user jobs in the queue, the server first serves these jobs. System jobs can only be served, if all user jobs have been served and at least one system job is waiting for service. If the server changes to system jobs, only one job is served and afterwards the server polls the user jobs queue again. We can have an infinite number of user jobs and at most K system jobs in the system. We have modeled this system as iSPN [17]; from this iSPN the underlying QBD is automatically generated. The order of this QBD depends on the actual value of K; each level of the underlying QBDs consists of $2K + 1$ states that model the number of system jobs in the queue and the presence of the server at the system-job queue. The QBD for $K = 1$ is given in Figure 4.

Its states can be interpreted as follows: j indicates the number of user jobs currently in the system, $i = 0$ means that a system job is being served, $i = 1$ means that no system job is waiting, and $i = 2$ means that a system job just arrived but is not being served yet.

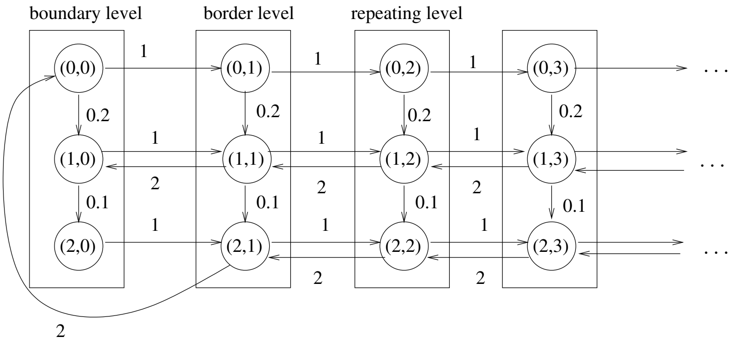


Fig. 4. QBD of the Dual-Job-Class Cyclic-Server System

Steady-State Property. We want to know whether the steady-state probability of having a full system-job queue is greater than 0.1. As CSL formula, this property can be stated as $\mathcal{S}_{>0.1}(\mathtt{Q}_{\mathtt{Sys}}\mathtt{full})$, where $\mathtt{Q}_{\mathtt{Sys}}\mathtt{full}$ is the atomic proposition that is valid in all states where the system-job queue is full.

For any K, every level contains exactly two states satisfying atomic proposition $\mathtt{Q}_{\mathtt{Sys}}\mathtt{full}$, being the state with K system jobs present (queued), and with the server active with either a system job or a user job. In case of $K = 1$, we see in Figure 4 that these are the states $(0, \cdot)$ and $(2, \cdot)$. Hence, $Sat(\mathtt{Q}_{\mathtt{Sys}}\mathtt{full})$ has infinite size. For $K < 11$, $Sat(\mathcal{S}_{>0.1}(\mathtt{Q}_{\mathtt{Sys}}\mathtt{full})) = S$, thus, the formula holds in all states. For $K \geq 11$, $Sat(\mathcal{S}_{>0.1}(\mathtt{Q}_{\mathtt{Sys}}\mathtt{full})) = \varnothing$.

Figure 5 shows the number of iterations (as discussed in Section 4.3), needed to verify the property, depending on the system parameter K. If the actual steady-state probability of $\mathtt{Q}_{\mathtt{Sys}}\mathtt{full}$-states comes close to the given bound 0.1, more iterations are needed. This explains the peak at $K = 11$. Figure 5 also gives the computation time for different K. Note that the smaller number of iterations for $K > 11$ does not lead to a smaller computation time, since more time is needed per iteration (as the matrices become larger).

Time-Bounded Until Property. As system jobs have a low priority compared to the user jobs, we would like to know for which states of the QBD the probability of the system-job queue to become empty in a certain time interval is greater than 0.1. Stated in CSL, we analyze $\Phi = \mathcal{P}_{>0.1}(\neg \mathtt{Q}_{\mathtt{Sys}}\mathtt{Empty}\, \mathcal{U}^{[0,t]} \mathtt{Q}_{\mathtt{Sys}}\mathtt{Empty})$.

For $K = 1$ the formula Φ can be interpreted as follows: Is the probability greater than 0.1 that a waiting system job is served in a certain time interval? For a time interval $I = [0, 2]$, a given error $\varepsilon = 10^{-7}$, uniformization considers 23 steps. As Φ is level-independent as of level 1 and we have a level-diameter of 1, level 24 can serve as a representative for the higher repeating levels. Analyzing the resulting satisfaction set $Sat(\mathcal{P}_{>0.1}(\neg \mathtt{Q}_{\mathtt{Sys}}\mathtt{Empty}\, \mathcal{U}^{[0,2]} \mathtt{Q}_{\mathtt{Sys}}\mathtt{Empty}))$ shows the following.

All states with first index $i = 1$ are trivially included in the satisfaction set, because $\mathtt{Q}_{\mathtt{Sys}}\mathtt{Empty}$ is already valid in these states. States with first index $i = 0$ are included as they model a situation in the system where the server is serving

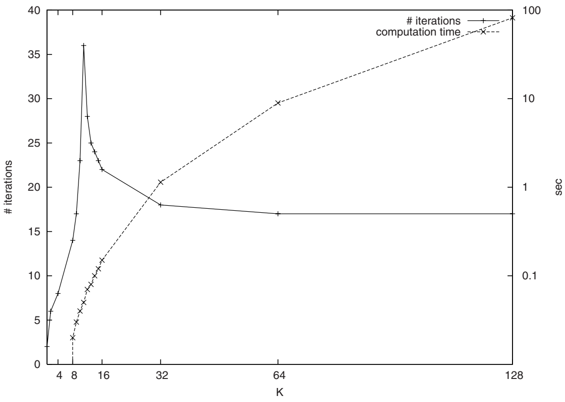


Fig. 5. Number of iterations and computation time required for checking $\mathcal{S}_{>0.1}(\mathtt{Q_{Sys}full})$, as a function of the maximum number of system jobs K

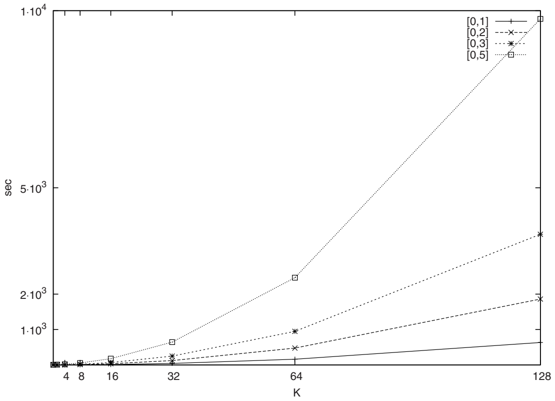


Fig. 6. Computation time required for checking $\mathcal{P}_{>0.1}(\neg\mathtt{Q_{Sys}Empty}\,\mathcal{U}^{[0,t]}\mathtt{Q_{Sys}Empty})$, as a function of the maximum number of system jobs K

a system job. Hence, for those states the probability for the system job to be served in time interval $[0, 2]$ is greater than 0.1. If the system job just arrived in the queue ($i = 2$), model checking shows that the probability for this job to be served in time is only greater than 0.1 if less than three user jobs are waiting for service.

For the computation of the satisfaction sets, we have to deal with state spaces of the size $(2K+1)\cdot(2n+2)$. The left-hand term accounts for the size of one level and the right-hand term for the number of levels considered by uniformization. n gives the number of steps which is considered by uniformization, depending on the error ε_t. In Figure 6 the computation time is depicted for different time intervals. For larger time intervals the state space grows as uniformization needs to consider more steps which results in larger computation times.

6 Conclusions

In this paper we have presented model-checking algorithms for checking CSL properties against infinite-state CTMCs, in particular for QBDs. The model checking algorithms make extensive use of uniformization for transient analysis (for time-bounded until) and matrix-geometric methods for determining steady-state probabilities (for the steady-state operator). The model checking algorithms as presented are new. Our approach to analyze the transient state probabilities of infinite-state CTMC is also new. We have shown the feasibility of the model checking algorithms by a case study.

We are aware of the fact that when checking nested formulas, the number of levels that have level-dependent properties grows, which makes the algorithms less efficient. On the other hand, practice reveals that the nesting depth of logical expressions to be checked is typically small [6], so that this is not so much of a disadvantage after all.

At various points, the presented algorithms can be made more efficient. For instance, for checking time-bounded until we have introduced the notion of level diameter. In practice, there might be two different diameters, depending on the direction of crossing a level (to higher or to lower levels). Exploiting this fact might lead to smaller finite-state Markov chains to be considered.

We also required the QBD under study to be strongly connected, in order to make use of the fact that the steady-state probabilities do not depend on the starting state. It is left for further investigation how the model checking algorithms have to be adapted to account for non-strongly connected QBDs.

By restricting ourselves to level-independent formulas, we restrict the set of CSL formulas that can be checked. For model checking level-*dependent* CSL formulas new model checking algorithms will be needed, since in that case we cannot exploit the level-independent QBD structure to cut the infinite set of states.

We note that there has been done a substantial amount of work on model checking infinite-state systems, e.g., on regular model checking [1] and probabilistic lossy channel systems [19], however, not in the context of continuous-time Markov chains, as we have presented here. It remains to be investigated whether and how we can exploit these results in our context.

Finally, we need to complete our work on the tool chain for specifying and model checking infinite-state systems, and possibly will integrate it into other model checking tools for CTMCs. First details on this, and on many of the other issues addressed in this paper, can be found in the recently completed diploma thesis [18].

References

1. P. Abdulla, B. Jonsson, M. Nilsson, and M. Saksena. A survey of regular model checking. In P. Gardner and N. Yoshida, editors, *Proc. Concur 2004*, number 3170 in Lecture Notes in Computer Science, pages 35–48, 2004.
2. A. Aziz, K. Sanwal, and R. Brayton. Model checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.

3. C. Baier, B.R. Haverkort, H. Hermanns, and J.-P. Katoen. On the logical characterisation of performability properties. In *Proc. 27th Int. Colloquium on Automata, Languages and Programming (ICALP'00)*, number 1853 in Lecture Notes in Computer Science, pages 780–792, 2000.
4. C. Baier, B.R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):524–541, July 2003.
5. A. Bell. *Distributed evaluation of stochasic Petri nets.* PhD thesis, Dept. of Computer Science, RWTH Aachen, 2004.
6. M.B. Dwyer, G.S. Avrunin, and J.C. Corbett. Patterns in property specification for finite-state verification. In *Proc. 21st Int. Conf. on Software Engineering*, pages 411–420. IEEE CS Press, 1999.
7. A. El-Rayes, M. Kwiatkowska, and G. Norman. Solving infinite stochastic process algebra models through matrix-geometric methods. In *Proc. 7th Process Algebras and Performance Modelling Workshop (PAPM'99)*, pages 41–62. University of Zaragoza, 1999.
8. D. Gross and D.R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.
9. B.R. Haverkort. *Performance of Computer Communication Systems.* John Wiley & Sons, 1998.
10. B.R. Haverkort, H. Hermanns, and J.-P. Katoen. On the use of model checking techniques for dependability evaluation. In *Proc. 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00)*, pages 228–237. IEEE CS Press, 2000.
11. H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle. A tool for model-checking Markov chains. *International Journal on Software Tools for Technology Transfer*, 4(2):153–172, 2003.
12. J.-P. Katoen. *Concepts, Algorithms, and Tools for Model Checking.* Arbeitsberichte des IMMD 32(1), Friedrich-Alexander Universität Erlangen Nürnberg, June 1999.
13. M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: a hybrid approach. *International Journal on Software Tools for Technology Transfer*, 6(2):128–142, 2004.
14. G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi birth and death processes. *Journal of Applied Probability*, 30:650–674, 1993.
15. I. Mitrani, A. Ost, and M. Rettelbach. TIPP and the spectral expansion method. In F. Baccelli, A. Jean-Marie, and I. Mitrani, editors, *Quantitative Models in Parallel Systems*, pages 99–113. Springer, 1995.
16. M.F. Neuts. *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach.* Johns Hopkins University Press, 1981.
17. A. Ost. *Performance of Communication Systems. A Model-Based Approach with Matrix-Geometric Methods.* PhD thesis, Dept. of Computer Science, RWTH Aachen, 2001.
18. A. Remke. Model Checking Quasi Birth Death Processes. Master's thesis, Dept. of Computer Science, RWTH Aachen, 2004
(`http://www.cs.utwente.nl/∼anne/pub/modelchecking.pdf`).
19. Ph. Schnoebelen. The verification of probabilistic lossy channel systems. In C. Baier, B.R. Haverkort, H. hermanns, J.-P. Katoen, and M. Siegle, editors, *Validation of Stochastic Systems*, volume 2925 of *Lecture Notes in Computer Science*, pages 445–465, 2004.
20. J. Zhang and E.J. Coyle. Transient analysis of quasi-birth-death processes. *Stochastic Models*, 5(3):459–496, 1989.