Beyond Model-Checking CSL for QBDs: Resets, Batches and Rewards

Anne Remke and Boudewijn R. Haverkort
Design and Analysis of Communication Systems
University of Twente
[anne,brh]@cs.utwente.nl

Abstract

We propose and discuss a number of extensions to quasi-birth-death models (QBDs) for which CSL model checking is still possible, thus extending our recent work [12] on CSL model checking of QBDs. We then equip the QBDs with rewards, and discuss algorithms and open research issues for model checking CSRL for QBDs with rewards.

1 Introduction

Over the last few years, considerable progress has been made in model checking algorithms for finite Markov chains. The logic CSL (continuous stochastic logic) has been shown to be suitable for the specification of performance or dependability measures for finite Markov chains [4, 1, 7]. Furthermore, efficient uniformization-based algorithms have been proposed to make such model checking practically feasible [2]. The logic CSRL (continuous stochastic reward logic) has been proposed to reason about time *and* rewards in finite Markov chains [3]. As such, CSRL comprises a natural way of specifying performability properties of systems. In [6] we proposed a number of numerical algorithms to efficiently model check finite Markov-reward models. Recently, we considered the step of applying CSL model checking algorithms for **in**finite-state Markov chains, in particular for so-called quasi-birth-death models (QBDs) [12]. We showed under which restrictions we can automatically model check performance or dependability properties specified in CSL for QBDs. Matrix-geometric methods were used to evaluate CSL properties involving the steady-state operator, as well as an uniformization-based procedure to evaluate CSL properties involving the time-bounded until operator. In the current paper we propose a number of extensions for the model checking of QBDs. First of all, we extend the QBD model class such that we can still apply the procedures we have derived so far. This encompasses the case where we allow the QBD to have transitions from any level to the boundary level, e.g., modeling a complete loss of buffer content due to failures, as well as the case where transitions can take place between non-neighboring levels, albeit in a regular way. We also alleviate a limitation we previously had in CSL model checking of QBDs: we can allow for atomic properties that are not level-independent as long as they are periodic. These extensions make the CSL model checking framework for QBDs more versatile. The second group of extensions we propose relates to the inclusion of rewards, that is, in the use of the logic CSRL (including a number of earlier-proposed extensions) instead of just CSL for model checking QBDs with rewards. This will result in a model checking that allows for performability evaluation of QBDs.

2 Background: QBDs and CSL

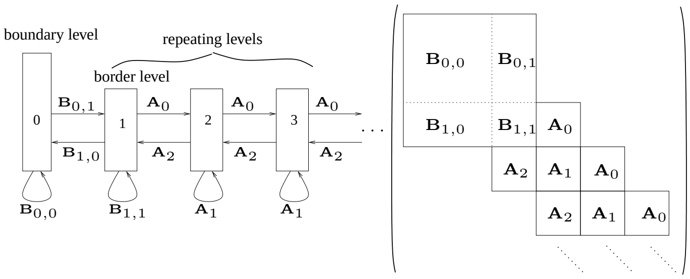
Quasi-birth-death models The infinite state space of a QBD can be viewed as a two-dimensional strip, which is finite in one dimension and infinite in the other. Figure 1(a) gives a graphical representation of a QBD. Formally, a *labeled QBD* $\mathcal{Q}$ of order (N_0, N) (with $N_0, N \in \mathbb{N}^+$) is a labeled infinite-state CTMC. The set of states is composed as $S = \{0, \cdots, N_0 - 1\} \times \{0\} \cup \{0, \cdots, N-1\} \times \mathbb{N}^+$, 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 $\mathbf{Q}$ is composed out of eight finite matrices describing the inter- and intra-level transitions as shown in Figure 1(b). $\mathbf{Q}$ describes an infinite-state CTMC, $\{X_t \in S \mid t/geq0\}$. The steady-state probabilities of a QBD can be calculated in a level-wise fashion, using e.g., *matrix-geometric methods* [9, 11], which exploit the repetitive structure in the generator matrix. To compute transient state probabilities for the infinite-state QBDs, we developed an uniformization-based approach [12].

Continuous stochastic logic The logic CSL [2] allows for state formulas Φ of the form

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

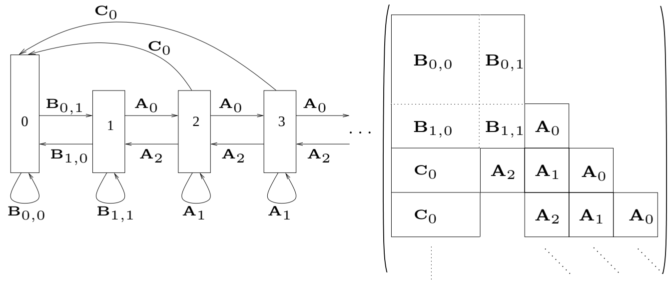
where ϕ is a so-called CSL path formula, which may take the following form: $\phi ::= \mathcal{X}^I \Phi \mid \Phi\, \mathcal{U}^I \Phi$. The next operator $\mathcal{X}^I \Phi$ states that a transition to a Φ-state is made at some time instant $t \in I$. The until operator $\Phi\, \mathcal{U}^I \Psi$ asserts that Ψ is satisfied at some time instant in $t \in I$ and that at all preceding time instants Φ holds. For a CSL formula Φ, the satisfaction set contains all states that fulfill Φ.

CSL model checking of QBDs We have applied the logic CSL to QBDs with so-called *level-independent atomic propositions*: an atomic proposition that is valid in a certain state of an arbitrary repeating level, has to be valid in the corresponding states of all repeating levels. Even though



(a) levels in a QBD (b) generator matrix

Figure 1. Quasi-Birth-Death Process



(a) transitions in a QBD with resets (b) generator matrix

Figure 2. QBD with resets

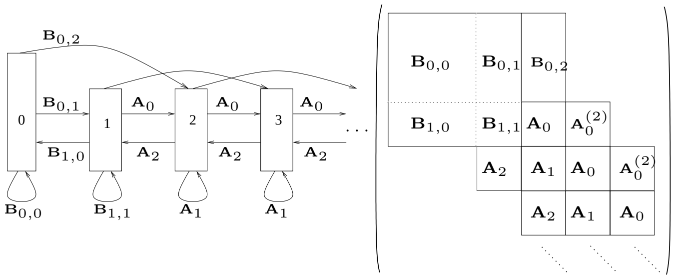
we require level-independent atomic propositions, CSL formulas are not level-independent in general. Considering CSL properties that contain path formulas, no two repeating levels can *a priori* be considered the same as we need to examine all possible paths in a level-wise fashion. For CSL formulas, we therefore generalized the idea of level independence: we only require the validity of an atomic proposition in a state to be *level-independent as of (repeating) level* k. Using this notion, we arrive at a situation in which "only" some finite boundary portion of the QBD may exhibit irregularities, which can be exploited in the further verification process. For model checking the CSL steady-state operator, we need to compute the steady-state probabilities in a QBD. As 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. That is, a steady-state formula is always level-independent as of level 1. For $\mathcal{S}_{\bowtie p}(\Phi)$ 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, $Sat(\mathcal{S}_{\bowtie p}(\Phi)) = \varnothing$. For model checking the CSL time-bounded until operator, i.e., $\mathcal{P}_{\bowtie p}(\Phi \ \mathcal{U}^I \Psi)$, we adapt the general approach as proposed for finite CTMCs [2]. This approach encompasses a transient analysis of an adapted CTMC; the adaptation depends on the subformulas Φ and Ψ in the until formula. For an allowed numerical error ϵ, uniformization requires a finite number n of steps to be taken into account in order to compute these transient probabilities. This fact, combined with the repetitive structure of the QBD, allows us to distill a finite portion of the QBD from which we can compute all relevant transient probabilities.

3 Resets

In standard QBDs, transitions can only occur between states of the same level or between states of neighboring levels. In QBDs with *resets* we additionally allow for transitions that lead from any repeating level to the boundary level. This is useful to model situations where all jobs in a system are lost due to some special event (like a server breakdown). The lower block-tridiagonal generator matrix as in Figure 2(b) is then composed out of nine matrices, where $\mathbf{C}_0$ denotes the jumps from the repeating levels to the boundary level. Again, considering level-independent atomic propositions, CSL formulas on QBDs with buffer resets are *not* level-independent in general. Even though it is now possible to reach the boundary level from every repeating level in one step, there still is the possibility to reach the boundary level via repeating levels. Due to this, the transient probabilities differ from level to level and, hence, we again have to apply the concept of level independence as of level k. The steady-state probabilities can be computed with known techniques [13, 9, 11]; the only difference to applying matrix-geometric methods on standard QBDs is that the boundary equations must now account for the newly introduced matrix $\mathbf{C}_0$, yielding $v_0 = v_0\mathbf{B}_{0,0} + \sum_{j=1}^{\infty} v_j \mathbf{C}_0$, with $\pi = (v_0, v_1, v_2, \cdots)$ being the steady-state vector. As $\mathbf{C}_0$ is a constant matrix, this equation can easily be solved due to the geometric structure of the solution vectors v_j. Computing the transient probabilities can again be done with uniformization. As we require the transition matrix $\mathbf{C}_0$ to be constant, we still have a finite number of different diagonal entries so that the uniformization rate can be determined. Model checking the until operator on QBDs with resets thus has the same complexity as on standard QBDs. In conclusion, model checking of QBDs with buffer resets can be done exactly as for standard QBDs [12].

4 Batches

For modeling the case that jobs enter or leave a system not only in single instances but also in *finite* batches, we need transitions between non-neighboring levels. We distinguish between (a) *batch arrivals*, (b) *batch departures*, and (c) the *general case*, which contains (a) and (b). The batches can have different size distributions. To describe a QBD with batch arrivals we need additional transition matrices $\mathbf{A}_0^{(i)}$, for every possible batch size i. $\mathbf{A}_0^{(1)}$ just equals $\mathbf{A}_0$. The additional transition matrices for batch departures are denoted as $\mathbf{A}_2^{(i)}$. Since we allow only for finite batches, the number of extra matrices is finite as well. The generator matrix of a QBD with batch arrivals, see Figure 3, has an upper block-tridiagonal from and can be seen as a special $M|G|1$ process, the generator matrix of a QBD with batch departures has lower block-tridiagonal form and can be seen as a special $G|M|1$ process. In the case of both batch arrivals and departures, the generator matrix takes a block-banded form. By regrouping as many states into one level as necessary to guarantee that transitions entering or leaving a level are restricted to neighboring levels only, we can transform the QBD with batches to a standard QBD. This procedure always works, as long as the maximum batch size is finite [8]. With regrouping, the level size is multi-



(a) batch arrivals of size 2 (b) generator matrix

Figure 3. QBD with batch arrivals

plied with the maximum batch size. If the QBD with batch arrivals or departures has level-independent atomic propositions than the regrouped QBD has level-independent atomic propositions as well, because only complete levels are regrouped. The regrouped QBD can then be model-checked with the standard procedure. We just have to make sure that the left and right level-diameter (that is, the number of state-transitions that are need to "step through" to the neighboring lower, resp. higher level) are adapted accordingly to keep uniformization efficient. As reducing QBDs with batch arrivals/departures to standard QBDs might lead to a considerable increase of the QBD order, it might be advantageous to model check these QBDs, without regrouping. This can be done with specialized matrix-geometric algorithms algorithms for $M|G|1$ and $G|M|1$ processes with batch arrivals and services [9], [13], or with the spectral expansion method [10]. Whether this is computationally more attractive has to be established. Uniformization can be done as for standard QBDs, we just have to consider that one uniformization step possibly crosses more than one level. That is, the amount of reachable levels to the right and to the left grows according to the maximum batch size for arrivals and departures. Thus, for model checking the until operator and the notion of level independence, the batch sizes have to be taken into account accordingly.

5 Periodic atomic propositions

Until now we always required QBDs with level-independent atomic propositions for model checking CSL. However, we are able to check QBDs with so-called *periodic atomic propositions* as well. An atomic proposition is called periodic with period p, iff its validity repeats every p levels. Atomic propositions with period $p = 1$ are called level-independent, with period $1 < p < \infty$ are called periodic, and with period $p = \infty$ are called level-dependent. For example an atomic proposition *odd* which is valid in states with odd level index, is periodic with period 2. QBDs with periodic atomic propositions with period p can be regrouped to QBDs with level-independent atomic propositions by combining p levels to one. In the case of several periodic atomic propositions with different periods the number of levels that need to be combined is given by the least common multiple of all periods. As far as we know, regrouping is the only possibility to model check QBDs with periodic atomic propositions.

6 Model checking CSRL

The continuous stochastic reward logic (CSRL) is a specification formalism for performability measures over CTMCs extended with a reward structure (Markov reward models) [6]. We can also extend QBDs with a *reward structure* $\rho : S \rightarrow \mathbf{R}_{\geq 0}$ that assigns a reward $\rho(s)$ to each state s. The syntax of a CSRL state formula is defined as follows:

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

where ϕ is a path formula constructed by $\phi ::= \mathcal{X}^{\leq t}_{\leq r}\Phi \mid \Phi\ \mathcal{U}^{\leq t}_{\leq r}\ \Phi$. The key difference to CSL is that the path-operators are equipped with two parameters. The additional parameter r represents a bound on the accumulated reward. We will consider three different types of rewards: *level-independent*, *level-dependent* and *periodic* rewards (see below). In doing so, we will concentrate on the four possible variations of the until operator, with and without time and with and without reward constraint, as presented in [3] and [6]. We are currently investigating how to exploit the regular QBD structure in the infinite size equation system that is necessary to check an until formula without time- nor reward constraint; this equation system follows directly from the finite-state case discussed in [6]. The until operator with only a time constraint $\Phi\ \mathcal{U}^{\leq t}\Psi$ is just the usual CSL until and can be checked as proposed in [12]. For checking the until operator with only a reward constraint, i.e., $\Phi\ \mathcal{U}_{\leq r}\Psi$, for finite CTMCs the *Duality Theorem* cab be used. This theorem states that the progress of time can be regarded as the earning of reward and vice versa [3]. Formulas with only a reward constraint can then be checked as formulas with just a time constraint on a transformed CTMCS. On QBDs, the Duality Theorem is applicable only in case of level-independent rewards. As the transition rates are rescaled by the reward rates, the QBD structure would be destroyed otherwise. For level-independent rewards the QBD structure does not change by this transformation and the QBD can be checked as stated in [6]. For the case with both time and reward constraint, $\Phi\ \mathcal{U}^{\leq t}_{\leq r}\Psi$, we consider how to check the formula for only one starting state. In doing so, we can use well-known algorithms as only a finite number of steps is considered on the uniformized or discretized variant of the MRM. In order to compute the satisfaction set we have to distinguish between the three different reward types, as distinguished above.

In the case of *level-dependent rewards* we require an increasing function of the level-index, the satisfaction set is then always finite, as from a certain level onwards the reward constraint cannot be fulfilled anymore. This is the case for level j, when the states of the leftmost reachable level (reachable in the sense of the maximum number of steps taken in uniformization) have a reward r_{low}, such that $r_{low} \cdot t > r$. The number of states with level index smaller than j is finite, which allows for a direct verification.

In the case of *level-independent rewards* we require the same reward for corresponding states in different levels, the satisfaction set is potentially of infinite size. Fortunately we will eventually find a level from which onwards the validity of the formula will be the same in all corresponding states of the repeating levels. This is just a straightforward extension of the ideas presented in [12], that can be used because of the special reward structure.

In the case of *periodic rewards with reward period* p the QBD with periodic reward structure can be transformed to a QBD with level-independent rewards by regrouping of levels, as discussed previously.

7 Extensions of CSRL

There are several extensions of CSRL [3]; here we will discuss how to apply the *expected reward* ($\mathcal{E}_{\leq r}(\Phi)$) and the *instantaneous reward* ($\mathcal{E}^t_{\leq r}(\Phi)$) operator on QBDs. The semantics of the expected reward operator is:

$$s \models \mathcal{E}_{\leq r}(\Phi) \quad \text{iff} \sum_{s' \in Sat(\Phi)} \pi(s, s')\rho(s') \leq r,$$

where $\pi(s, s')$ is the steady-state probability to be in state s' when having started in state s. In case of a finite satisfaction set $Sat(\Phi)$, we have a possibly large but finite summation, that can be dealt with. For level-independent rewards and an infinite satisfaction set $Sat(\Phi)$, we do not know how to check the expected reward operator. The iterative approach that has been used in [12] to check the steady-state operator cannot be used as the probability mass is modified by the reward. In the special case, where the reward equals the level-index we can derive a closed-form solution for the expected reward (by applying a geometric argument to the infinite sum), hence, model checking seems feasible. As the steady-state probabilities in a QBD are independent of the starting state, we immediately know the satisfaction set after checking the reward operator for one starting state.

The semantics of the instantaneous reward operator is:

$$s \models \mathcal{E}^t_{\leq r}(\Phi) \quad \text{iff} \sum_{s' \in Sat(\Phi)} \pi(s, s', t)\rho(s') \leq r,$$

where $\pi(s, s', t)$ is the transient probability to reach state s' from state s in time t. To calculate the transient probabilities in a QBD we always consider only a finite number of steps. That is, the instantaneous reward operator can always be checked for a single starting state s, regardless of the reward structure. To calculate the satisfaction set we distinguish between level-independent and level-dependent rewards. We will eventually find a level from which onwards the transient probabilities do not change anymore. With level-independent rewards the validity of the instantaneous reward operator does not change anymore from this level onwards. We do not know how to check the instantaneous reward operator with level-dependent rewards and an infinite satisfaction set in all cases as the reward modifies the transient probabilities.

8 Conclusions

We have shown that for QBDs with resets, batches, and periodic atomic propositions, the algorithms for model checking CSL as presented in [12] still apply, after an appropriate modification of the QBD. Furthermore, we discussed how we can extend the model checking approach for QBDs toward CSRL (and its extensions), so that we can model check for combined performance and dependability, that is, performability measures. We discussed a number of cases for which this is possible, and we conjecture two cases for which we think CSRL model checking will not be possible for QBDs.

Acknowledgments The authors would like to thank Lucia Cloth for fruitful discussions on this topic. 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).

References

[1] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In *CAV'00*, LNCS 1855:358–372. Springer, 2000.

[2] C. Baier, B. 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.

[3] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. On the logical characterisation of performability properties. In *Proc. ICALP '00*, pages 780–792. Springer, 2000.

[4] C. Baier, J.-P. Katoen, and H. Hermanns. Approximate symbolic model checking of continuous-time Markov chains. In *CONCUR'99*, LNCS 1664:146–161. Springer, 1999.

[5] L. Cloth, J.-P. Katoen, M. Khattri, and R. Pulungan. Model checking Markov reward models with impulse rewards. In *Proc. DSN'05*,IEEE CS press, 2005.

[6] B. Haverkort, L. Cloth, H. Hermanns, J. Katoen, and C. Baier. Model checking performability properties. In *Proc. DSN'02*. IEEE CS press, 2002.

[7] B. Haverkort, H. Hermanns, and J.-P. Katoen. On the use of model checking techniques for dependability evaluation. In *Proc. SRDS'00*, pages 228–237. IEEE CS Press, 2000.

[8] B. Haverkort and A. Ost. Steady-state analysis of infinite stochastic petri nets: A comparing between the spectral expansion and the matrix-geometric method. In *Proc. PNPM'97*. IEEE CS Press, 1997.

[9] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM, 1999.

[10] I. Mitrani and R. Chakka. Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method. *Performance Evaluation*, 23(3):241–260, 1995.

[11] M. Neuts. *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Johns Hopkins University Press, 1981.

[12] A. Remke, B. Haverkort, and L. Cloth. Model checking infinite-state Markov chains. In *TACAS'05* LNCS 3440:237–252. Springer, 2005.

[13] A. Riska. *Aggregate matrix-analytic techniques and their applications*. PhD thesis, The Faculty of the Dept. of Computer Science, The College of William and Mary in Virginia, 2002.