

# CSL Model Checking Algorithms for Infinite-State Structured Markov Chains<sup>\*</sup>

Anne Remke and Boudewijn R. Haverkort

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

**Abstract.** Jackson queueing networks (JQNs) are a very general class of queueing networks that find their application in a variety of settings. The state space of the continuous-time Markov chain (CTMC) that underlies such a JQN, is highly structured, however, of infinite size in as many dimensions as there are queues. We present CSL model checking algorithms for labeled JQNs. To do so, we rely on well-known product-form results for the steady-state probabilities in (stable) JQNs. The transient probabilities are computed using an uniformization-based approach. We develop a new notion of property independence that allows us to define model checking algorithms for labeled JQNs.

## 1 Introduction

Queueing networks have been used for about half a century now, for modeling and analyzing a wide variety of phenomena in computer, communication and logistics systems. Seminal work on queueing networks was done by Jackson in the 1950s [11,12]; in which he developed an important theorem that characterizes the steady-state (long-run) probabilities to reside in certain states in a restricted class of queueing networks (see the next section).

However, there are many phenomena in the above classes of systems that cannot be studied well using these long-run probabilities. In communication system models, the following situations reflect such cases: (i) what is the probability that starting from an initial empty system, within  $t$  time-units, at least  $k_i$  packets are buffered at queue  $i$ ? (ii) what is the probability that starting from an overload situation, e.g., characterized by at least  $L$  packets at each queue, within  $t$  time-units, a low load situation, e.g., characterized by at most  $l \ll L$  packets at each queue is reached again?

Clearly, the above sketched scenarios require more than just long-run probabilities. Since logics like CSL have been shown to be extremely helpful in specifying similar properties for finite CTMCs [2,3], we also pursue a CSL model checking procedure for the CTMCs that underlie so-called Jackson queueing networks

---

<sup>\*</sup> 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). The authors thank Lucia Cloth for fruitful discussions on the topic.

(JQNs). As we know from [3], for CSL model checking of CTMCs, we need to be able to compute both steady-state and transient state probabilities for all states, and for all possible starting states. The key issue lies in the fact that the CTMC underlying a JQN is infinite in as many dimensions as there are queues in the JQN. For the steady-state probabilities we can rely on the seminal work of Jackson, however, for the transient state probabilities, no results are readily available. Inspired by our recent work on CSL model checking for quasi-birth-death processes (QBDs), which form a different class of infinite-state CTMCs [15], we use an uniformization-based approach to compute just those transient state probabilities in JQNs that are needed to verify the validity of CSL properties. The highly structured state space allows us to conclude the validity of CSL properties for groups of states on the basis of the validity for a so-called representative state in such a group. This reduces the infinite number of state probabilities to be computed to a finite number.

We are not aware of any other work that addresses the model checking problem solved in this paper. However, there is *related work* on model checking infinite state systems, such as, e.g., Boucherie product processes [4], probabilistic lossy channel systems [18], regular model checking [1], recursive Markov chains [8] and probabilistic pushdown automata [5], as well as on transient analysis of queueing networks [14,10] and on transient analysis of infinite-state systems with uniformization [19,6]. None of this work, however, addresses the model checking questions that we address.

The rest of this paper is organized as follows. We introduce JQNs in Section 2 and discuss the form of the underlying state space and transition relation in detail in Section 3. We briefly rehearse the logic CSL in Section 4, before we discuss the model checking algorithms for all the CSL operators in Section 5. Note that we restrict ourselves to the time interval  $[0, t]$  for the time-bounded until operator. Finally, Section 6 presents some conclusions. A running example to illustrate the key concepts and procedures is provided throughout the paper.

## 2 Jackson Queueing Networks

*Jackson queueing networks* (JQNs) consist of a number of interconnected queueing stations, numbered  $1, \dots, n$ . At each individual queueing station  $i$ , jobs arrive with a negative exponential inter-arrival time distribution with rate  $\lambda_i$ , and the job service requirements are also negative exponentially distributed, however, with rate  $\mu_i$ . There is a single server at each queueing station<sup>1</sup> Jobs arriving at a queue are served in *first come first served* (FCFS) order. Jobs arriving when the server is busy, are queued in an unbounded buffer. Each queue in a JQN behaves, in essence, as a simple so-called  $M|M|1$  queue. We assume a never empty source from which customers originate and arrive at the JQN, and into which they disappear after having received their service. This environment is indexed 0 and the overall arrival process from the environment is a Poisson process with rate  $\lambda$ .

---

<sup>1</sup> This can be alleviated easily to  $m$ -servers.

A finite routing matrix  $\mathbf{R} \in \mathbb{R}^{(n+1) \times (n+1)}$  contains the routing probabilities from queue  $i$  to queue  $j$  with  $i, j \in \{0, 1, \dots, n\}$ :  $r_{i,j} \in [0, 1]$ . Note that  $\sum_{j=0}^n r_{i,j} = 1$  for all  $i$ . In case  $r_{i,j} = 0$  there is no routing from queue  $i$  to queue  $j$  and in case  $r_{i,j} = 1$ ,  $j$  is the only output for queue  $i$ . The routing probability  $r_{i,0}$  gives the probability that a job actually leaves the queueing network after completion at queue  $i$  and the routing probability  $r_{0,j}$  gives the probability that an arriving customer is routed to queue  $j$ . Note that  $r_{0,0} = 0$  by definition and that we do allow for self loops (e.g.,  $r_{i,i} > 0$  is allowed). A state in a JQN can be defined as  $\bar{s} = (s_1, s_2, \dots, s_n)$ , where  $s_i \geq 0$  represents the number of customers in queue  $i$ . A more precise discussion of the state space  $\mathbf{S}$  is postponed to Section 3. For model checking purposes, we also need a state labeling. This leads us to the following definition.

**Definition 1 (Jackson queueing networks)**

A labeled Jackson queueing network **LJQN**  $\mathcal{J}$  of order  $n$  (with  $n \in \mathbb{N}^+$ ) is a tuple  $(\lambda, \underline{\mu}, \mathbf{R}, L)$  with arrival rate  $\lambda$ , a vector of size  $n$  of service rates  $\underline{\mu}$ , a routing matrix  $\mathbf{R} \in \mathbb{R}^{(n+1) \times (n+1)}$  and a labeling function  $L$  that assigns a set of valid atomic propositions from a fixed and finite set  $AP$  of atomic propositions to each state  $\bar{s} = (s_1, s_2, \dots, s_n)$ . □

**Restriction 1**

In the following we will restrict ourselves to atomic propositions of the form  $\bigwedge_{i=1}^n (s_i \leq m_i)$  for  $m_i \in \mathbb{N}$ . This restricts the formulas we are able to check.

**Example 1**

In Figure 1(a) we present a LJQN with two queues that will serve as running example. The external arrival rate is  $\lambda$ , the vector of service rates is given as

$$\underline{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \text{ and the routing matrix is } \mathbf{R} = \begin{pmatrix} 0 & r_{0,1} & r_{0,2} \\ r_{1,0} & 0 & r_{1,2} \\ r_{2,0} & r_{2,1} & 0 \end{pmatrix}. \text{ The labeling } L \text{ will}$$

be introduced later.

**Definition 2 (Traffic equations)**

The overall flow of jobs through each queue  $j$  is given as:  $\lambda_j = \lambda r_{0,j} + \sum_{i=1}^n \lambda_i r_{i,j}$ , for  $j = 1, \dots, n$ . These equations are called (*first-order*) *traffic equations*. □

**Definition 3 (Utilization)**

The utilization per queue  $i$  is defined as  $\rho_i = \frac{\lambda_i}{\mu_i}$ . □

**Restriction 2**

In case all  $\rho_i = \frac{\lambda_i}{\mu_i} < 1$ , the QN is said to be stable. That is the number of arriving jobs per unit time is smaller than the amount of jobs that each queue can handle per unit time. This guarantees that the queue will not build up infinitely. In the following we restrict ourselves to stable LJQNs, to be able to compute steady-state probabilities.

The long-run probability that  $s$  customers are presently in a single  $M|M|1$  queue, that is, the so-called *steady-state probability*, for a single stable  $M|M|1$  queue (with  $\rho = \frac{\lambda}{\mu} < 1$ ) is:  $\Pr\{S = s\} = (1 - \rho)\rho^s$ , where  $S$  is the random variable indicating the number of customers in the queue [13]. In [11,12], Jackson proved the following theorem:

**Theorem 1 (Jackson)**

The overall steady-state probability distribution under restriction 2 (stability) is the product of the per-queue steady-state probability distributions, where the queues can be regarded as if operating independently from each other:

$$\Pr\{\bar{S} = \bar{s}\} = \prod_{i=1}^n (1 - \rho_i)\rho_i^{s_i}, \quad \bar{s} = (s_1, \dots, s_n) \in S. \quad (1)$$

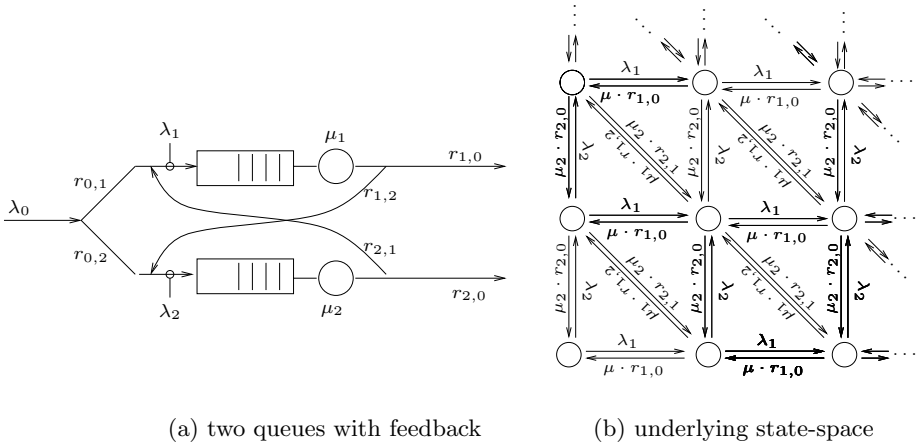


Fig. 1. JQN with two queues

### 3 State Space, Transitions, Independence and Paths

This section addresses in detail the underlying state space of a LJQN. In Section 3.1 the underlying infinite state structured Markov chain is described. Section 3.2 explains how the infinite state space can be structured using atomic propositions and a new notion of independence is introduced. In Section 3.3 we define paths and transient as well as steady-state probabilities on LJQNs.

#### 3.1 Infinite State Structured Markov Chain

The underlying state space of a LJQN  $\mathcal{J}$  of order  $n$  is a highly structured labeled infinite-state continuous-time Markov chain with state space  $S = \mathbb{N}^n$  that is infinite in  $n$  dimensions. Every state  $\bar{s} \in S$  can be represented as an  $n$ -tuple  $\bar{s} = (s_1, s_2, \dots, s_n)$ , with  $s_i \geq 0$ . The labeling function  $L : S \rightarrow 2^{AP}$  on the

state space then assigns from the set  $AP$  of atomic propositions the set of valid atomic propositions to each state. The state  $\bar{s} = (0, \dots, 0)$  is called origin. As the number of customers in queue  $i$  is restricted to  $s_i \geq 0$ , for all  $i \in \{1, \dots, n\}$ , the underlying state-space is limited towards the origin in every dimension. An  $n$  dimensional state space  $S$  is bounded by  $n$  so-called *boundary hyperplanes* of dimension  $n - 1$ . Note that these boundary hyperplanes consist of an infinite number of states for  $n \geq 2$ . State changes may occur due to an arrival at queue  $i$  from the environment or a departure to the environment from queue  $i$ , or by routing of a customer from queue  $i$  to queue  $j$  for  $1 \leq i, j \leq n$ . By adding a state change vector to the source state, the destination state is defined: an arrival at queue  $i$  is denoted by the vector  $\bar{a}_i$ , a departure from queue  $i$  is denoted by the vector  $\bar{d}_i$ , and a job routing from queue  $i$  to queue  $j$  is denoted as vector  $\bar{f}_{i,j}$ , with:

$$\bar{a}_i = \begin{cases} a_j = 1 & i = j, \\ a_j = 0 & j \neq i. \end{cases} \quad \bar{d}_i = \begin{cases} d_j = -1 & j = i, \\ d_j = 0 & j \neq i. \end{cases} \quad \bar{f}_{i,j} = \begin{cases} f_k = -1 & k = i, \\ f_k = 1 & k = j, \\ f_k = 0 & \text{otherwise.} \end{cases}$$

Note that an arrival is always possible, the new state is then defined as  $\bar{s}' = \bar{s} + \bar{a}_i$ . A departure or a job routing from queue  $i$  is only possible, when there is at least one customer in queue  $i$ ; in this case the new state  $\bar{s}'$  is computed as:  $\bar{s}' = \bar{s} + \bar{d}_i$  or  $\bar{s}' = \bar{s} + \bar{f}_{i,j}$ , respectively.

**Definition 4 (Generator function)**

The rate for a state change from a state  $\bar{s}$  to another state  $\bar{s}'$  within the infinite state space  $S$  is given by the generator function  $\mathbf{G}(\bar{s}, \bar{s}') : S \times S \rightarrow \mathbb{R}^+$ , for  $\bar{s} \neq \bar{s}'$ , as follows:

cause	restriction	state change	$\mathbf{G}(\bar{s}, \bar{s}')$
arrival	none	$\bar{s}' = \bar{s} + \bar{a}_i$	$\lambda \cdot r_{0,i}$
departure	$s_i > 0$	$\bar{s}' = \bar{s} + \bar{d}_i$	$\mu_i \cdot r_{i,0}$
routing	$s_i > 0$	$\bar{s}' = \bar{s} + \bar{f}_{i,j}$	$\mu_i \cdot r_{i,j}$

and  $G(\bar{s}, \bar{s}') = 0$  for  $\bar{s} \neq \bar{s}'$  in all other cases.  $\mathbf{G}(\bar{s}, \bar{s})$  is defined as the negative sum over all possible outgoing rates from  $\bar{s}$ , that is

$$\mathbf{G}(\bar{s}, \bar{s}) = - \left( \lambda + \sum_{i=1}^n \mu_i \cdot \mathbf{1}_{s_i > 0} \right),$$

where the indicator function  $\mathbf{1}_{s_i > 0}$  returns 1 if  $s_i > 0$ , and 0 otherwise. □

**Example 2**

Figure 1(b) shows the underlying state space of the LJQN from Example 1 that is infinite in two dimensions. Arrivals occur in both dimensions with rate  $\lambda \cdot r_{0,1}$  and  $\lambda \cdot r_{0,2}$ , departures happen from both dimensions with rates  $\mu_1 \cdot r_{1,0}$  and  $\mu_2 \cdot r_{2,0}$ , respectively, and jobs are routed from queue 1 to queue 2 with rate  $\mu_1 \cdot r_{1,2}$  and from queue 2 to queue 1 with rate  $\mu_2 \cdot r_{2,1}$ .

### 3.2 Independence of Atomic Propositions

Recall that we restrict ourselves to atomic propositions of the form  $\bigwedge_{i=1}^n (s_i \leq m_i)$  for  $m_i \in \mathbb{N}$ . Due to this restriction, the validity of an atomic proposition does not change anymore for  $s_i \geq m_i$  onwards for dimension  $i$ . Hence, we can define the so-called *independence vector*  $\overline{m} = (m_1, \dots, m_n)$  and call the atomic proposition *independent as of*  $\overline{m}$ . For the set of states  $\{\overline{s} \in \mathbb{S} \mid \forall i (s_i \geq m_i)\}$  the validity of the atomic proposition remains the same.

The state space can be partitioned into a finite set of boundary states  $S_b$  and a finite number of infinite representative sets of states (denoted  $S_{\overline{r}}$ ) such that the validity of an atomic proposition  $ap \in AP$  does not change any more in this set. We choose a *representative state*  $\overline{r}$  for each of these infinite sets  $S_{\overline{r}}$  such that for all  $\overline{s} \in S_{\overline{r}}$  the labeling does not change:  $L(\overline{r}) = L(\overline{s})$ , for all  $\overline{s} \in S_{\overline{r}}$ . In general, in an  $n$ -dimensional LJQN, there are  $n$  types of representative sets that account for 1 up to  $n$  infinite dimensions. A representative set is called infinite in dimension  $i$  if and only if  $r_i = m_i$ , and restricted in dimension  $i$  otherwise. In case a representative state  $\overline{r}$  is infinite in  $k$  dimensions it represents a  $k$  dimensional set  $S_{\overline{r}}$ , such that

$$\overline{s} \in S_{\overline{r}} \Leftrightarrow \begin{cases} s_i \geq r_i & \text{iff } r_i = m_i, \\ s_i = r_i & \text{otherwise.} \end{cases}$$

Hence, a state  $\overline{s}$  belongs to  $S_{\overline{r}}$  when it takes the same value as  $\overline{r}$  in the restricted dimensions and any value  $\geq r_i$  in the infinite dimensions. For atomic propositions, the origin  $\hat{s}$  represents the finite set of boundary states that is defined as  $S_b = \{\overline{s} \in \mathbb{S} \mid s_i < m_i, \forall i \in \{1, \dots, n\}\}$ . The finite union of all representative states is called *representative front* and defined as  $R(\overline{m}) = \{\overline{r} \in \mathbb{S} \mid \exists i : (r_i = m_i) \wedge (\forall j \neq i : (r_j \leq m_j))\}$ . Note that  $\mathbb{S} = S_b \cup \bigcup_{\overline{r} \in R(\overline{m})} S_{\overline{r}}$ . For atomic propositions the representative front can be made smaller, however, for model checking CSL properties in general we need the full representative front as defined above.

#### Example 3

Suppose we define the atomic proposition  $ap_1 = (s_1 \geq 2) \wedge (s_2 \geq 3)$  for the LJQN from Example 1; the white states in Figure 2(a) then depict those states where  $ap_1$  is valid. The representative front for  $ap_1$  is formed by the states in the grey polygon:  $(0, 3)$  accounts for the states  $(0, j)$ , with  $j \geq 3$ , and  $(1, 3)$  represents the states  $(1, j)$  with  $j \geq 3$ . For  $i \geq 2$ ,  $(2, 0)$  represents  $(i, 0)$ ,  $(2, 1)$  represents  $(i, 1)$  and  $(2, 2)$  represents  $(i, 2)$ , respectively. These five representative states all account for a one dimensional set of states. The representative state  $(2, 3)$  accounts for the two-dimensional set of states  $S_{(2,3)} = \{\overline{s} \in \mathbb{S} \mid s_1 > 2 \wedge s_2 > 3\}$ . The black states belong to the boundary set  $S_b$  with representative  $(0, 0)$ .

#### Example 4

Figure 2(b) shows the representative front for the atomic proposition  $ap_2 = (s_1 < 3) \wedge (s_2 < 3) \wedge (s_3 < 3)$  in a LJQN of dimension three. The atomic proposition is valid in the black states only (of the 27 only 9 are visible). All the remaining depicted states are representative states. We have different types of

representative states: the white ones represent a set of states that is infinite in one dimension, the grey ones represent a set that is infinite in two dimensions and the light grey one (3, 3, 3) represents a set that is infinite in three dimensions.

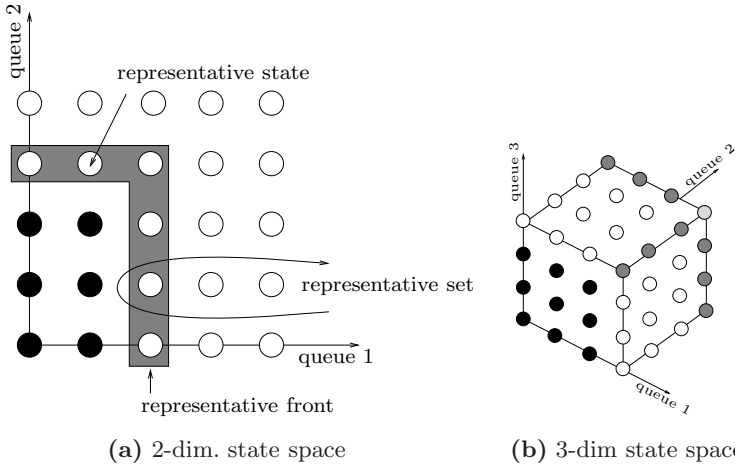


Fig. 2. Representative front of states for independent atomic propositions

### 3.3 Paths and Probabilities

In what follows, we first present the notion of path, before we define steady-state and transient state probabilities.

#### Definition 5 (Paths)

An *infinite path*  $\sigma$  is a sequence  $\bar{s}_0 \xrightarrow{t_0} \bar{s}_1 \xrightarrow{t_1} \bar{s}_2 \xrightarrow{t_2} \dots$  with, for  $i \in \mathbb{N}$ ,  $\bar{s}_i \in \mathbb{S}$  and  $t_i \in \mathbb{R}_{>0}$  such that  $\mathbf{G}(\bar{s}_i, \bar{s}_{i+1}) > 0$  for all  $i$ . A *finite path*  $\sigma$  of length  $l + 1$  is a sequence  $\bar{s}_0 \xrightarrow{t_0} \bar{s}_1 \xrightarrow{t_1} \dots \bar{s}_{l-1} \xrightarrow{t_{l-1}} \bar{s}_l$  such that  $\bar{s}_l$  is absorbing<sup>2</sup>, and  $\mathbf{G}(\bar{s}_i, \bar{s}_{i+1}) > 0$  for all  $i < l$ . For an infinite path  $\sigma$ ,  $\sigma[i] = \bar{s}_i$  denotes for  $i \in \mathbb{N}$  the  $(i + 1)$ st state of path  $\sigma$ . The time spent in state  $\bar{s}_i$  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  $\sigma$  with length  $l + 1$ ,  $\sigma[i]$  and  $\delta(\sigma, i)$  are defined in the way described above for  $i < l$  only and  $\delta(\sigma, l) = \sigma[l] = \infty$  and  $\delta@t = \bar{s}_l$  for  $t > \sum_{j=0}^{l-1} t_j$ .  $\text{Path}^{\mathcal{J}}(\bar{s})$  is the set of all finite and infinite paths in the LJQN  $\mathcal{J}$  that start in state  $\bar{s}$  and  $\text{Path}^{\mathcal{J}}$  includes all (finite and infinite) paths of the LJQN  $\mathcal{J}$ .  $\square$

As for finite CTMCs, a probability measure on paths can now be defined depending on the starting state [3]. Starting from there, two different types of state probabilities can be distinguished.

<sup>2</sup> A state  $\bar{s}$  is called absorbing if for all  $\bar{s}'$  the rate  $\mathbf{G}(\bar{s}, \bar{s}') = 0$ .

**The transient state probability** is a time-dependent measure that considers the LJQN at a given time instant  $t$ . The probability to be in state  $\bar{s}'$  at time instant  $t$ , given initial state  $\bar{s}$ , is denoted as  $\mathbf{V}^{\mathcal{J}}(\bar{s}, \bar{s}', t) = \Pr\{\sigma \in \text{Path}^{\mathcal{J}}(\bar{s}) \mid \sigma @ 0 = \bar{s} \wedge \sigma @ t = \bar{s}'\}$ . The transient probabilities are characterized by a linear system of differential equations of infinite size. Let  $\mathbf{V}(t)$  be the matrix of transient state probabilities at time  $t$  for all possible starting states  $\bar{s}$  and for all possible goal states  $\bar{s}'$  (we omit the superscript  $\mathcal{J}$  for brevity here), then we have  $\mathbf{V}'(t) = \mathbf{V}(t) \cdot \mathbf{G}$ , given  $\mathbf{V}(0) = \mathbf{I}$ . An efficient method to compute the transient probabilities will be discussed in Section 5.4.

**The steady-state probabilities** to be in state  $\bar{s}'$ , given initial state  $\bar{s}$ , are defined as  $\pi^{\mathcal{J}}(\bar{s}, \bar{s}') = \lim_{t \rightarrow \infty} \mathbf{V}^{\mathcal{J}}(\bar{s}, \bar{s}', t)$ , and indicate the probabilities to be in some state  $\bar{s}'$  “in the long run”. If steady-state is reached, the above mentioned derivatives  $\mathbf{V}'(t)$  will approach zero. As we require stable queues the underlying state space of the LJQN is ergodic and, the initial state does not influence the steady-state probabilities (we therefore write  $\pi(\bar{s}')$  instead of  $\pi(\bar{s}, \bar{s}')$  for brevity). In the context of LJQNs the steady-state probabilities per state can be computed using the product-form of Theorem 1.

## 4 The logic CSL

We use the logic CSL [3] to express properties for LJQNs. The syntax and semantics are the same as for finite CTMCs, with the only difference that we now interpret the formulas over states and paths in LJQNs. Let  $p \in [0, 1]$  be a real number,  $\bowtie \in \{\leq, <, >, \geq\}$  a comparison operator,  $t_1, t_2 \in \mathbb{R}^+$  real numbers and  $AP$  a set of atomic propositions with  $ap \in AP$ . CSL state formulas  $\Phi$  are defined by

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

where  $\phi$  is a path formula constructed by

$$\phi ::= \mathcal{X}^{[t_1, t_2]}\Phi \mid \Phi \mathcal{U}^{[t_1, t_2]}\Psi.$$

For a CSL state formula  $\Phi$  and a LJQN  $\mathcal{J}$ , the satisfaction set  $Sat(\Phi)$  contains all states of  $\mathcal{J}$  that fulfill  $\Phi$  and is computed with a recursive descent procedure over the parse tree of  $\Phi$ , as for CTL [7]. Satisfaction is stated in terms of a satisfaction relation  $\models$ . The relation  $\models$  for states and CSL state formulas is defined as:

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

where  $\pi^{\mathcal{J}}(\bar{s}, Sat(\Phi)) = \sum_{\bar{s}' \in Sat(\Phi)} \pi^{\mathcal{J}}(\bar{s}, \bar{s}')$ , and  $Prob^{\mathcal{J}}(\bar{s}, \phi)$  describes the probability measure of all paths  $\sigma \in \text{Path}(\bar{s})$  that satisfy  $\phi$  when starting in state  $\bar{s}$ , that is,  $Prob^{\mathcal{J}}(\bar{s}, \phi) = \Pr\{\sigma \in \text{Path}^{\mathcal{J}}(\bar{s}) \mid \sigma \models \phi\}$ . The steady-state operator  $\mathcal{S}_{\bowtie p}(\Phi)$  denotes that the steady-state probability for  $\Phi$ -states meets the bound  $p$ .



$\mathcal{P}_{\triangleright p}(\phi)$  asserts that the probability measure of the paths satisfying  $\phi$  meets the bound  $p$ . The next operator  $\mathcal{X}^{[t_1, t_2]}\Phi$  states that a transition to a  $\Phi$ -state is made between  $t_1$  and  $t_2$ . The until operator  $\Phi \mathcal{U}^{[t_1, t_2]}\Psi$  asserts that  $\Psi$  is satisfied at some time instant in between  $[t_1, t_2]$  and that at all preceding time instants  $\Phi$  holds. The relation  $\models$  for paths and CSL path formulas is defined as:

$$\begin{aligned} \sigma &\models \mathcal{X}^{[t_1, t_2]}\Phi && \text{iff } \sigma[1] \text{ is defined and } \sigma[1] \models \Phi \text{ and } t_1 \leq \delta(\sigma, 0) \leq t_2, \\ \sigma &\models \Phi \mathcal{U}^{[t_1, t_2]}\Psi && \text{iff } \exists t(t_1 \leq t \leq t_2) (\sigma @ t \models \Psi \wedge (\forall t' \in [0, t)(\sigma @ t' \models \Phi)). \end{aligned}$$

From [15] we know that CSL formulas are not level independent on QBDs in general, even if the atomic propositions are level independent. However, their validity does not change arbitrarily between levels. In the following we will show that for CSL formulas on LJQNs we can also find a state from which the validity of the CSL formula does not change anymore.

**Definition 6 (Independence of CSL formulas)**

Let  $\mathcal{J}$  be a LJQN of order  $n$ . A CSL state formula  $\Phi$  is *independent* as of  $\overline{m}$  if and only if there exists a finite *representative front*  $R(\overline{m}) = \{\overline{r} \in S \mid \exists i(r_i = m_i) \wedge (\forall j \neq i(r_j \leq m_j))\}$  such that for all  $\overline{r} \in R(\overline{m})$  and for all  $\overline{s} \in S_{\overline{r}}$  it holds that  $\overline{r} \models \Phi \Leftrightarrow \overline{s} \models \Phi$ . □

The following propositions states, under the assumption of independent atomic propositions as of  $\overline{m}$ , that such a finite representative front  $R(\overline{m}')$  exists for any CSL state formula. We will justify this proposition inductively over the structure of the logic in Section 5.

**Proposition 1**

Let  $\mathcal{J}$  be a LJQN of order  $n$  with independent atomic propositions as of  $\overline{m}$  and let  $\Phi$  be a CSL state formula other than  $\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^{[t_1, t_2]}\Psi)$ . Then there exists a finite *representative front*  $R(\overline{m}')$  such that the validity of  $\Phi$  does not change within each subset  $S_{\overline{r}}$  for  $\overline{r} \in R(\overline{m}')$  in  $\mathcal{J}$ . For the until operator  $\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^I\Psi)$  we assume that for no state  $\overline{s}$  the probability measure is exactly equal to  $p$ . Under this assumption, there exists a vector  $\overline{m}'$ , such that  $\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^I\Psi)$  is independent as of  $\overline{m}'$  in  $\mathcal{J}$ . □

For a CSL formula that is independent as of  $\overline{m}$ , the satisfaction set can be considered as the union of the *boundary satisfaction set*  $Sat^{S_b}(\Phi) = S_b \cap Sat(\Phi)$  and the *representative satisfaction front*  $Sat^{R(\overline{m})}(\Phi) = R(\overline{m}) \cap Sat(\Phi)$ . The representative state  $\overline{r} \in R(\overline{m})$  then represent the remaining infinite state space.

## 5 Model Checking Algorithms

In this section we present model checking algorithms for CSL. In Section 5.1 we explain how independence changes when applying logical operators. We explain how to model check the steady-state operator in Section 5.2, the next operator in 5.3 and the until operator in Section 5.4, including a discussion on how to compute transient probabilities in LJQNs.

### 5.1 Logical Operators

The model checking procedure for logical operators is the same as for finite CTMCs. The only thing we need to take care of is how independence changes. Negating a CSL formula does not change its independence. For a CSL formula  $\Phi$  that is independent as of  $\overline{m}$  the negation  $\neg\Phi$  is also independent as of  $\overline{m}$ . However, combining a CSL formula  $\Phi$  that is independent as of  $\overline{m}^\Phi$  with a CSL formula  $\Psi$  that is independent as of  $\overline{m}^\Psi$  with conjunction, changes independence depending on the structure of  $\Phi$  and  $\Psi$ . In any case, we can state that  $\Phi \wedge \Psi$  is independent as of  $\overline{m} = \max\{\overline{m}^\Phi, \overline{m}^\Psi\}$ , where we choose the maximum of  $m_i^\Phi$  and  $m_i^\Psi$  in every dimension  $i$ . Note that this new independence vector  $\overline{m}$  might be too pessimistic, depending on the structure of  $\Phi$  and  $\Psi$ .

### 5.2 Steady-State Operator

Theorem 1 states the steady-state probabilities in a JQN. A state  $s$  satisfies  $\mathcal{S}_{\bowtie p}(\Phi)$  if the sum of the steady-state probabilities of all  $\Phi$ -states reachable from  $s$  meets the bound  $p$ . Since a JQN is by definition ergodic, 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 independent as of  $\overline{m} = (0, \dots, 0)$ . We sum the steady state probabilities of all states that satisfy  $\Phi$  by summing over all states  $\overline{s} \in S_{\overline{r}}$  for  $\overline{r} \models \Phi$  and over all states from  $S_b$  that satisfy  $\Phi$ :

$$\overline{s} \models \mathcal{S}_{\bowtie p}(\Phi) \Leftrightarrow \sum_{\substack{\overline{s} \in S_b, \\ \overline{s} \in Sat(\Phi)}} p(\overline{s}) + \sum_{\substack{\overline{r} \in R(\overline{m}), \\ \overline{r} \in Sat(\Phi)}} \sum_{\overline{s} \in S_{\overline{r}}} p(\overline{s}) \bowtie p \tag{2}$$

We obtain  $Sat(\mathcal{S}_{\bowtie p}(\Phi)) = S$ , if the accumulated steady-state probability meets the bound  $p$  otherwise  $Sat(\mathcal{S}_{\bowtie p}(\Phi)) = \emptyset$ . In case the representative state  $\overline{r} \in Sat(\Phi)$ , all states  $\overline{s} \in S_{\overline{r}}$  are in  $Sat(\Phi)$ . The accumulated steady-state probability for all states  $\overline{s} \in S_{\overline{r}}$  is given by the following expression:

$$\sum_{\overline{s} \in S_{\overline{r}}} p(\overline{s}) = \prod_{i=1}^n \Omega(i), \text{ with } \Omega(i) = \begin{cases} (1 - \rho_i)\rho_i^{r_i} & \text{for } r_i \neq m_i, \\ \rho_i^{m_i} & \text{for } r_i = m_i. \end{cases} \tag{3}$$

In this expression we distinguish between the finite and the infinite dimensions of a representative state  $\overline{r}$ . In a finite dimension  $i$  we multiply with  $(1 - \rho_i)\rho_i^{r_i}$  and in an infinite dimension  $i$  we multiply with  $\rho_i^{m_i}$ .

#### Proof 1 (Accumulated steady-state probability)

Applying Jackson’s Theorem, the accumulated steady-state probability is given by

$$\sum_{\overline{s} \in S_{\overline{r}}} p(\overline{s}) = \sum_{\overline{s} \in S_{\overline{r}}} \left( \prod_{i=1}^n (1 - \rho_i)\rho_i^{s_i} \right). \tag{4}$$

Recall that

$$\overline{s} \in S_{\overline{r}} \Leftrightarrow \begin{cases} s_i \geq r_i & \text{iff } r_i = m_i, \\ s_i = r_i & \text{otherwise.} \end{cases}$$

According to Jackson’s Theorem we can consider the dimensions independently from each other. Hence, for every dimension  $i = 1, \dots, n$ , a state  $\bar{s} \in S_{\bar{\tau}}$  may take all values  $j \geq m_i$  in case  $r_i = m_i$  and the value  $r_i$  in case  $r_i \neq m_i$ . Thus,

$$\sum_{\bar{s} \in S_{\bar{\tau}}} p(\bar{s}) = \prod_{i=1}^n \Omega(i), \text{ with } \Omega(i) = \begin{cases} (1 - \rho_i)\rho_i^{r_i} & \text{for } r_i \neq m_i, \\ \sum_{j=m_i}^{\infty} (1 - \rho_i)\rho_i^j & \text{for } r_i = m_i. \end{cases} \quad (5)$$

The infinite sum can be rewritten

$$\sum_{j=m_i}^{\infty} (1 - \rho_i)\rho_i^j = (1 - \rho_i) \sum_{j=m_i}^{\infty} \rho_i^j = (1 - \rho_i) \frac{\rho_i^{m_i}}{1 - \rho_i} = \rho_i^{m_i}$$

and replaced to match (3):

$$\Omega(i) = \begin{cases} (1 - \rho_i)\rho_i^{r_i} & \text{for } r_i \neq m_i, \\ \rho_i^{m_i} & \text{for } r_i = m_i. \end{cases}$$

□

**Example 5**

We want to check the CSL formula  $S_{\bowtie p}((s_1 \geq 2) \wedge (s_2 \geq 3))$ . Recall from Example 3 that all states  $\bar{r} \in R((2, 3)) = \{(0, 3), (1, 3), (2, 0), (2, 1), (2, 3)\}$  satisfy  $ap_1 = (s_1 \geq 2) \wedge (s_2 \geq 3)$  and the boundary states do not satisfy  $ap_1$ . Using (2) and (3) and accumulating the probabilities for all representative states  $\bar{r} \in R((2, 3))$  we obtain

$$\begin{aligned} \bar{s} \models S_{\bowtie p}((s_1 \geq 2) \wedge (s_2 \geq 3)) &\Leftrightarrow \\ ((1 - \rho_1)\rho_2^3 + (1 - \rho_1)\rho_1\rho_2^3 + \rho_1^2(1 - \rho_2) + \rho_1^2(1 - \rho_2)\rho_2 + \rho_1^2\rho_2^3) &\bowtie p. \end{aligned} \quad (6)$$

**5.3 Time-Bounded Next Operator**

The time-bounded next operator for LJQNs is computed just as for QBDs [15]. Recall that a state  $\bar{s}$  satisfies  $\mathcal{P}_{\bowtie p}(\mathcal{X}^{[t_1, t_2]}\Phi)$  if the one-step probability to reach a state that fulfills  $\Phi$  within a time  $t \in [t_1, t_2]$ , outgoing from  $\bar{s}$  meets the bound  $p$ . The possibly infinite summation over all  $\Phi$ -states can be truncated by only considering those  $\Phi$ -states that can be reached in one step from  $\bar{s}$ . We define the set of states that is reachable in one step from state  $\bar{s}$  as  $B_{\bar{s}} = \{\bar{s}' \in S \mid G(\bar{s}, \bar{s}') > 0\}$ . Note that  $B_{\bar{s}}$  is always finite. We then have:

$$\begin{aligned} \bar{s} \models \mathcal{P}_{\bowtie p}(\mathcal{X}^{[t_1, t_2]}\Phi) &\Leftrightarrow \Pr\{\sigma \in Path(\bar{s}) \mid \sigma \models \mathcal{X}^{[t_1, t_2]}\Phi\} \bowtie p \\ &\Leftrightarrow \left( \left( e^{G(\bar{s}, \bar{s}) \cdot t_1} - e^{G(\bar{s}, \bar{s}) \cdot t_2} \right) \cdot \sum_{\substack{\bar{s}' \in \text{Sat}(\Phi) \cap B_{\bar{s}} \\ \bar{s} \neq \bar{s}'}} \frac{G(\bar{s}, \bar{s}')}{-G(\bar{s}, \bar{s})} \right) \bowtie p, \end{aligned} \quad (7)$$

where  $e^{G(\bar{s}, \bar{s}) \cdot t_1} - e^{G(\bar{s}, \bar{s}) \cdot t_2}$  is the probability of residing in  $\bar{s}$  for a time  $t \in [t_1, t_2]$ , and  $\frac{G(\bar{s}, \bar{s}')}{-G(\bar{s}, \bar{s})}$  specifies the probability to step from state  $\bar{s}$  to state  $\bar{s}'$ , provided a step takes place.

Now, let the inner formula  $\Phi$  be independent as of  $\bar{m}$ . Hence, the validity of  $\Phi$  might be different for all states  $\bar{s} \in S_b$ . Therefore, the representative states  $\bar{r} \in R(\bar{m})$  may satisfy  $\mathcal{P}_{\bowtie p}(\mathcal{X}^{[t_1, t_2]}\Phi)$ , whereas the remaining states  $\bar{s} \in S_{\bar{r}}$  do not necessarily satisfy  $\mathcal{P}_{\bowtie p}(\mathcal{X}^{[t_1, t_2]}\Phi)$ , since  $S_b$  is reachable in one step. However, from  $\bar{m} + \bar{1}$  (with  $\bar{1} = (1, 1, \dots, 1, 1)$ ) onwards, with one step only states with equivalent  $\Phi$  validity can be reached. Thus, in case the inner formula  $\Phi$  is independent as of  $\bar{m}$ ,  $\mathcal{P}_{\bowtie p}(\mathcal{X}^{[t_1, t_2]}\Phi)$  is independent as of  $\bar{m} + \bar{1}$ . For the construction of the satisfaction set of such a formula we have to compute explicitly the satisfying states in  $S_b$ .  $Sat^{R(\bar{m} + \bar{1})}(\mathcal{P}_{\bowtie p}(\mathcal{X}^{[t_1, t_2]}\Phi))$  then provides the validity of  $\mathcal{P}_{\bowtie p}(\mathcal{X}^{[t_1, t_2]}\Phi)$  for the remaining infinite state space  $S \setminus S_b$ .

#### 5.4 Until Operator for $I = [0, t]$

For model checking  $\mathcal{P}_{\bowtie p}(\Phi \mathcal{U}^I \Psi)$  we adopt the general approach for finite CTMCs [3] and QBDs [15]. Recall that the CSL path formula  $\varphi = \Phi \mathcal{U}^I \Psi$  is valid if a  $\Psi$ -state is reached on a path during the time interval  $[t_1, t_2]$  via only  $\Phi$ -states. We restrict ourselves to intervals of the form  $I = [0, t]$ . The future behavior of the LJQN is then irrelevant for the validity of  $\varphi$ , as soon as a  $\Psi$ -state is reached. Thus all  $\Psi$ -states can be made absorbing without affecting the satisfaction set of formula  $\varphi$ . On the other hand, as soon as a  $(\neg\Phi \wedge \neg\Psi)$ -state is reached,  $\varphi$  will be invalid, regardless of the future evolution.

As a result of the above consideration, we may switch from checking the LJQN  $\mathcal{J}$  to checking a new, derived, LJQN, denoted as  $\mathcal{J}[\Psi][\neg\Phi \wedge \neg\Psi] = \mathcal{J}[\neg\Phi \vee \Psi]$ , where all states in the underlying Markov chain that satisfy the formula in square brackets are made absorbing. The generator function  $\tilde{\mathbf{G}}(\bar{s}, \bar{s}')$  for  $\mathcal{J}[\neg\Phi \vee \Psi]$  is then defined as

$$\tilde{\mathbf{G}}(\bar{s}, \bar{s}') = \begin{cases} \mathbf{G}(\bar{s}, \bar{s}'), & \bar{s} \notin \neg\Phi \vee \Psi, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

for  $\bar{s} \neq \bar{s}'$ .  $\tilde{\mathbf{G}}(\bar{s}, \bar{s})$  is adapted accordingly. Model checking a formula involving the until operator then reduces to calculating the transient probabilities  $\pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\bar{s}, \bar{s}', t)$  for all  $\Psi$ -states  $\bar{s}'$ . Exploiting the structure of LJQNs yields

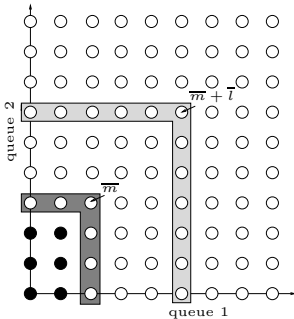
$$\begin{aligned} s \models \mathcal{P}_{\bowtie p}(\Phi \mathcal{U}^{[0, t]}\Psi) &\Leftrightarrow Prob^{\mathcal{J}}(s, \Phi \mathcal{U}^{[0, t]}\Psi) \bowtie p \\ &\Leftrightarrow \left( \sum_{\bar{s}' \in Sat^{S_b}(\Psi)} \pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\bar{s}, \bar{s}', t) + \sum_{i=0}^{\infty} \sum_{s' \in Sat^{R(\bar{m} + \bar{i})}(\Psi)} \pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\bar{s}, \bar{s}', t) \right) \bowtie p. \end{aligned} \quad (9)$$

The transient probabilities are accumulated for the  $\Psi$  states in  $S_b$  and for the  $\Psi$  states in the representative front  $R(\bar{m} + \bar{i})$  for  $i \in \mathbf{N}$ . Note that these representative fronts are situated in layers around  $S_b$  and cover the infinite state space  $S \setminus S_b$  for  $i \in \mathbf{N}$ .

In [15] we have shown how transient probabilities can be computed for QBDs using uniformization [9] by considering only a finite fraction of the infinite state

space. As standard property of uniformization, the finite time bound  $t$  is transformed to a finite number of steps  $l$  [9]. To do the same for LJQNs, first the probability function  $\mathbf{P}(\bar{s}, \bar{s}')$  (or for  $\tilde{\mathbf{P}}$ ) for the embedded DTMC is defined as  $\mathbf{P}(\bar{s}, \bar{s}') = \frac{\mathbf{G}(\bar{s}, \bar{s}')}{\nu}$  for  $\bar{s} \neq \bar{s}'$  and  $\mathbf{P}(\bar{s}, \bar{s}) = \frac{\mathbf{G}(\bar{s}, \bar{s})}{\nu} + 1$ . The uniformization constant  $\nu$  must be at least equal to the maximum of the negative rates  $\mathbf{G}(\bar{s}, \bar{s})$ ; for LJQNs, the value  $\nu = \lambda + \sum_{i=1}^n \mu_i$  suffices. For an allowed maximal numerical error  $\varepsilon$ , uniformization requires a finite number  $l$  of steps (state changes) to be taken into account in order to compute the transient probabilities;  $l$  can be computed *a priori*, given  $\varepsilon$ ,  $\nu$  and  $t$ . Summarizing, we have obtained the following result:

If  $\neg\Phi \vee \Psi$  is independent as of  $\bar{m}$  then using uniformization with  $l$  steps, we obtain the same transient probabilities for states  $\bar{s} \in S_{\bar{\tau}}$ , with  $\bar{\tau} \in R(\bar{m} + \bar{l})$ , since from all states  $\bar{s} \in S_{\bar{\tau}}$ , only corresponding states can be reached when taking  $l$  steps in the LJQN.



**Example 6**

The figure to the left shows the finite fraction of the infinite state space that is needed to compute the validity of  $\mathcal{P}_{\infty p}(\Phi \mathcal{U}^l \Psi)$ . Starting from every representative state  $\bar{\tau} \in R(\bar{m} + \bar{l})$ , still  $l$  steps can be undertaken in every direction without reaching the boundary set  $S_b$ . The total amount of states we have to consider equals  $(m_1 + l) \cdot (m_2 + l)$ , from which  $m_1 + m_2 + 2l - 1$  are representative states.

For all states  $s \in \mathcal{S}$ , we add the computed transient probabilities to reach any  $\Psi$ -state and check whether the accumulated probability meets the bound  $p$ . We define the accumulated probability for up to  $l$  steps in the uniformized Markov chain as:

$$\tilde{\pi}^{(l)} = \sum_{\bar{s}' \in \text{Sat}^{S_b}(\Psi)} \pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\bar{s}, \bar{s}', t) + \sum_{i=0}^{l-1} \sum_{s' \in \text{Sat}^{R(\bar{m} + \bar{l})}(\Psi)} \pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\bar{s}, \bar{s}', t) \quad (10)$$

Note that the above expression, for  $l \rightarrow \infty$ , equals the exact probability that is to be compared with  $p$  in (9). Once the accumulated probabilities are calculated, similar inequalities as presented in [15], can be used to decide the validity of  $\mathcal{P}_{\infty p}(\Phi \mathcal{U}^l \Psi)$  on LJQNs. The accumulated probability is always an underestimation of the actual probability. The value of the maximum error  $\varepsilon$  accumulated for all states and depending on the number of steps  $l$  decreases as  $l$  increases. Thus, we obtain the following implications:

- (a)  $\tilde{\pi}^{(l)}(s, \text{Sat}(\Psi), t) > p \Rightarrow \pi(s, \text{Sat}(\Psi), t) > p$
- (b)  $\tilde{\pi}^{(l)}(s, \text{Sat}(\Psi), t) < p - \varepsilon \Rightarrow \pi(s, \text{Sat}(\Psi), t) < p$

If one of these inequalities (a) or (b) holds, we can decide that the bound  $< p$  or  $> p$  is met. For the bounds  $\leq p$  and  $\geq p$ , similar implications can be

given. If  $\tilde{\pi}(s, \text{Sat}(\Psi), t) \in [p, p - \varepsilon]$ , we cannot decide whether  $\pi(s, \text{Sat}(\Psi), t)$  meets the bound  $p$ . In this case, increasing  $l$  resolves the problem. However, note that in case  $p = \pi(s, \text{Sat}(\psi), t)$  we cannot decide and iteration does not stop. As already mentioned, for all representative states  $\bar{\tau} \in R(\bar{m} + \bar{l})$  the transient probabilities for all  $\bar{s} \in S_{\bar{\tau}}$  computed with  $l$  steps will be the same. Thus, if we can decide whether the bound  $p$  is met (case (a) or (b) above), we can be sure that  $\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^{[0,t]}\Psi)$  is independent as of  $\bar{m} + \bar{l}$ . In that case, we check for all states  $\bar{s} \leq \bar{m} + \bar{l}$  whether the accumulated transient probability of reaching a  $\Psi$ -state meets the bound  $p$ . The states  $\bar{s} \in S_b$  that satisfy  $\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^{[0,t]}\Psi)$  form the boundary satisfaction set  $\text{Sat}^{S_b}$  and the representative states that satisfy  $\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^{[0,t]}\Psi)$  form the representative satisfaction set  $\text{Sat}^{R(\bar{m} + \bar{l})}(\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^{[0,t]}\Psi))$ .

For model checking the until operator we need to consider  $\prod_{i=1}^n (m_i + 2l)$  states in total, depending on the number of steps  $l$  considered, the dimension  $n$  of the LJQN, and the independence vector  $\bar{m}$  of the inner formula. Hence,  $\prod_{i=1}^n (m_i + 1) - \prod_{i=1}^n m_i$  representative states suffice to represent the infinite states space  $S \setminus S_b$ .

In [16] we presented a method to efficiently compute the transient probabilities in QBDs and check the until operator with a so-called *dynamic stopping criterion*. This method can also be applied to LJQNs. When iteratively computing the approximation  $\tilde{\pi}^{(l)}(s, \text{Sat}(\Psi), t)$  and regularly checking whether either (a) or (b) holds for all starting states, the number of considered steps with uniformization is minimized. It is expected that such an approach will lead to similar efficiency improvements.

## 6 Conclusions

In this paper we presented model checking algorithms for checking CSL properties for a very general class of queueing networks, namely for labeled Jackson queueing networks. The underlying state space of such LJQNs is a highly structured CTMC that is, of infinite size in as many dimensions as there are queues. We introduce a new notion of property independence on LJQNs that is needed for model checking. Steady-state probabilities are computed in (stable) LJQNs with well-known product-form results and transient probabilities are computed with an adaption of our uniformization-based approach. We provided a running example throughout the paper to illustrate our approach.

Note that the model checking procedure for LJQN as introduced in this paper, and the model checking procedure for QBDs [15], in their simplest setting, i.e., the case that the model is a simple M|M|1 queue (which is both a QBD and a LJQN), are in essence the same.

At various points, the presented algorithms can be made more efficient. For instance, applying a dynamic stopping criterion (as in [16]) when model checking the until operator in LJQNs might considerably decrease the number of states that has to be taken into account. To do so, we need to develop efficient data structures to store the step-wise computed probabilities.

The complexity for checking the steady-state operator is linear in the number of queues and the complexity for checking the until operator is given by matrix vector multiplications of matrices that are exponential in the number of queues.

We restricted ourselves to atomic propositions of the form  $\bigwedge_{i=1}^n (s_i \lesseqgtr m_i)$  for  $m_i \in \mathbb{N}$ ; this facilitates the determination of the independence vector  $\overline{m}$ . For model checking other properties like the balance of queues or a general threshold, the notion of property independence needs to be (and can be) generalized.

Finally, we developed the model checking algorithm for the until operator with time-bound in  $[0, t]$ , however, other time intervals can be handled similarly as for finite state CTMCs or QBDs [3,17]. In further work, we will introduce rewards to allow for CSRL model checking. Furthermore, we also consider addressing networks of QBDs. However, to the best of our knowledge, there is no method available to compute steady-state probabilities on such multi-dimensional QBDs, hence, we will have to do without the steady-state operator.

## References

1. Abdulla, P., Jonsson, B., Nilsson, M., Saksena, M.: A survey of regular model checking. In: Gardner, P., Yoshida, N. (eds.) CONCUR 2004. LNCS, vol. 3170, pp. 35–48. Springer, Heidelberg (2004)
2. Aziz, A., Sanwal, K., Brayton, R.: Model checking continuous-time Markov chains. *ACM Transactions on Computational Logic* 1(1), 162–170 (2000)
3. Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering* 29(7), 524–541 (2003)
4. Ballarini, P., Hillston, J.: Compositional csl model checking for boucherie product processes. In: 3rd Workshop on Automated Verification of Critical Systems (AV-OCS'03), DSSE Technical Report DSSE-TR-2003-2 (2003)
5. Brázdil, T., Kucera, A., Strazovský, O.: On the decidability of temporal properties of probabilistic pushdown automata. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, pp. 145–157. Springer, Heidelberg (2005)
6. Ciardo, G.: Discrete-time Markovian stochastic Petri nets. In: *Computations with Markov Chains*, pp. 339–358. Raleigh (1995)
7. Clarke, E., Emerson, E., Sistla, A.: Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems* 8(2), 244–263 (1986)
8. Etesami, K., Yannakakis, M.: Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, pp. 340–352. Springer, Heidelberg (2005)
9. Gross, D., Miller, D.R.: The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research* 32(2), 343–361 (1984)
10. Harrison, P.G.: Transient behaviour of queueing networks. *Journal of Applied Probability* 18(2), 482–490 (1981)
11. Jackson, J.R.: Networks of waiting lines. *Operations Research* 5(4), 518–521 (1957)
12. Jackson, J.R.: Jobshop-like queueing systems. *Management Science* 10(1), 131–142 (1963)

13. Kleinrock, L.: *Queueing Systems. Theory*, vol. 1. John Wiley & Sons, Chichester (1975)
14. Matis, T.I., Feldman, R.M.: Transient analysis of state-dependent queueing networks via cumulant functions. *Journal of Applied Probability* 38(4), 841–859 (2001)
15. Remke, A., Haverkort, B.R., Cloth, L.: Model checking infinite-state markov chains. In: Halbwachs, N., Zuck, L.D. (eds.) *TACAS 2005*. LNCS, vol. 3440, pp. 237–252. Springer, Heidelberg (2005)
16. Remke, A., Haverkort, B.R., Cloth, L.: Uniformization with representatives. In: *Proc. Workshop on Tools for solving structured Markov chains*, ACM press (CD only) (2006)
17. Remke, A., Haverkort, B.R., Cloth, L.: CSL model checking algorithms for QBDs. *Theoretical Computer Science* (2007) doi: 10.1016/j.tcs.2007.05.007
18. Schnoebelen, P.: The verification of probabilistic lossy channel systems. In: Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P., Siegle, M. (eds.) *Validation of Stochastic Systems*. LNCS, vol. 2925, pp. 445–465. Springer, Heidelberg (2004)
19. van Moorsel, A.P.A.: *Performability Evaluation Concepts and Techniques*. PhD thesis, Dept. of Computer Science, University Twente (1993)