Model Checking CSL Until Formulae with Random Time Bounds*

Marta Kwiatkowska[1], Gethin Norman[1] and António Pacheco[2]

[1] School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom
[2] Department of Mathematics and CEMAT, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Abstract. Continuous Time Markov Chains (CTMCs) are widely used as the underlying stochastic process in performance and dependability analysis. Model checking of CTMCs against Continuous Stochastic Logic (CSL) has been investigated previously by a number of authors [2, 4, 13]. CSL contains a time-bounded until operator that allows one to express properties such as "the probability of 3 servers becoming faulty within 7.01 seconds is at most 0.1". In this paper we extend CSL with a random time-bounded until operator, where the time bound is given by a random variable instead of a fixed real-valued time (or interval). With the help of such an operator we can state that the probability of reaching a set of goal states within some generally distributed delay while passing only through states that satisfy a certain property is at most (at least) some probability threshold. In addition, certain transient properties of systems which contain general distributions can be expressed with the extended logic. We extend the efficient model checking of CTMCs against the logic CSL developed in [13] to cater for the new operator. Our method involves precomputing a family of coefficients for a range of random variables which includes Pareto, uniform and gamma distributions, but otherwise carries the same computational cost as that for ordinary time-bounded until in [13]. We implement the algorithms in MATLAB and evaluate them by means of a queueing system example.

1 Introduction

Continuous time Markov chains (CTMCs) are widely used as the underlying stochastic process in performance and dependability analysis. CTMCs are characterised by allowing only exponential distributions – the time that the system remains in a state is governed by a (negative) exponential distribution. The restriction to exponential distributions yields efficient analysis techniques for both transient and steady-state probabilities, and hence also for calculating standard performance measures such as throughput, mean waiting time and average cost. Recently extensions of temporal logic have been proposed which can express such

* Partly supported by EPSRC grant GR/M13046, FCT, grant SFRH/BSAB/ 251/01 and the projects POSI/34826/CPS/2000 and POSI/42069/CPS/2001.

H. Hermanns and R. Segala (Eds.), 2nd Joint Int. Workshop on Process Algebra and Performance Modelling and Probabilistic Methods in Verification, volume 2399 of LNCS, pages 152–168, 2002.

properties. The temporal logic CSL (Continuous Stochastic Logic) introduced by Aziz *et al.* [1, 2] and since extended by Baier *et al.* [4] is based on the temporal logics CTL [5] and PCTL [11] and provides a powerful means to specify both path-based and traditional state-based performance measures on CTMCs in a concise, flexible and unambiguous way. CSL contains a time-bounded until operator that allows one to express properties such as "the probability of 3 servers becoming faulty within 7.01 seconds is at most 0.1" (more generally, one can additionally require that the executions step through states that satisfy a given property). Model checking of CTMCs against CSL has been improved in [3, 13] through the use of uniformisation [10, 12] and transient analysis, and the usefulness of this approach demonstrated by a number of case studies.

However, in practice it is often the case that exponential distributions are not an adequate modelling tool for capturing the behaviour of stochastic systems. Examples of such situations include modelling file transfer over the Internet, timeouts in communication protocols and the residence time in a wireless cell. For these cases the modelling framework must be capable of handling *general distributions*, such as Pareto, Erlang, gamma or phase-type. An unfortunate consequence of including general distributions within the modelling framework, as has been demonstrated recently, for example in [8] and [15], is a considerable increase in the complexity of performance analysis, or if using phase-type distributions a substantial increase in the size of the state space.

In this paper we make an alternative proposal, namely, to remain in the CTMC framework and instead extend the logic CSL with a variant of the time-bounded until operator which allows *random* (generally distributed) *time bound*, by replacing the constant time bound with a random variable. With the new operator one can specify properties such as the probability of reaching a set of goal states of a CTMC within some generally distributed delay while passing only through states that satisfy a certain property is at most (at least) some probability value. Although general distributions are not added explicitly to the model, using the random time-bounded until operator enables one to establish specific transient properties of systems which include generally distributed delays. As an example application consider a queue where the customers arrive with some generally distributed delay, then by letting the random time bound have *the same* distribution we can express (and verify) properties such as: "if the queue if full then, with probability at least p, when the next customer arrives there will be at most k customers in the queue".

The semantics of the new random time-bounded until operator involves a Riemann-Stieltjes integral, and so one would expect that the resulting complexity of its model checking is prohibitive. The central observation of this paper is that this integral reduces to an infinite summation involving mixed Poisson probabilities (equivalent to the α-factors of [8]). This result is similar to the case of the ordinary time-bounded until, except for an additional vector of coefficients that can be precomputed beforehand. Thanks to this observation we can propose an efficient model checking algorithm for the new operator, which is derived from that of [13] and which does not carry an increase in computational cost except the precomputation of coefficients. This somewhat surprising result yields a powerful and fast method for analysing certain properties of stochastic systems with generally distributed delays. Moreover, for a large class of general distributions which includes Pareto, gamma and Erlang that have been observed in stochastic systems, we also provide methods for pre-computing the mixed Poisson probabilities.

Finally, we model a queueing system and describe, with the help of a MATLAB implementation of the algorithms, the experimental results obtained when verifying the system against random time-bounded until formulae. We are able to demonstrate the inaccuracy of using exponential distributions where the actual arrivals warrant the use of heavy-tailed or other general distributions.

Outline of paper: We begin by recalling the definition of CTMCs and the logic CSL. Next we introduce the new random time-bounded until operator, give its semantics and a model checking algorithm based on [13] which uses a family of coefficients. In the remainder of the paper we calculate the coefficients for a number of well-known distributions and describe the experimental results.

2 Continuous Time Markov Chains and the logic CSL

In this section we briefly recall the basic concepts of CTMCs and the logic CSL, concentrating on the time bounded until operator. Let AP be a finite set of atomic propositions.

Definition 1. *A (labelled) CTMC $\mathcal{C}$ is a tuple $(S, \mathbf{R}, L)$ where S is a finite set of* states, *$\mathbf{R} : S \times S \to \mathbb{R}_{\geq 0}$ is the* rate matrix *and $L : S \to 2^{AP}$ is a* labelling function *which assigns to each state $s \in S$ the set $L(s)$ of atomic propositions that are valid in s.*

For any state $s \in S$, the probability of leaving state s within t time units is given by $1 - e^{-E(s)\cdot t}$ where $E(s) = \sum_{s' \in S} \mathbf{R}(s, s')$. If $\mathbf{R}(s, s') > 0$ for more than one $s' \in S$, then there is a *race* between the transitions leaving s, where the probability of moving to s' in a single step equals the probability that the delay corresponding to moving from s to s' "finishes before" the delays of any other transition leaving s. Note that, as in [3, 4], we do allow self-loops.

A path through a CTMC is an alternating sequence $\sigma = s_0\, t_0\, s_1\, t_1\, s_2 \ldots$ such that $\mathbf{R}(s_i, s_{i+1}) > 0$ and $t_i \in \mathbb{R}_{>0}$ for all $i \geq 0$. The time stamps t_i denote the amount of time spent in state s_i. Let $Path^{\mathcal{C}}(s)$ denote the set of paths of $\mathcal{C}$ which start in state s (i.e. $s_0 = s$); $\sigma@t$ denote the state of σ occupied at time t, i.e. $\sigma@t = \sigma[i]$ where i is the largest index such that $\sum_{j=0}^{i-1} t_j \leq t$; and $\Pr_s$ denote the unique probability measure on sets of paths that start in s [4].

We now recall the logic CSL first introduced in [1, 2] and extended in [4].

Definition 2 (Syntax of CSL). *The syntax of* CSL *is defined as follows:*

$$\Phi ::= \texttt{true} \;\Big|\; a \;\Big|\; \Phi \wedge \Phi \;\Big|\; \neg\Phi \;\Big|\; \mathcal{S}_{\bowtie p}(\Phi) \;\Big|\; \mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{\leq t}\, \Phi)$$

where $a \in$ AP, $p \in [0, 1]$, $t \in \mathbb{R}_{>0}$ and $\bowtie \in \{\leq, \geq\}$.

The semantics of CSL for can be found in [4]. Here we concentrate on the time bounded until operator defined by:

$$s \models \mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{\leq t}\, \Psi) \quad \Leftrightarrow \quad Prob^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi) \bowtie p$$

where $Prob^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi)$ is given by:

$$Prob^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi) \stackrel{\text{def}}{=} \Pr_s\{\sigma \in Path^{\mathcal{C}}(s) \mid \sigma \models \Phi\, \mathcal{U}^{\leq t}\, \Psi\}$$

and $\Phi\, \mathcal{U}^{\leq t}\, \Psi$ asserts that Ψ will be satisfied at some time instant in the interval $[0, t]$ and that at all preceding time instants Φ holds:

$$\sigma \models \Phi\, \mathcal{U}^{\leq t}\, \Psi \quad \Leftrightarrow \quad \exists x \leq t.\, (\sigma@x \models \Psi \wedge \forall y < x.\, \sigma@y \models \Phi)\ .$$

In the remainder of this section we describe the model checking algorithm for time bounded until formulae originally presented in [13], based on a reduction to transient analysis and model checking PCTL. Further details on model checking the other CSL operators are available in, for example, [4, 3, 13]. To begin with we define uniformisation, a transformation of a CTMC into a DTMC (discrete-time Markov chain).

Definition 3. *For CTMC $\mathcal{C} = (S, \mathbf{R}, L)$ the* uniformised *DTMC is given by $unif(\mathcal{C}) = (S, \mathbf{P}^{unif(\mathcal{C})}, L)$ where $\mathbf{P} = \mathbf{I} + \mathbf{Q}/q$ for $q \geq \max\{E(s) \mid s \in S\}$ and $\mathbf{Q} = \mathbf{R} - diag(\underline{E})$.*

Using this transformation, in [13, Proposition 2] it is shown that for any $t \in \mathbb{R}$:

$$Prob^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi) = \sum_{k=0}^{\infty} \gamma(k, q{\cdot}t){\cdot}Prob^{unif(\mathcal{C})}(s, \Phi\, \mathcal{U}^{\leq k}\, \Psi) \tag{1}$$

where $\gamma(k, q{\cdot}t)$ is the kth Poisson probability with parameter $q{\cdot}t$, i.e. $\gamma(k, q{\cdot}t) = e^{-q\cdot t}{\cdot}(q{\cdot}t)^k/k!$, and $Prob^{unif(\mathcal{C})}(s, \Phi\, \mathcal{U}^{\leq k}\, \Psi)$ is the probability that, in the DTMC $unif(\mathcal{C})$, from s a state satisfying Ψ is reached within k (discrete) steps while passing through only states that satisfy Φ.

Applying the model checking algorithm for PCTL presented in [11], calculating $Prob^{unif(\mathcal{C})}(s, \Phi\, \mathcal{U}^{\leq k}\, \Psi)$ for all $s \in S$ reduces to computing the vector of probabilities $(\mathbf{P}^{unif(\mathcal{C})[\neg\Phi\vee\Psi]})^k{\cdot}\underline{\iota_\Psi}$, where for any $s, s' \in S$:

$$\mathbf{P}^{unif(\mathcal{C})[\neg\Phi\vee\Psi]}(s, s') \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } s \models \neg\Phi \vee \Psi \text{ and } s = s' \\ 0 & \text{if } s \models \neg\Phi \vee \Psi \text{ and } s \neq s' \\ \mathbf{P}^{unif(\mathcal{C})}(s, s') & \text{otherwise} \end{cases}$$

and $\underline{\iota_\Psi}$ characterises $Sat(\Psi)$, i.e. $\underline{\iota_\Psi}(s) = 1$ if $s \models \Psi$, and 0 otherwise. It then follows that calculating $Prob(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi)$ for *all* states amounts to computing the following vector of probabilities:

$$\underline{Prob}(\Phi\, \mathcal{U}^{\leq t}\, \Psi) = \sum_{k=0}^{\infty} \gamma(k, q{\cdot}t){\cdot}(\mathbf{P}^{unif(\mathcal{C})[\neg\Phi\vee\Psi]})^k{\cdot}\underline{\iota_\Psi}\ .$$

Note that, as iterative squaring is not attractive for stochastic matrices due to fill-in [20], the matrix product is typically computed in an iterative fashion: $\mathbf{P}^{unif(\mathcal{C})[\neg\Phi\vee\Psi]}{\cdot}(\cdots(\mathbf{P}^{unif(\mathcal{C})[\neg\Phi\vee\Psi]}{\cdot}\underline{\iota_\Psi}))$.

3 Random time-bounded until formulae

In this section we extend the logic CSL to include *random* time-bounded until formulae and consider model checking algorithms for such formulae. Let T denote a *nonnegative random variable*. We let F_T denote the (cumulative) distribution function of T (i.e. $F_T(t) = \mathbf{P}[T \leq t]$), and assume that the support of T is contained in the interval $[L_T, R_T]$, where L_T may either be zero or positive and R_T may either be finite or infinite. Note that, in particular, we have $F_T(t)$ is zero for $t < L_T$ and one for $t \geq F_T$.

We now extend CSL by allowing formulae of the form $\mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{\leq T}\, \Psi)$ where T is a *nonnegative random variable*[3]. The formula asserts that, with probability $\bowtie p$, by the *random* time T a state satisfying Ψ will be reached such that all preceding states satisfy Φ. Formally, the semantics is given by:

$$s \models \mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{\leq T}\, \Psi) \quad \Leftrightarrow \quad \mathit{Prob}^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq T}\, \Psi) \bowtie p$$

where $\mathit{Prob}^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq T}\, \Psi)$ is defined as a Riemann-Stieltjes integral involving (deterministic) time-bounded until probabilities:

$$\mathit{Prob}^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq T}\, \Psi) \stackrel{\text{def}}{=} \int_{L_T}^{R_T} \mathit{Prob}^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi)\, dF_T(t)\,.$$

We now give our main observation concerning model checking random time-bounded until formulae, which allows us to replace the Riemann-Stieltjes integral with summation, at the cost of pre-computing coefficients.

Proposition 1. *For any $s \in S$:*

$$\mathit{Prob}^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq T}\, \Psi) = \sum_{k=0}^{\infty} \alpha_T(k, q) \cdot \mathit{Prob}^{\mathrm{unif}(\mathcal{C})}(s, \Phi\, \mathcal{U}^{\leq k}\, \Psi)$$

where $\alpha_T(k, q) = \int_{L_T}^{R_T} \gamma(k, q{\cdot}t)\, dF_T(t)$.

Proof. By definition we have:

$$\begin{aligned}
\mathit{Prob}^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq T}\, \Psi) &= \int_{L_T}^{R_T} \mathit{Prob}^{\mathcal{C}}(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi)\, dF_T(t) \\
&= \int_{L_T}^{R_T} \left(\sum_{k=0}^{\infty} \gamma(k, q{\cdot}t) \cdot \mathit{Prob}^{\mathrm{unif}(\mathcal{C})}(s, \Phi\, \mathcal{U}^{\leq k}\, \Psi) \right) dF_T(t) && \text{by (1)} \\
&= \sum_{k=0}^{\infty} \left(\int_{L_T}^{R_T} \gamma(k, q{\cdot}t)\, dF_T(t) \right) \cdot \mathit{Prob}^{\mathrm{unif}(\mathcal{C})}(s, \Phi\, \mathcal{U}^{\leq k}\, \Psi) && \text{rearranging} \\
&= \sum_{k=0}^{\infty} \alpha_T(k, q) \cdot \mathit{Prob}^{\mathrm{unif}(\mathcal{C})}(s, \Phi\, \mathcal{U}^{\leq k}\, \Psi) && \text{by definition}
\end{aligned}$$

as required. □

[3] One condition we impose on the random variables is that they are *independent* of the CTMC under study.

It then follows from the algorithm of [13] for calculating $Prob(s, \Phi\, \mathcal{U}^{\leq t}\, \Psi)$ outlined in Section 2 that calculating $Prob(s, \Phi\, \mathcal{U}^{\leq T}\, \Psi)$ for all states of the CTMC reduces to computing the following vector of probabilities:

$$\underline{Prob}(\Phi\, \mathcal{U}^{\leq T}\, \Psi) = \sum_{k=0}^{\infty} \alpha_T(k,q)\cdot(P^{unif(\mathcal{C})[\neg\Phi\vee\Psi]})^k\cdot \underline{\iota}_{\Psi}\,. \tag{2}$$

The coefficients $\alpha_T(k,q)$ take values on the interval $[0,1]$ since they are probabilities. Namely, $\alpha_T(k,q) = \mathbf{P}(N_q(T) = k)$ where N_q is a Poisson process with rate q, independent of T. Moreover,

$$\sum_{k=0}^{\infty} \alpha_T(k,q) = \sum_{k=0}^{\infty} P(N_q(T) = k) = 1\,.$$

Following Grandell [9], we call the coefficients $\{\alpha_T(k,q)\}_{k\in\mathbb{N}}$ *mixed Poisson probabilities*; $\alpha_T(k,q)$ is the k-th probability of a Poisson process with rate q at random time T. Note that the mixed Poisson probabilities are equivalent to the α-factors introduced in [8] which are used in the calculation of steady-state probabilities for non-Markovian stochastic Petri nets.

Figure 1 presents the pseudo-code for a generic algorithm for computing $Prob^{\mathcal{C}}(s, \Phi\mathcal{U}^{\leq T}\, \Psi)$ with an error of at most ε. Note that the DTMC $unif(\mathcal{C})[\neg\Phi\vee\Psi]$ may reach steady state before $R(\epsilon)$ and, in this case, the summation can be truncated at this earlier point [17]. In the next section we will consider methods for calculating these coefficients and the bound $R(\varepsilon)$.

```
input : α_T(0,q), ..., α_T(R(ε),q) such that Σ_{k=R(ε)+1}^∞ α_T(k,q) < ε
P := P^{unif(C)[¬Φ∨Ψ]}
b := ι_Ψ
sol := 0
for k = 0 to R(ε)
    sol := sol + α_T(k,q)·b
    b := P·b
endfor
// Prob(Φ U^{≤T} Ψ) = sol
```

Fig. 1. Generic algorithm for computing $\underline{Prob}(\Phi\, \mathcal{U}^{\leq T}\, \Psi)$

4 Computation of the mixed Poisson probabilities

Under complete generality the computation of mixed Poisson probabilities relies on the evaluation of the integrals:

$$\alpha_T(k,q) = \int_{L_T}^{R_T} \gamma(k, q\cdot t)\, dF_T(t) \quad \text{for } k \in \mathbb{N}.$$

Next we will develop algorithms to compute the mixed Poisson probabilities $\alpha_T(k,q)$ for the case where the distribution of the random time T has a finite discrete, uniform, gamma or Pareto distribution, or is a finite mixture of distributions of these types. These algorithms can then be integrated with that presented in Figure 1 to compute $\underline{Prob}(\Phi\ \mathcal{U}^{\leq T}\ \Psi)$ when the distribution of T belongs to this class of distributions.

We start by showing how the mixed Poisson probabilities for a finite mixture can be computed in terms of the mixed Poisson probabilities of the random times involved in the mixture.

4.1 Finite mixture of random times

Suppose that T is a mixture of n random variables, $T_1, T_2, \ldots, T_n$ with weights $a_1, a_2, \ldots, a_n$ ($a_i > 0$ for $1 \leq i \leq n$ and $\sum_{i=1}^{n} a_i = 1$). In this case, the distribution function of T is given by:

$$F_T(t) = \sum_{i=1}^{n} a_i \cdot F_{T_i}(t) \ \text{ for } t \in \mathbb{R},$$

and hence the mixed Poisson probabilities are given by

$$\alpha_T(k,q) = \sum_{i=1}^{n} a_i \cdot \int_0^{\infty} \gamma(k, q \cdot t)\, dF_{T_i}(t) = \sum_{i=1}^{n} a_i \cdot \alpha_{T_i}(k,q)\,. \tag{3}$$

Therefore, the mixed Poisson probabilities for a mixture are a linear combination of the mixed Poisson probabilities of the random variables involved in the mixture, and the coefficients of the linear combination are the weights of the associated random times.

For a given precision ε, the following algorithm may be used to compute the coefficients $\alpha_T(j,q)$, $j = 0, 1, \ldots, R(\varepsilon)$, such that $\sum_{j=R(\varepsilon)+1}^{\infty} \alpha_T(j,q) < \varepsilon$.

```
sum := 0
k := −1
while sum ≤ 1 − ε do
    k := k + 1
    c[k] := 0
    for i = 1 to n do
        c[k] := c[k] + a_i·α_{T_i}(k, q)
    endfor
    sum := sum + c[k]
endwhile
Output : R(ε) = k and α_T(j, q) = c[j], j = 0, 1, ... , k
```

4.2 Random times with finite discrete distribution

Suppose that T is a (finite) discrete random variable taking nonnegative values $t_1, t_2, \ldots, t_n$ with probabilities $p_1, p_2, \ldots, p_n$, then its distribution function is given by:

$$F_T(t) = \sum \{p_i \,|\, 1 \leq i \leq n \ \wedge \ t_i \leq t\} \ \text{ for } t \in \mathbb{R}.$$

Hence, in this case the mixed Poisson probabilities are given by

$$\alpha_T(k,q) = \int_0^\infty \gamma(k, q{\cdot}t)\, dF_T(t) = \sum_{i=1}^{n} p_i{\cdot}\gamma(k, q{\cdot}t_i) \ \text{ for } k = 0, 1, 2, \ldots \tag{4}$$

Therefore, if T is a discrete random variable, the amount of time needed to compute a mixed Poisson probability is of the same order as the time needed to compute a single Poisson probability.

If T is almost surely constant and equal to t, $\mathbf{P}(T = t) = 1$, then we have

$$\alpha_T(k,q) = \int_0^\infty \gamma(k, qu)\, dF_T(u) = \gamma(k, q{\cdot}t) \ \text{ for } k = 0, 1, 2, \ldots$$

Thus, we obtain the Poisson probabilities for fixed time t. Notice also that (4) is a consequence of the previous equation and (3), as a discrete random variable is also a mixture of almost surely constant random variables (the values that these random variables assume with probability one are the values to which the given discrete random variable assigns positive probabilities).

Now to compute the mixed Poisson probabilities for discrete distributions we may use the algorithm presented in Section 4.1 by replacing $\alpha_{T_i}(k,q)$ and a_i by $\gamma(k, q.t_i)$ and p_i respectively. Note that the Fox-Glynn algorithm [7] can be used to avoid overflow when computing the Poisson probabilities $\gamma(k, q{\cdot}t_i)$ for large values of $q{\cdot}t_i$.

4.3 Random times with gamma distribution

Suppose that T has a gamma distribution with (positive) parameters r and λ, $T \sim \text{Gamma}(r, \lambda)$, then T has probability density function:

$$f_T(t) = \frac{\lambda{\cdot}(\lambda{\cdot}t)^{r-1}{\cdot}e^{-\lambda \cdot t}}{\Gamma(r)} \ \text{ for } t > 0$$

where Γ is the gamma function, that is, for any $r > 0$:

$$\Gamma(r) = \int_0^\infty e^{-t}{\cdot}t^{r-1}\, dt\,.$$

Using the fact that $\Gamma(j+1) = j!$ for all $j \in \mathbb{N}$ we have:

$$
\begin{aligned}
\alpha_T(k,q) &= \int_0^{\infty} e^{-q\cdot t}\,\frac{(q\cdot t)^k}{\Gamma(k+1)}\cdot\frac{\lambda\cdot(\lambda\cdot t)^{r-1}\cdot e^{-\lambda\cdot t}}{\Gamma(r)}\,dt \\
&= \frac{\lambda^r\cdot q^k}{\Gamma(k+1)\cdot\Gamma(r)}\int_0^{\infty} e^{-(q+\lambda)\cdot t}\cdot t^{(k+r)-1}\,dt && \text{rearranging} \\
&= \frac{\lambda^r\cdot q^k}{\Gamma(k+1)\cdot\Gamma(r)}\int_0^{\infty} e^{-u}\cdot u^{(k+r)-1}\cdot(q+\lambda)^{-(k+r)}\,du && \text{setting } u = \frac{t}{q+\lambda} \\
&= \frac{1}{\Gamma(k+1)\cdot\Gamma(r)}\left(\frac{q}{\lambda+q}\right)^k\left(\frac{\lambda}{\lambda+q}\right)^r\int_0^{\infty} e^{-u}\cdot u^{(k+r)-1}\,du && \text{rearranging} \\
&= \frac{\Gamma(r+k)}{\Gamma(k+1)\cdot\Gamma(r)}\left(\frac{q}{\lambda+q}\right)^k\left(\frac{\lambda}{\lambda+q}\right)^r && \text{by definition of } \Gamma \\
&= p_{\mathrm{NegBin}\left(r,\frac{\lambda}{\lambda+q}\right)}(k)
\end{aligned}
$$

where $p_{\mathrm{NegBin}(r,\lambda/(\lambda+q))}(k)$ is the k-th probability of the negative binomial distribution with parameters r and $\lambda/(\lambda+q)$. Thus, if T has a gamma distribution, the mixed Poisson probabilities are negative binomial probabilities. Therefore, the coefficients $\alpha_T(k,q)$ may be computed recursively by the following scheme, for $k \in \mathbb{N}$:

$$
\alpha_T(0,q) = \left(\frac{\lambda}{\lambda+q}\right)^r \quad \text{and} \quad \alpha_T(k+1,q) = \left(\frac{k+r}{k+1}\right)\cdot\left(\frac{q}{\lambda+q}\right)\cdot\alpha_T(k,q)\,.
$$

Note that this recursion is an instance of the Panjer recursion [18]. Furthermore, for sufficiently large k, the coefficients $\alpha_T(k,q)$ exhibit an exponential decay towards zero. For a given precision ε, the following algorithm may be used to compute the coefficients $\alpha_T(k,q)$ recursively.

```
// Compute α_T(j,q) for j = 0,1,...,R(ε) if T ~ Gamma(r,λ)
k := 0
c[k] := (λ/(λ+q))^r
sum := c[k]
while sum ≤ 1 − ε do
    k := k + 1
    c[k] := (k+r−1)/k · q/(λ+q) · c[k − 1]
    sum := sum + c[k]
endwhile
Output : R(ε) := k and α_T(j,q) := c[j], j = 0,1,...,k
```

4.4 Random times with Erlang or exponential distribution

In computer network systems the most used particular case of the gamma distribution is the Erlang distribution, which corresponds to the case where the parameter r is integer. In that case it is common to call r the number of phases of the Erlang distribution and λ the rate. The popular and intensively used exponential distribution with rate λ corresponds to the particular case of $r = 1$. We also note that the Erlang(r, λ) is distributed as the sum of r independent exponential random variables with rate λ, so that these exponential random variables may be seen as the r phases of the Erlang distribution.

Although the Erlang and exponential distributions are special cases of the gamma family of distributions, they deserve special treatment due to their importance for applications. Moreover, these distributions lead to a clearer interpretation of the results obtained for the mixed Poisson probabilities. In fact these results, which we will comment on briefly, are very well known results in probability and statistics that appear recurrently in applications.

If T has an exponential distribution with rate λ, the mixed Poisson probabilities are geometric probabilities with parameter (success probability) $\lambda/(\lambda+q)$:

$$\alpha_T(k,q) = \left(\frac{q}{\lambda+q}\right)^k \cdot \left(\frac{\lambda}{\lambda+q}\right) \quad \text{for } k = 0, 1, 2, \ldots$$

The success probability represents the probability that the random time T is smaller than the first arrival epoch in the uniformising Poisson process with rate q. Due to the memoryless property of the exponential, the number of arrivals in the uniformising Poisson process with rate q that occur before time T corresponds to the number of trials observed before the occurrence of the first success and has a geometric distribution with parameter $\lambda/(\lambda+q)$.

Note that $\sum_{k=R+1}^{\infty} \alpha_T(k,q) = (q/(\lambda+q))^{k+1}$ for all $R \in \mathbb{N}$, and hence as a consequence for all $\varepsilon \in (0,1)$:

$$\sum_{k=R+1}^{\infty} \alpha_T(k,q) \le \varepsilon \quad \text{if and only if} \quad R \ge \left\lceil \frac{\ln(\varepsilon)}{\ln(q/(\lambda+q))} \right\rceil - 1\,.$$

That is, to achieve precision ε in the computation of (2) we need to evaluate order $\ln(\varepsilon)$ mixed Poisson probabilities.

If T has Erlang distribution with parameters r and λ, then the coefficients $\alpha_T(k,q)$ are the probabilities associated to a negative binomial random variable with parameters r and $\lambda/(\lambda+q)$, and hence for $k \in \mathbb{N}$:

$$\alpha_T(k,q) = \binom{r+k-1}{k} \cdot \left(\frac{q}{\lambda+q}\right)^k \cdot \left(\frac{\lambda}{\lambda+q}\right)^r .$$

This is a natural result as the sum of r independent random variables with geometric distribution with parameter $\lambda/(\lambda+q)$ has a negative binomial distribution with parameters r and $\lambda/(\lambda+q)$, which may be interpreted as the number of trials that are observed before the observation of the r-th success. If, again, a success is seen as an arrival in a Poisson process with rate λ and a failure as an arrival in the independent uniformising Poisson process with rate q, the value of the negative binomial random variable corresponds to the number of arrivals in the uniformising Poisson process with rate q that occur prior to the r-th arrival in the Poisson process with rate λ.

4.5 Random times with uniform distribution

If T has uniform distribution on $[L_T, R_T]$, then the coefficients $\alpha_T(k,q)$ are given by

$$\alpha_T(k,q) = \int_{L_T}^{R_T} \frac{1}{R_T - L_T} \cdot e^{-q\cdot t} \frac{(q\cdot t)^k}{k!}\, dt \quad \text{for } k = 0,1,\ldots$$

It follows that $\alpha_T(0,q) = \frac{e^{-q\cdot L_t} - e^{-q\cdot R_t}}{q\cdot(R_T - L_T)}$ and for $k = 0,1,2,\ldots$

$$\alpha_T(k+1,q) = \alpha_T(k,q) + \frac{1}{q\cdot(R_T - L_T)} \cdot \left[e^{-q\cdot L_t} \frac{(q\cdot L_T)^{k+1}}{(k+1)!} - e^{-q\cdot R_t} \frac{(q\cdot R_T)^{k+1}}{(k+1)!} \right]$$

which provides a recursive scheme of computing the coefficients $\alpha_T(k,q)$. Moreover, using induction, we conclude that for $k = 0,1,2,\ldots$:

$$\begin{aligned}
\alpha_T(k,q) &= \frac{1}{q\cdot(R_t - L_T)} \cdot \left[\sum_{j=0}^{k} e^{-q\cdot L_t} \cdot \frac{(q\cdot L_T)^j}{j!} - \sum_{j=0}^{k} e^{-q\cdot R_t} \cdot \frac{(q\cdot R_T)^j}{j!} \right] \\
&= \frac{1}{q\cdot(R_t - L_T)} \cdot \left[\sum_{j=0}^{k} \gamma(k, q\cdot L_T) - \sum_{j=0}^{k} \gamma(j, q\cdot R_T) \right] \\
&= \frac{1}{q\cdot(R_T - L_T)} \cdot \left[F_{\text{Poisson}(q\cdot L_T)}(k) - F_{\text{Poisson}(q\cdot R_T)}(k) \right]
\end{aligned}$$

where $F_{\text{Poisson}(\lambda)}(\cdot)$ is the distribution function of a Poisson random variable with parameter λ. Note that if, in particular, $L_T = 0$, then

$$\alpha_T(k,q) = \frac{1 - F_{\text{Poisson}(q\cdot R_T)}(k)}{q\cdot R_T} = \frac{e^{-q\cdot R_t}}{q\cdot R_t} \cdot \sum_{j=k+1}^{\infty} \frac{(q\cdot R_T)^j}{j!}\,.$$

For a given precision ε, the following algorithm may be used to compute the coefficients $\alpha_T(k,q)$ recursively.

```
// Compute α_T(j,q) for j = 0,1,...,R(ε) if T ~ Uniform([L_T, R_T])
k := 0
d := e^{-q·L_T} / (q·(R_T − L_T))
e := e^{-q·R_T} / (q·(R_T − L_T))
c[k] := d − e
sum := c[k]
while sum ≤ 1 − ε do
    k := k + 1
    d := d·(q·L_T / k)
    e := e·(q·R_T / k)
    c[k] := c[k − 1] + d − e
    sum := sum + c[k]
endwhile
Output : R(ε) := k and α_T(j,q) := c[j], j = 0,1,...,k
```

4.6 Random times with Pareto distribution

The Pareto distribution has recently gained high importance in telecommunications as it has been shown, e.g., that it fits the distribution of times in-between the start of Internet sessions [16] and the size of files available in the Web [6]. Suppose that T has a Pareto distribution with (positive) parameters κ and β, $T \sim \text{Pareto}(\kappa, \beta)$; i.e., T has probability density function:

$$f_T(t) = \begin{cases} 0 & \text{if } x \le \kappa \\ \frac{\beta \cdot \kappa^\beta}{t^{\beta+1}} & \text{if } x > \kappa \,. \end{cases}$$

This is a heavy-tailed distribution that has infinite variance for $\beta \le 2$ and infinite expected value for $\beta \le 1$. The mixed Poisson probabilities in this case are:

$$\begin{aligned}
\alpha_T(k,q) &= \int_\kappa^\infty e^{-q \cdot t} \cdot \frac{(q \cdot t)^k}{k!} \cdot \frac{\beta \cdot \kappa^\beta}{t^{\beta+1}}\, dt \\
&= \frac{\beta \cdot \kappa^\beta \cdot q^{\beta+1}}{k!} \cdot \int_\kappa^\infty e^{-q \cdot t} \cdot (q \cdot t)^{k-\beta-1}\, dt && \text{rearranging} \\
&= \frac{\beta \cdot \kappa^\beta \cdot q^\beta}{k!} \cdot \int_{q\kappa}^\infty e^{-y} \cdot y^{k-\beta-1}\, dy && \text{letting } y = q \cdot t \\
&= \beta \cdot (q\kappa)^\beta \cdot \frac{\Gamma(k-\beta, q\kappa)}{k!}
\end{aligned}$$

for $k = 0, 1, 2, \ldots$, where $\Gamma(\cdot, \cdot)$ is the incomplete gamma function. Now, for any $x > 0$ integrating by parts we have

$$\Gamma(a+1, x) = \int_x^\infty e^{-y} \cdot y^a\, dy = e^{-x} \cdot x^a + a \cdot \Gamma(a, x) \tag{5}$$

and, as a consequence for $a \neq 0$:

$$\Gamma(a, x) = \frac{1}{a} \left[\Gamma(a+1, x) - e^{-x} \cdot x^a \right] . \tag{6}$$

Applying these results we get that the following upward and backward recursions for $\alpha_T(k,q)$, for $k = 0, 1, 2, \ldots$:

$$\begin{aligned}
\alpha_T(k+1,q) &= \beta \cdot (q\kappa)^\beta \cdot \frac{\Gamma(k+1-\beta, q\kappa)}{(k+1)!} \\
&= \beta \cdot (q\kappa)^\beta \cdot \left[\frac{k-\beta}{k+1} \cdot \frac{\Gamma(k-\beta, q\kappa)}{k!} + e^{-q\kappa} \cdot \frac{(q\kappa)^{k-\beta}}{(k+1)!} \right] && \text{by (5)} \\
&= \frac{1}{k+1} \cdot \left[(k-\beta) \cdot \alpha_T(k,q) + \beta \cdot e^{-q\kappa} \cdot \frac{(q\kappa)^k}{k!} \right] && \text{rearranging.}
\end{aligned}$$

Similarly, using (6) for any $k \neq \beta$:

$$\alpha_T(k,q) = \frac{1}{k-\beta} \cdot \left[(k+1) \cdot \alpha_T(k+1,q) - \beta \cdot e^{-q\kappa} \cdot \frac{(q\kappa)^k}{k!} \right] .$$

For a given precision ε, the following algorithm may be used to compute the coefficients $\alpha_T(k,q)$ recursively resorting to a single evaluation of the incomplete gamma function.

// *Compute* $\alpha_T(j,q)$ *for* $j = 0, 1, \ldots, R(\varepsilon)$ *if* $T \sim \text{Pareto}(\kappa, \beta)$
$k_{aux} := \lceil \beta \rceil$
$c[k_{aux}] := \beta \cdot (q\kappa)^{\beta} \cdot \frac{\Gamma(k_{aux}-\beta, q\kappa)}{k_{aux}!}$
$d_{aux} := \beta \cdot e^{-q\kappa} \cdot \frac{(q\kappa)^{k_{aux}}}{k_{aux}!}$
$sum := c[k_{aux}]$
// *Backward loop*
$k := k_{aux}$
$d := d_{aux}$
`while` $k > 0$ `do`
    $k := k - 1$
    $d := d \cdot \frac{k+1}{q\kappa}$
    $c[k] := [(k+1) \cdot c[k+1] - d]/(k - \beta)$
    $sum := sum + c[k]$
`endwhile`
// *Forward loop*
$k := k_{aux}$
$d := d_{aux}$
`while` $sum \leq 1 - \varepsilon$ `do`
    $k := k + 1$
    $c[k] := [(k - 1 - \beta) \cdot c[k-1] + d]/k$
    $d := d \cdot \frac{q\kappa}{k}$
    $sum := sum + c[k]$
`endwhile`
`Output` : $R(\varepsilon) := k$ `and` $\alpha_T(j,q) := c[j]$, $j = 0, 1, \ldots, k$

This algorithm is stable for any values of the parameters, including the situations where the variance, and even the expected value, is infinite. The computing time increases linearly with the number of mixed Poisson probabilities computed which, for a given precision, grows considerably as the shape parameter β decreases. As reported in [8], $R(\varepsilon)$ exhibits a fast increase when ε decreases, contrarily to the other distributions considered. Moreover, for a given precision ε, $R(\varepsilon)$ takes values several orders of magnitude larger than for non heavy-tailed distributions.

5 Example

We consider the $GI/M/a/a+c$ queueing system. In this system the arrival process is a renewal process with a fixed (but otherwise general) inter-arrival time distribution, the service times are exponential (with rate λ), there are a identical servers, and there are c positions for waiting. We can model-check transient properties of this queueing system involving the time at which a new customer arrives by:

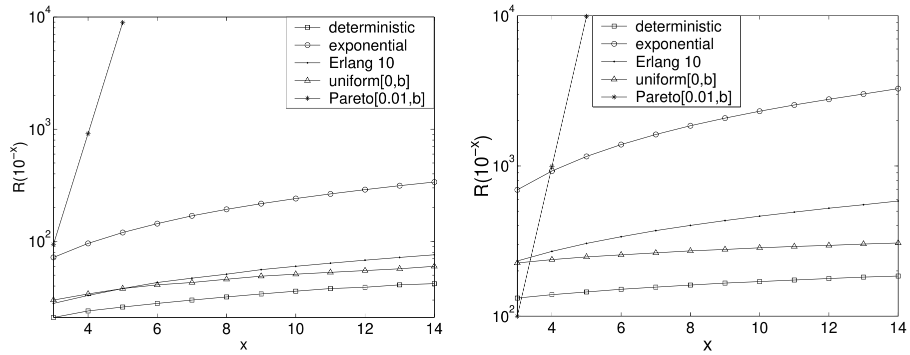

Fig. 2. Change in $R(\varepsilon)$ as the accuracy ε changes

– constructing a model of the queueing system in which transitions corresponding to new arrivals are removed;
– verifying this restricted model against formulae which use the inter-arrival time distribution as their random time-bound.

Note that, in this restricted model, all transitions have an exponential delay, that is, it is a CTMC, and since the service time distribution is independent of the arrival time distribution, the inter-arrival time distribution is independent of this model.

Formally, the CTMC is given by $\mathcal{C} = (S, \mathbf{R}, L)$ with $S = \{s_0, s_1, \ldots, s_{a+c}\}$ where s_k denotes the state in which there are k customers in the queue, and the rate matrix is given by:

$$\mathbf{R}(s_i, s_j) = \begin{cases} a{\cdot}\lambda & \text{if } a \leq i \leq a+c \text{ and } j = i-1 \\ i{\cdot}\lambda & \text{if } 0 \leq i < a \text{ and } j = i-1 \\ 0 & \text{otherwise} \end{cases}$$

where λ is the rate of service of each server. Now, if T represents the inter-arrival time distribution, then the satisfaction of $\mathcal{P}_{\bowtie p}(\Phi\, \mathcal{U}^{\leq T}\, \Psi)$ in $\mathcal{C}$ corresponds to the following property holding in the $GI/M/a/a+c$ queueing system: with probability $\bowtie p$, when the next arrival occurs, a state satisfying Ψ will be reached such that all preceding states satisfy Φ.

We fix $a = 10$ and $c = 5$ and calculate, for each state s of the CTMC $\mathcal{C}$, the value of $Prob^{\mathcal{C}}(s, \texttt{true}\, \mathcal{U}^{\leq T}\, \Psi)$ where Ψ is true in the states in which there are at most $k(= 3)$ customers in the queue (including those being served). In other words, we calculate for each state of the $GI/M/a/a+c$ queueing system the probability of there being at most k customers in the queue when the next arrival occurs. We consider five different distributions for T: deterministic (which represents the standard time-bounded until formula), exponential, Erlang (with 10 phases), uniform and Pareto.

The results were obtained with a prototype implementation in MATLAB with an accuracy $\varepsilon = 10^{-8}$. We note that the mixed Poisson probabilities for the Pareto distribution were slow to converge; however, in this case, we use the fact

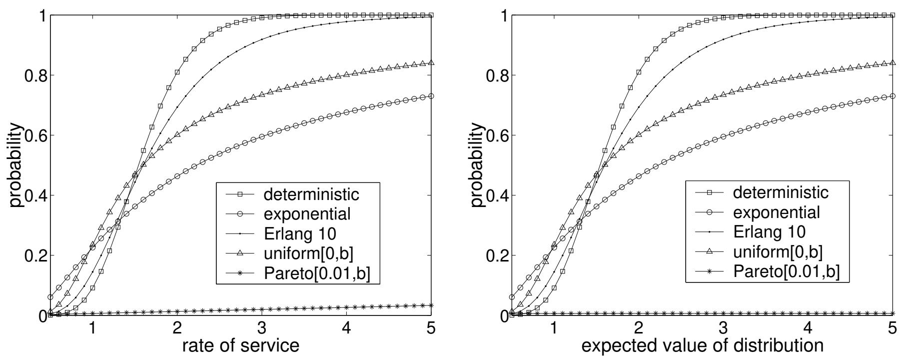

Fig. 3. Value of $Prob^{\mathcal{C}}(s, \mathtt{true}\, \mathcal{U}^{\leq T}\, \Psi)$ when the queue is full

that we can stop iterating when the DTMC $unif(\mathcal{C})[\neg\Phi \vee \Psi]$ reaches steady state. To illustrate this point, in Figure 2 we have plotted the values of $R(\varepsilon)$ for different values of ε. Note that we have used a logarithmic scale on the y-axis to allow the results for the Pareto to be plotted in the same graph as the remaining distributions. The graph on the left corresponds to the case when T has an expected value of 1, while the one on the right to the case when the expected value of T is 10.

As can be seen from Figure 2, the case of the Pareto distribution quickly becomes unmanageable as we increase the accuracy. This result follows from the fact that the Pareto distribution is *heavy tailed*. In fact, the differences in the values of $R(\varepsilon)$ for the different distributions correspond to the difference in their "tails": the heavier the tail the larger the value of $R(\varepsilon)$. For example, the deterministic distribution is zero for all values greater than the expected value, whereas the uniform distribution is zero for any value greater than two times the expected value, and the exponential distribution has a heavier tail than an Erlang (with more than one phase). T

In Figure 3 we plot the value of $Prob^{\mathcal{C}}(s, \mathtt{true}\, \mathcal{U}^{\leq T}\, \Psi)$ when s is the state in which the queue is full. The left graph corresponds to the case when we vary the service rate and fix the expected value of T at 1, while in the graph on the right the service rate is fixed at 1 and we vary the expected value of T. Similarly, in Figure 4 we plot the values of $Prob^{\mathcal{C}}(s, \mathtt{true}\, \mathcal{U}^{\leq T}\, \Psi)$ when s is the state in which there are $k+1$ customers in the queue.

The graphs in Figure 3 and Figure 4 demonstrate the expected result: as either the expected time of T increases (the expected time between consecutive arrivals increases) or the rate of service increases (the expected duration of a service decreases) the probability of there being at most k customers in the queue at time T increases. Furthermore, the probability is much lower for the state where the queue is full (Figure 3) than for the state where there are only $k+1$ customers in the queue (Figure 4). This follows from the fact that, from the state in which the queue is full, to reach a Ψ state $s+c-k$ customers need to be served, as opposed to only 1 customer from the state where there are $k+1$ customers in the queue.

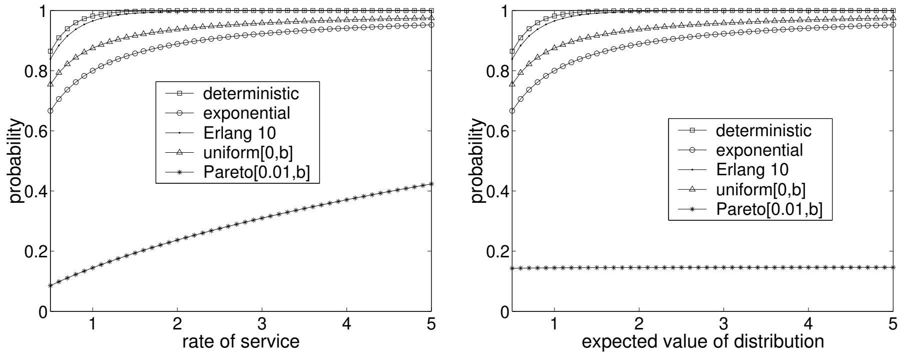

Fig. 4. Value of $Prob^{\mathcal{C}}(s, \texttt{true}\ \mathcal{U}^{\leq T}\ \Psi)$ when there are $k+1$ customer in the queue

The first point to notice about the results is that approximating general distributions with exponential distributions leads to inaccurate results. For example, compare the results when T is exponential (customers arrive with an exponential distribution) to when T has a Pareto distribution (customers arrive with a Pareto distribution). It is also apparent that the computed probabilities are much smaller for the Pareto distribution than for the remaining distributions, whose survival functions exhibit exponential decay to zero. This holds because, although the Pareto distribution is heavy-tailed, it assumes very small values with much higher probability than the other distributions. Significant differences may also arise for light-tailed distributions such as the exponential and Erlang distributions, as illustrated in Figure 3. Finally, note the close similarity between the cases when T has a deterministic and Erlang distribution; this is also to be expected as the Erlang distribution is often used as a continuous approximation of a (discrete) deterministic distribution.

6 Conclusions

This paper presents an extension of CSL with until formulae where the time bound is given as a general nonnegative random variable. This extension allows one, in certain cases, to calculate transient measures of systems which include general distributions. We demonstrate that model checking for such formulae can be efficiently carried out by first precomputing a vector of mixed Poisson probabilities and then using a straightforward adaptation of the algorithm for ordinary time-bounded until.

Currently, we have only considered a prototype implementation of these algorithms in MATLAB. In future we plan to implement these algorithms in the probabilistic symbolic model checker PRISM [14, 19] in order to tackle the verification of more complex models.

Additionally, we would like to work on generalising this approach to other important families of distributions; apply analytic methods to finding upper bounds for $R(\varepsilon)$; consider random expected time; and extend our approach to express random time intervals rather than simply the time bound T.

References

1. A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Verifying continuous time Markov chains. In *Proc. CAV'96*, volume 1102 of *LNCS*, pages 269–276. Springer, 1996.
2. A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model checking continuous time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.
3. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In *Proc. CAV'00*, volume 1855 of *LNCS*, pages 358–372, 2000.
4. C. Baier, J.-P. Katoen, and H. Hermanns. Approximative symbolic model checking of continuous-time Markov chains. In *Proc. CONCUR'99*, volume 1664 of *LNCS*, pages 146–162. Springer, 1999.
5. E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.
6. M. Crovella and A. Bestavros. Self-similarity in world wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.
7. B. Fox and P. Glynn. Computing Poisson probabilities. *Communications of the ACM*, 31(4):440–445, 1988.
8. R. German. *Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets*. John Wiley and Sons, 2000.
9. J. Grandell. *Mixed Poisson Processes*. Chapman & Hall, 1997.
10. D. Gross and D. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.
11. H. Hansson and B. Jonsson. A logic for reasoning about time and probability. *Formal Aspects of Computing*, 6:512–535, 1994.
12. A. Jensen. Markov chains as an aid in the study of Markov processes. *Skandinavisk Aktuarietidsskrift, Marts*, pages 87–91, 1953.
13. J.-P. Katoen, M. Kwiatkowska, G. Norman, and D. Parker. Faster and symbolic CTMC model checking. In *Proc. PAPM-PROBMIV'01*, volume 2165 of *LNCS*, pages 23–38. Springer, 2001.
14. M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: A hybrid approach. In *Proc. TACAS'02*, volume 2280 of *LNCS*, pages 52–66. Springer, 2002.
15. G. I. Lópes, H. Hermanns, and J.-P. Katoen. Beyond memoryless distributions. In *Proc PAPM-PROBMIV'01*, volume 2165 of *LNCS*, pages 57–70. Springer, 2001.
16. S. Molnár and I. Maricza, editors. *Source characterization in broadband networks*. COST 257 Mid-term seminar interim report on source characterization, 2000.
17. J. Muppala and K. Trivedi. *Queueing Systems, Queueing and Related Models*, chapter Numerical Transient Solution of Finite Markovian Queueing Systems, pages 262–284. Oxford University Press, 1992.
18. H. Panjer. Recursive evaluation of a family of compound distributions. *Astin Bulletin*, 12(1):22–26, 1982.
19. PRISM web page. http://www.cs.bham.ac.uk/ dxp/prism/.
20. W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton, 1994.