Model-Checking Algorithms for Continuous-Time Markov Chains

Christel Baier, Boudewijn Haverkort, *Senior Member*, *IEEE*,
Holger Hermanns, and Joost-Pieter Katoen, *Member*, *IEEE Computer Society*

Abstract—Continuous-time Markov chains (CTMCs) have been widely used to determine system performance and dependability characteristics. Their analysis most often concerns the computation of steady-state and transient-state probabilities. This paper introduces a branching temporal logic for expressing real-time probabilistic properties on CTMCs and presents approximate model checking algorithms for this logic. The logic, an extension of the continuous stochastic logic CSL of Aziz et al., contains a time-bounded until operator to express probabilistic timing properties over paths as well as an operator to express steady-state probabilities. We show that the model checking problem for this logic reduces to a system of linear equations (for unbounded until and the steady-state operator) and a Volterra integral equation system (for time-bounded until). We then show that the problem of model-checking time-bounded until properties can be reduced to the problem of computing transient state probabilities for CTMCs. This allows the verification of probabilistic timing properties by efficient techniques for transient analysis for CTMCs such as uniformization. Finally, we show that a variant of lumping equivalence (bisimulation), a well-known notion for aggregating CTMCs, preserves the validity of all formulas in the logic.

Index Terms—Continuous-time Markov chain, lumping, model checking, temporal logic, steady-state analysis, transient analysis, uniformization.

1 Introduction

Continuous-time Markov chains (CTMCs) [34], [46], [50], [53], [69] are an important class of stochastic processes that have been widely used in practice to determine system performance and dependability characteristics. To mention just a few practical applications, these models have been used to quantify the throughput of production lines, to determine the mean time between failure in safety-critical systems, and to identify bottlenecks in high-speed communication networks. Due to the rapidly increasing size and complexity of systems, obtaining such models in a direct way becomes more and more cumbersome and error-prone. To avoid the specification of CTMCs directly at the state level, high-level model specification techniques have been developed, most notably those based on queuing networks [26], stochastic Petri nets [2], stochastic activity networks [57], [61], and stochastic process algebras [41], [44]. With appropriate software tools supporting these specification methods, such as those provided by MACOM [52], SPNP [21], UltraSAN [68], or TIPPtool [42], it is relatively comfortable to specify performance and dependability models of which the underlying CTMCs have millions of states, cf. [69]. In combination with state-of-the art numerical means to compute state-based probabilities, a good workbench is available to construct and solve CTMC models of complex systems.

The design of performance and dependability *models* is usually complemented by a specification of the performance and dependability *measures* of interest, such as throughput, mean response time, and utilization. The measure of interest determines the kind of analysis that is to be carried out in order to compute the measure under study. Whereas the specification of performance and dependability models has become quite comfortable, the specification of performance measures of interest often has remained fairly cumbersome and is typically done in a rather informal, ad hoc manner. In particular, usually only simple state-based performance measures—such as *steady-state* and *transient-state probabilities*—can be defined and analyzed with relative ease. Steady-state probabilities refer to the system behavior in the "long run," whereas the transient-state probabilities consider the system at a fixed time instant t.

In contrast, in the area of formal methods, very powerful means have been developed to express temporal properties of systems, based on temporal logics. In this context, systems are specified as transition systems consisting of a finite set of states and a set of transitions that describe how the system evolves from one state to another. Branching-time logics such as CTL (Computation Tree Logic) [32] allow one to express state-based properties as well as properties over paths, i.e., state sequences through transition systems. Typical properties expressible in CTL are that along all (or some) paths a certain set of (goal) states can eventually be reached while visiting only states of a particular kind before reaching one of these goal-states.

- *C. Baier is with the Institut für Informatik I, University of Bonn, Römerstraße 164, D-53117 Bonn, Germany. E-mail: baier@cs.uni-bonn.de.*
- *B. Haverkort, H. Hermanns, and J.-P. Katoen are with the Department of Computer Science, University of Twente, PO Box 217, NL-7500 AE Enschede, The Netherlands. E-mail: {brh, hermanns, katoen}@cs.utwente.nl.*

Manuscript received 3 Apr. 2002; revised 18 Feb. 2003; accepted 24 Feb. 2003. Recommended for acceptance by E. Clarke.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 116221.

Similar capabilities would also be very useful for specifying performance and dependability measures over models such as CTMCs. Note that we can view a finite-state CTMC as a special kind of a transition system. The validity of CTL-formulas over finite-state automata can be established by fully automated techniques such as *model checking* [32], [65]; for an overview, see [25]. The basis of model checking CTL is a systematic, usually exhaustive, state-space exploration to check whether a property is satisfied in each state of the model, thereby using effective methods to combat the state-space explosion problem. Model checking has been successfully used to validate, among others, hardware and software systems, security protocols, and e-commerce systems. With appropriate tools such as SMV [22], SPIN [45], and Murφ [31], systems of several millions of states have been analyzed.

In this paper, we present the branching-time logic CSL (*Continuous Stochastic Logic*) that provides us ample means to specify state as well as path-based performance and dependability measures for CTMCs in a compact and unambiguous way. This logic is basically a probabilistic timed extension of CTL and is strongly based on the (equally named) logic by Aziz et al. [9] and on PCTL, a variant of CTL for discrete-time Markov chains (DTMCs) [38]. Besides the standard steady-state and transient measures, the logic allows for the specification of (constraints over) probabilistic measures over paths through CTMCs. For instance, the probability can be expressed as follows: Starting from a particular state, within t time units, a set of goal-states is reached, thereby avoiding or deliberately visiting particular intermediate states before. This is a useful feature for dependability analysis, as demonstrated in [40], and goes beyond the standard measures in performance and dependability analysis.

The model checking problem for CSL is known to be decidable [9] (for rational time bounds), but, to the best of our knowledge, no algorithms have been considered yet to verify CTMCs mechanically. In this paper, we investigate which numerical methods can be adapted to "model check" CSL-formulas over finite-state CTMCs. We show that next and until-formulas (without time bound) can be treated in a similar way as in the discrete-time probabilistic setting using matrix-vector multiplication and solving a system of linear equations [38]. Checking steady-state properties reduces to solving a system of linear equations combined with standard graph analysis methods, while checking until formulas with a time bound requires the solution of a (recursive) Volterra integral equation system. These characterizations provide the theoretical basis for model checking CSL over CTMCs in the same way as the fixed-point characterizations for CTL provide the basis for the model checking algorithms for CTL [23].

We show that model checking time-bounded until-formulas can be reduced to the problem of computing transient-state probabilities for CTMCs. In particular, our result states that, for a given CTMC $\mathcal{M}$ and state s in $\mathcal{M}$, the measure $Prob^{\mathcal{M}}(s, \varphi)$ for path-formula φ to hold when the system starts in state s can be calculated by means of a transient analysis of another CTMC $\mathcal{M}'$, which can easily be derived from $\mathcal{M}$ using φ. This allows us to adopt efficient and numerically stable techniques for performing transient analysis of CTMCs, like *uniformization* [36], [37], [47], for model checking time-bounded until-formulas. The reduction of the model checking problem for the time-bounded until-operator to the transient analysis of a CTMC has the advantage that—besides avoiding an awkward numerical integration of the Volterra equation system—it employs a *measure-driven* transformation of the CTMC.

In addition, we show that *lumping*—an equivalence notion on Markov chains to aggregate state spaces [18], [44] that can be viewed as a continuous variant of probabilistic bisimulation [55]—preserves the validity of all CSL-formulas. This allows us to switch from the original state space to the (possibly much smaller) quotient space under lumping prior to carrying out the model checking. Using this property, we indicate how the state space for checking probabilistic timing properties on the derived CTMC $\mathcal{M}'$ can be obtained. This result is in the same spirit as [17] where bisimulation is shown to agree with CTL and CTL* equivalence.

Summarizing, the main contributions of this paper are:

- the definition of a stochastic branching-time logic that facilitates the formal specification of state-based, path-based, and more complex performance measures;
- the characterization of the probability measure for time-bounded until formulas in terms of a Volterra integral equation system;
- the transformation and subsequent computation of probability measures for time-bounded until formulas by transient analysis;
- the preservation of the validity of CSL formulas under lumping.

This paper is based on the extended abstract [11] and the paper [12].

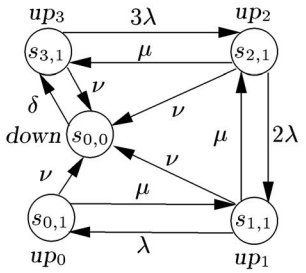
Organization of the paper. Section 2 introduces the basic concepts of CTMCs. Section 3 presents the logic CSL and provides fixed-point characterizations of CSL-formulas that form the basis for a model checking procedure. Section 4 presents the reduction of the model checking problem for time-bounded until to a transient analysis of CTMCs and discusses the use of uniformisation. Section 5 discusses lumping and the preservation of CSL-formulas. Section 6 presents efficiency considerations for model checking CSL, whereas Section 7 places our work in the context of related research. Finally, Section 8 concludes the paper.

2 Continuous-Time Markov Chains

This section recalls the basic concepts of continuous-time Markov chains (CTMCs) as originally developed by Markov [56] for finite state spaces and Kolmogorov [51] for denumerable and continuous state spaces. The presentation is focused on the concepts needed for the understanding of the rest of this paper; for a more elaborate treatment, we refer to [34], [46], [49], [53], [69].

2.1 Labeled CTMCs

To ease the definition of the semantics of the logic CSL, we slightly depart from the standard notations for CTMCs and



$$\mathbf{R} = \begin{pmatrix} 0 & 3\lambda & 0 & 0 & \nu \\ \mu & 0 & 2\lambda & 0 & \nu \\ 0 & \mu & 0 & \lambda & \nu \\ 0 & 0 & \mu & 0 & \nu \\ \delta & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \underline{E} = \begin{pmatrix} 3\lambda+\nu \\ 2\lambda+\mu+\nu \\ \lambda+\mu+\nu \\ \mu+\nu \\ \delta \end{pmatrix}$$

Fig. 1. The CTMC modeling a triple modular redundant system.

consider a CTMC as an ordinary finite transition system (Kripke structure) where the edges are equipped with probabilistic timing information. Let AP be a fixed, finite set of atomic propositions.

Definition 1. *A (labeled) CTMC $\mathcal{M}$ is a tuple $(S, \mathbf{R}, L)$ with S as a finite set of* states, *$\mathbf{R} : S \times S \to \mathbb{R}_{\geq 0}$ as the* rate matrix, *and $L : S \to 2^{AP}$ as the* labeling *function.*

Intuitively, function L assigns to each state $s \in S$ the set $L(s)$ of atomic propositions $a \in AP$ that are valid in s. It should be noted that Definition 1 does not require $\mathbf{R}(s,s) = -\sum_{s' \neq s} \mathbf{R}(s,s')$, as is usual for CTMCs. In the traditional interpretation, at the end of a stay in state s, the system will move to a different state. According to Definition 1, self-loops at state s are possible and are modeled by having $\mathbf{R}(s,s) > 0$. We thus allow the system to occupy the same state before and after taking a transition. The inclusion of self-loops neither alters the transient nor the steady-state behavior of the CTMC, but allows the usual interpretation of linear-time temporal operators like next step and until. This will be exploited when we address the semantics of the logic CSL in Section 3.2. CTMCs are also treated in this way in, among others, the textbook [64].

A state s is called *absorbing* iff $\mathbf{R}(s,s') = 0$ for all states s'. Whenever appropriate, we assume that, for any state s, AP contains an atomic proposition at_s which is characteristic for s, i.e., $at_s \in L(s)$ and $at_s \notin L(s')$ for any $s' \neq s$. For $S' \subseteq S$, the atomic proposition $at_{S'}$ stands for $\bigvee_{s \in S'} at_s$.

Intuitively, $\mathbf{R}(s,s') > 0$ iff there is a transition from s to s'. Furthermore, $1 - e^{-\mathbf{R}(s,s') \cdot t}$ is the probability that the transition $s \to s'$ can be triggered within t time units. Thus, the delay of transition $s \to s'$ is governed by the exponential distribution with rate $\mathbf{R}(s,s')$. If $\mathbf{R}(s,s') > 0$ for more than one state s', a competition between the transitions originating in s exists, known as the *race condition*. The probability to move from a nonabsorbing state s to a particular state s' within t time units, i.e., the transition $s \to s'$ wins the race, is given by:

$$\mathbf{P}(s,s',t) = \frac{\mathbf{R}(s,s')}{E(s)} \cdot \left(1 - e^{-E(s) \cdot t}\right),$$

where $E(s) = \sum_{s' \in S} \mathbf{R}(s,s')$ denotes the *total rate* at which any transition outgoing from state s is taken. More precisely, $E(s)$ specifies that the probability of taking a transition outgoing from state s within t time units is $1 - e^{-E(s) \cdot t}$, due to the fact that the minimum of two exponentially distributed random variables is an exponentially distributed random variable with as rate the sum of their rates. Consequently, the probability of moving from a nonabsorbing state s to s' by a single transition, denoted $\mathbf{P}(s,s')$, is determined by the probability that the delay of going from s to s' finishes before the delays of other outgoing edges from s; formally, $\mathbf{P}(s,s') = \mathbf{R}(s,s')/E(s)$. For an absorbing state s, the total rate $E(s)$ is 0. In that case, we have $\mathbf{P}(s,s') = 0$ for any state s'. The matrix $\mathbf{P}$ is usually known as the transition matrix of the embedded discrete-time Markov chain of $\mathcal{M}$ (except that usually $\mathbf{P}(s,s) = 1$ for absorbing s).

Definition 2. *An* initial distribution *on $\mathcal{M} = (S, \mathbf{R}, L)$ is a function $\alpha : S \to [0,1]$ such that $\sum_{s \in S} \alpha(s) = 1$.*

In case there is a unique initial state s, the initial distribution is denoted α_s^1, where $\alpha_s^1(s) = 1$ and $\alpha_s^1(s') = 0$ for any $s' \neq s$.

Example 1. As a running example, we address a *triple modular redundant system* (TMR) taken from [39], a fault-tolerant computer system consisting of three processors and a single (majority) voter. We model this system as a CTMC where state $s_{i,j}$ models that i $(0 \leq i \leq 3)$ processors and j $(0 \leq j \leq 1)$ voters are operational. As atomic propositions, we use $AP = \{up_i \mid 0 \leq i < 4\} \cup \{down\}$. The processors generate results and the voter decides upon the correct value by taking a majority vote. Initially, all components are functioning correctly, i.e., $\alpha = \alpha_{s_{3,1}}^1$. The failure rate of a single processor is λ and of the voter ν failures per hour (fph). The expected repair time of a processor is $1/\mu$ and of the voter $1/\delta$ hours. It is assumed that one component can be repaired at a time. The system is operational if at least two processors and the voter are functioning correctly. If the voter fails, the entire system is assumed to have failed and, after a repair (with rate δ), the system is assumed to start "as good as new." The details of the CTMC modeling this system are shown in Fig. 1 (with a clockwise ordering of states for the matrix/vector-representation, starting with $s_{3,1}$).

States are represented by circles and there is an edge between state s and state s' if and only if $\mathbf{R}(s,s') > 0$. The labeling is defined by $L(s_{i,1}) = \{up_i\}$ for $0 \leq i < 4$ and $L(s_{0,0}) = \{down\}$ and is indicated near the states (set braces are omitted for singletons). For the transition probabilities, we have, e.g., $\mathbf{P}(s_{2,1}, s_{3,1}) = \mu/(2\lambda+\mu+\nu)$ and $\mathbf{P}(s_{0,1}, s_{0,0}) = \nu/(\mu+\nu)$.

2.2 Paths in CTMCs

Definition 3. *Let $\mathcal{M} = (S, \mathbf{R}, L)$ be a CTMC. An infinite* path σ *is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \to t_2 \ldots$ with, for $i \in \mathbb{N}$, $s_i \in S$, and $t_i \in \mathbb{R}_{>0}$ such that $\mathbf{R}(s_i, s_{i+1}) > 0$ for all i. A finite path σ is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots s_{l-1} \xrightarrow{t_{l-1}} s_l$ such that s_l is absorbing, and $\mathbf{R}(s_i, s_{i+1}) > 0$ for all $i < l$.*[1]

For infinite path σ and $i \in \mathbb{N}$, let $\sigma[i] = s_i$, the $(i+1)$st state of σ, and $\delta(\sigma, i) = t_i$, the time spent in s_i. For $t \in \mathbb{R}_{\geq 0}$ and i the smallest index with $t \leq \sum_{j=0}^{i} t_j$, let $\sigma@t = \sigma[i]$, the state in σ occupied at time t. For finite σ that ends in s_l, $\sigma[i]$ and $\delta(\sigma, i)$ are only defined for $i \leq l$; they are defined for $i < l$ in the above way and $\delta(\sigma, l) = \infty$. For $t > \sum_{j=0}^{l-1} t_j$, let $\sigma@t = s_l$; otherwise, $\sigma@t$ is as above. For instance, for finite path

$$\sigma = s_0 \xrightarrow{1.7} s_1 \xrightarrow{\sqrt{2}} s_2 \xrightarrow{4} s_3,$$

we have $\delta(\sigma, 0) = 1.7$ and $\delta(\sigma, 1) = \sqrt{2}$, $\sigma[0] = s_0 = \sigma@0.758$, $\sigma[1] = s_1 = \sigma@1.8$, $\sigma[2] = s_2 = \sigma@3.4$, $\sigma[3] = s_3 = \sigma@57$, and $\sigma@t = s_3$ for all $t > 5.7 + \sqrt{2}$. Let $Path^{\mathcal{M}}$ denote the set of (finite and infinite) paths in the CTMC $\mathcal{M}$, and $Path^{\mathcal{M}}(s)$ the set of paths in $\mathcal{M}$ that start in s. The superscript $\mathcal{M}$ is omitted whenever convenient.

2.3 Borel Space

Our definition of a Borel space on paths through CTMCs follows [71], [38]. An initial distribution α yields a probability measure $\Pr_\alpha$ on paths as follows: Let $s_0, \ldots, s_k \in S$ with $\mathbf{R}(s_i, s_{i+1}) > 0$ $(0 \leq i < k)$ and $I_0, \ldots, I_{k-1}$ nonempty intervals in $\mathbb{R}_{\geq 0}$. Then, $C(s_0, I_0, \ldots, I_{k-1}, s_k)$ denotes the *cylinder set* consisting of all paths $\sigma \in Path(s_0)$ such that $\sigma[i] = s_i$ $(i \leq k)$, and $\delta(\sigma, i) \in I_i$ $(i < k)$. Let $\mathcal{F}(Path)$ be the smallest σ algebra on $Path$ which contains all sets $C(s, I_0, \ldots, I_{k-1}, s_k)$, where $s_0, \ldots, s_k$ ranges over all state-sequences with $s = s_0$, $\mathbf{R}(s_i, s_{i+1}) > 0$ $(0 \leq i < k)$ and $I_0, \ldots, I_{k-1}$ ranges over all sequences of nonempty intervals in $\mathbb{R}_{\geq 0}$. The probability measure $\Pr_\alpha$ on $\mathcal{F}(Path)$ is the unique measure defined by induction on k by $\Pr_\alpha(C(s_0)) = \alpha(s_0)$ and, for $k \geq 0$:

$$\Pr_\alpha(C(s_0, I_0, \ldots, s_k, I', s')) = \Pr_\alpha(C(s_0, I_0, \ldots, s_k)) \cdot \mathbf{P}(s_k, s') \cdot \left(e^{-E(s_k)\cdot a} - e^{-E(s_k)\cdot b}\right),$$

where $a = \inf I'$ and $b = \sup I'$. (For $b = \infty$ and $\lambda > 0$, let $e^{-\lambda \cdot \infty} = 0$.) Note that

$$\int_{I'} E(s_k) \cdot e^{-E(s_k)\cdot t}\, dt = e^{-E(s_k)\cdot a} - e^{-E(s_k)\cdot b}$$

is the probability of taking a transition outgoing from state s_k in the interval I', where the probability density function of the residence time of s_k equals $E(s_k) \cdot e^{-E(s_k)\cdot t}$ (for time instant t).

As opposed to the traditional approach in real-time systems [6], we do not assume time divergence for infinite paths $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$. Although $\sum_{j \geq 0} t_j$ might converge, in which case σ represents an "unrealistic" computation where infinitely many transitions are taken in a finite amount of time, the probability measure of such non-time-divergent paths is 0 (independent of α) as stated in the following proposition. This allows a lazy treatment of the notation $\sigma@t$ in the description of measurable sets of paths.

Proposition 1. *For any state s_0, the probability measure of the set of infinite paths $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$ for which $\sum_{i \geq 0} t_i$ is converging is zero.*

Proof. Let $\mathcal{M} = (S, \mathbf{R}, L)$ be a CTMC and let $\Lambda = \max\{\mathbf{R}(s, s') \mid s, s' \in S\}$ the maximum rate in $\mathcal{M}$. Let $ConvPath(s)$ denote the set of all convergent paths that start in state s. We show that $\Pr(ConvPath(s)) = 0$.

Consider the set $B(s)$ of all paths σ that start in state s for which the delay of the transitions never exceeds one time unit. Formally, $B(s)$ consists of all paths $\sigma = s \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \to t_2 \ldots$ such that $t_i \leq 1$ for all i. We first show that the set $B(s)$ has probability measure 0. The cylinder set $B_n(s) = C(s, [0,1], s_1, [0,1], \ldots, [0,1], s_n)$ is the superset of $B(s)$ that contains exactly those paths $\sigma = s \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \to t_2 \ldots$, where $t_i \leq 1$ for all $i < n$. Clearly, $B(s) = \bigcap_{n \geq 1} B_n(s)$. By induction on n, we obtain: $\Pr(B_n(s)) \leq \left(1 - e^{-\Lambda}\right)^n$. Since $0 < 1 - e^{-\Lambda} < 1$, we obtain:

$$\Pr(B(s)) = \lim_{n \to \infty} \Pr(B_n(s)) = 0.$$

We now show that the probability measure of the set of convergent paths is 0. For any convergent path $\sigma = s \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \ldots$, the sum $\sum_{i=0}^{\infty} t_i$ converges. In particular, the sequence $(t_i)_{i \geq 1}$ converges to 0. Thus, there exists some natural number $n \geq 1$ with $t_i \leq 1$ for all $i > n$. This implies that the "suffix path" (starting in state s_n) $s_n \xrightarrow{t_n} s_{n+1} \xrightarrow{t_{n+1}} s_{n+2} \xrightarrow{t_{n+2}} \ldots$ belongs to $B(s_n)$. $ConvPath(s)$ is thus a subset of

$$\bigcup_{n \geq 1} \bigcup_{s_1, \ldots, s_n \in S} \{\sigma \in Path(s) \mid \sigma \text{ is of the form } s \xrightarrow{t_0} \ldots \xrightarrow{t_{n-2}} s_{n-1} \xrightarrow{t_{n-1}} \sigma' \text{ for some } \sigma' \in B(s_n) \text{ and } t_0, \ldots, t_{n-1} \in \mathbb{R}\}.$$

This yields:

$$\Pr(ConvPath(s)) \leq \sum_{n=0}^{\infty} \sum_{s_1, \ldots, s_n \in S} \Pr(B_n(s)) = 0.$$

Here, we use the fact that the measure of the set consisting of all paths with a prefix of the form $\sigma = s \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \ldots \xrightarrow{t_{n-1}} s_n$ (where $t_0, \ldots, t_n$ are arbitrary nonnegative reals) is at most 1, as it equals $\mathbf{P}(s, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \ldots \cdot \mathbf{P}(s_{n-1}, s_n)$. □

Proposition 1 can also be deduced, though not in an easy way, from [3, Theorem 4.1].

2.4 Steady-State and Transient-State Probabilities

For a CTMC, two major types of state probabilities are distinguished: steady-state probabilities where the system is considered "on the long run," i.e., when an equilibrium has been reached, and transient-state probabilities where the system is considered at a given time instant t. Formally, the transient probability

1. Formally, paths are maximal alternating sequences $s_0, t_0, s_1, t_1, s_2, \ldots$ that are either infinite or end in an absorbing state.

$$\pi^{\mathcal{M}}(\alpha, s', t) \;=\; \Pr_{\alpha}\{\sigma \in Path^{\mathcal{M}} \mid \sigma@t = s'\}$$

stands for the probability to be in state s' at time t given the initial distribution α.[2] We denote with $\underline{\pi}^{\mathcal{M}}(\alpha, t)$ the vector of state probabilities (ranging over states s') at time t, when the initial distribution equals α, i.e., $\underline{\pi}^{\mathcal{M}}(\alpha, t) = (\ldots, \pi^{\mathcal{M}}(\alpha, s, t), \ldots)$. The transient probabilities are characterized by a system of linear differential equations, also known as the forward Chapman-Kolmogorov differential equations [34], [50], [53], [69]:

$$\frac{d}{dt}\,\underline{\pi}^{\mathcal{M}}(\alpha, t) = \underline{\pi}^{\mathcal{M}}(\alpha, t)\cdot\mathbf{Q} \text{ given } \underline{\pi}^{\mathcal{M}}(\alpha, 0) = \alpha, \qquad (1)$$

where $\mathbf{Q}$ is the *infinitesimal generator* matrix of $\mathcal{M}$ defined by $\mathbf{Q} = \mathbf{R} - diag(\underline{E})$. $\underline{E} = diag(\underline{E})$ denotes the diagonal matrix with $\underline{E}(s, s) = E(s)$ and 0 otherwise.

Steady-state probabilities are given by [34], [50], [53], [69]: $\pi^{\mathcal{M}}(\alpha, s') = \lim_{t\to\infty} \pi^{\mathcal{M}}(\alpha, s', t)$. This limit always exists for finite CTMCs [53]. For $S' \subseteq S$, let $\pi^{\mathcal{M}}(\alpha, S') = \sum_{s'\in S'} \pi^{\mathcal{M}}(\alpha, s')$ denote the steady-state probability for S' given α, i.e.,

$$\pi^{\mathcal{M}}(\alpha, S') \;=\; \lim_{t\to\infty} \Pr_{\alpha}\{\sigma \in Path^{\mathcal{M}} \mid \sigma@t \in S'\}.$$

We let $\pi^{\mathcal{M}}(\alpha, \emptyset) = 0$. Probabilities $\pi^{\mathcal{M}}(\alpha, s')$ are computed from a system of linear equations

$$\underline{\pi}^{\mathcal{M}}(s)\cdot\mathbf{Q} = \underline{0} \text{ with } \sum_{s'} \pi^{\mathcal{M}}(s, s') = 1. \qquad (2)$$

Notational remarks: In case of a unique initial state s, i.e., $\alpha = \alpha^1_s$, we write $\Pr_s$ for $\Pr_\alpha$, $\pi(s, s', t)$ for $\pi(\alpha, s', t)$, and $\pi(s, s')$ for $\pi(\alpha, s')$. For strongly connected CTMCs, steady-state probabilities are independent of the initial distribution. We then write $\pi(s')$ for $\pi(\alpha, s')$.

Notice that the above two types of measures are truly *state-based*. In many cases, however, there is a need to determine the occurrence probability of certain (sets of) state *sequences*. Stated differently, we would also like to be able to express measures that address the probability on paths through the CTMC obeying particular properties. Except for the recent work by Obal and Sanders [60], suitable mechanisms to express such measures have not been considered. In Section 3, we will introduce a logic-based approach that allows us to express such path-based measures.

3 The Continuous Stochastic Logic CSL

This section presents the syntax and the semantics of the continuous stochastic logic CSL. Next to that, fixed-point characterizations will be given for the stochastic operators in the logic that serve as the basis for the model-checking algorithms for CSL.

3.1 Syntax of CSL

CSL is a branching-time temporal logic á la CTL [32] with state and path formulas based on [9]. The state formulas are interpreted over states of a CTMC, whereas the path formulas are interpreted over paths in a CTMC. CSL extends CTL with two probabilistic operators that refer to the steady state and transient behavior of the system being studied. Whereas the steady-state operator refers to the probability of residing in a particular set of *states* (specified by a state formula) in the long run, the transient operator allows us to refer to the probability of the occurrence of particular *paths* in the CTMC, similar to [38]. In order to express the time span of a certain path, the path operators until ($\mathcal{U}$) and next (X) will be extended with a parameter that specifies a time interval.

Definition 4. *Let $p \in [0, 1]$ be a real number, $\trianglelefteq \in \{\leq, <, >, \geq\}$ a comparison operator, and $I \subseteq \mathbb{R}_{\geq 0}$ a nonempty interval. The syntax of CSL formulas over the set of atomic propositions AP is defined inductively as follows:*

- tt *is a state-formula.*
- *Each atomic proposition $a \in AP$ is a state formula.*
- *If Φ and Ψ are state formulas, then so are $\neg\Phi$ and $\Phi \wedge \Psi$.*
- *If Φ is a state formula, then so is $\mathcal{S}_{\trianglelefteq p}(\Phi)$.*
- *If φ is a path formula, then $\mathcal{P}_{\trianglelefteq p}(\varphi)$ is a state formula.*
- *If Φ and Ψ are state formulas, then $X_I\,\Phi$ and $\Phi\,\mathcal{U}_I\,\Psi$ are path formulas.*

Before we provide the formal semantics, we give an informal explanation of the CSL formulas. $\mathcal{S}_{\trianglelefteq p}(\Phi)$ asserts that the steady-state probability for a Φ state meets the boundary condition $\trianglelefteq p$. $\mathcal{P}_{\trianglelefteq p}(\varphi)$ asserts that the probability measure of the paths satisfying φ meets the bound given by $\trianglelefteq p$. The operator $\mathcal{P}_{\trianglelefteq p}(.)$ replaces the usual CTL path quantifiers $\exists$ and $\forall$. Intuitively, $\exists\varphi$—there exists a path for which φ holds—corresponds to $\mathcal{P}_{>0}(\varphi)$, and $\forall\varphi$—for all paths φ holds—corresponds to $\mathcal{P}_{\geq 1}(\varphi)$. For instance, $\mathcal{P}_{>_0}(\diamond a)$ is equivalent to $\exists \diamond a$, and $\mathcal{P}_{\geq 1}(\diamond a)$ stands for $\forall \diamond a$ given a fair interpretation [33] of the CTL formula $\forall \diamond a$. In a fair interpretation of CTL, paths that do not satisfy certain fairness constraints, like "visit a set of states infinitely often," are ruled out. Satisfaction of formulae is only with respect to the remaining fair paths. An elaborate discussion about the relation between fairness and probabilities goes beyond the scope of this paper; we refer the interested reader to [14]. The temporal operator X^I is the timed variant of the standard next operator in CTL; the path formula $X^I\Phi$ asserts that a transition is made to a Φ state at some time point $t \in I$. Operator $\mathcal{U}^I$ is the timed variant of the until operator of CTL; the path formula $\Phi\,\mathcal{U}^I\Psi$ asserts that Ψ is satisfied at some time instant in the interval I and that at all preceding time instants Φ holds.

3.2 Semantics

The state formulas are interpreted over the states of a CTMC. Let $\mathcal{M} = (S, \mathbf{R}, L)$ with labels in AP. The meaning of CSL state formulas is defined by means of a satisfaction relation, denoted by $\models$, between a CTMC $\mathcal{M}$, one of its states s, and a state formula Φ. The pair (s, Φ) belongs to the relation $\models$, denoted by $s \models \Phi$, if and only if Φ is valid in s.

Definition 5. *Let $Sat(\Phi) = \{s \in S \mid s \models \Phi\}$. The relation $\models$ for CSL state formulas is defined by:*

2. The fact that the set $\{\sigma \in Path^{\mathcal{M}} \mid \sigma@t = s'\}$ is measurable, follows by easy verification.

$$
\begin{array}{ll}
s \models \mathrm{tt} & \textit{for all } s \in S \\
s \models a & \textit{iff } a \in L(s) \\
s \models \neg\Phi & \textit{iff } s \not\models \Phi \\
s \models \Phi \wedge \Psi & \textit{iff } s \models \Phi \wedge s \models \Psi \\
s \models \mathcal{S}_{\trianglelefteq p}(\Phi) & \textit{iff } \pi^{\mathcal{M}}(s, Sat(\Phi)) \trianglelefteq p \\
s \models \mathcal{P}_{\trianglelefteq p}(\varphi) & \textit{iff } Prob^{\mathcal{M}}(s, \varphi) \trianglelefteq p.
\end{array}
$$

$Prob^{\mathcal{M}}(s, \varphi)$ denotes the probability measure of all paths $\sigma \in Path$ satisfying φ when the system starts in state s, i.e.,

$$Prob^{\mathcal{M}}(s, \varphi) = \Pr_s\{\sigma \in Path^{\mathcal{M}} \mid \sigma \models \varphi\}.$$

The fact that the set $\{\sigma \in Path \mid \sigma \models \varphi\}$ is measurable can be verified from the Borel space construction in Section 2.3.

In a similar way as for state formulas, the meaning of path formulas is defined by means of a satisfaction relation, (also) denoted by $\models$, between a CTMC $\mathcal{M}$, one of its paths σ, and path formula φ.

Definition 6. *The relation $\models$ for CSL path formulas is defined by:*

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

We note that, for $I = \emptyset$, the formula $\Phi\, \mathcal{U}^I \Psi$ is not satisfiable. Other Boolean connectives are derived in the usual way, i.e., $\mathrm{ff} = \neg\mathrm{tt}$, $\Phi \vee \Psi = \neg(\neg\Phi \wedge \neg\Psi)$, and $\Phi \Rightarrow \Psi = \neg\Phi \vee \Psi$. The usual (untimed) next and until operator are obtained as follows: $X\Phi = X^{[0,\infty)}\Phi$ and $\Phi\, \mathcal{U} \Psi = \Phi\, \mathcal{U}^{[0,\infty)} \Psi$. In the sequel, intervals of the form $[0, \infty)$ are often omitted from the operators. Temporal operators like $\diamond$ ("eventually") and its real-time variant $\diamond^I$ are derived as follows:

$$\mathcal{P}_{\trianglelefteq p}(\diamond^I\, \Phi) = \mathcal{P}_{\trianglelefteq p}(\mathrm{tt}\, \mathcal{U}^I \Phi) \text{ and } \mathcal{P}_{\trianglelefteq p}(\diamond\, \Phi) = \mathcal{P}_{\trianglelefteq p}(\mathrm{tt}\, \mathcal{U} \Phi).$$

For $\Box$ ("always") and its timed variant $\Box^I$, we have, for example:

$$\mathcal{P}_{\geq p}(\Box^I\ \Phi) = \mathcal{P}_{\leq 1-p}(\diamond^I \neg\Phi) \text{ and } \mathcal{P}_{\geq p}(\Box\, \Phi) = \mathcal{P}_{\leq 1-p}(\diamond\, \neg\Phi).$$

3.3 Specifying Properties in CSL

What types of performance and dependability properties can be expressed using CSL? First, we remark that, in CSL, one does not specify a measure but, in fact, a constraint (or: bound) on a performance or dependability measure. Four types of measures can be identified: steady-state measures, transient-state measures, path-based measures, and nested measures.

Steady-state measures. The formula $\mathcal{S}_{\trianglelefteq p}(at_s)$ imposes a requirement on the steady-state probability to be in state s. (Recall that the atomic proposition at_s is valid in state s and invalid in any other state.) For instance, $\mathcal{S}_{\leq 10^{-5}}(at_{s_{2,1}})$ is valid in state $s_{3,1}$ (cf. the running example) if the steady-state probability of having a system configuration in which a single processor has failed is at most 0.00001 (when starting in state $s_{3,1}$). This can be easily generalized toward selecting sets of states by using more general state formulas. The formula $\mathcal{S}_{\trianglelefteq p}(\Phi)$ imposes a constraint on the probability to be in some Φ state on the long run. For instance, the formula $\mathcal{S}_{\geq 0.99}(up_3 \vee up_2)$ states that, in the long run, for at least 99 percent of the time, at least two processors are operational.

Transient measures. The combination of the probabilistic operator with the temporal operator $\diamond^{[t,t]}$ can be used to reason about transient probabilities since

$$\pi^{\mathcal{M}}(s, s', t) = Prob^{\mathcal{M}}(s, \diamond^{[t,t]}\ at_{s'}).$$

More specifically, $\mathcal{P}_{\trianglelefteq p}(\diamond^{[t,t]}\ at_{s'})$ is valid in state s if the transient probability at time t to be in state s' satisfies the bound $\trianglelefteq p$. For instance, $\mathcal{P}_{\leq 0.2}(\diamond^{[5,5]}\ at_{s_{2,1}})$ is valid in state $s_{3,1}$ if the transient probability of state $s_{2,1}$ at time t is at most 0.2 when starting in state $s_{3,1}$. In a similar way as done for steady-state measures, the formula $\mathcal{P}_{>0.99}(\diamond^{[t,t]}\ (up_3 \vee up_2))$ requires that the probability to have at least two processors running at time t exceeds 0.99.

Path-based measures. The standard transient measures on (sets of) states are expressed using a specific instance of the $\mathcal{P}$-operator. However, by the fact that this operator allows an arbitrary path formula as argument, much more general measures can be described. An example is the probability of reaching a certain set of states provided that all paths to these states obey certain properties. For instance,

$$\mathcal{P}_{\leq 0.01}((up_3 \vee up_2)\mathcal{U}^{[0,10]}\ down)$$

is valid in state $s_{3,1}$ if the probability of the system being $down$ within 10 time units after having continuously operated with at least two processors is at most 0.01 when starting in state $s_{3,1}$.

Nested measures. By nesting the $\mathcal{P}$ and $\mathcal{S}$ operators, more complex properties of interest can be specified. These are useful to obtain a more detailed insight into the system's behavior and allow, e.g., to express probabilistic (timed) reachability that are conditioned on the system being in equilibrium. The property

$$\mathcal{S}_{\geq 0.9}(\mathcal{P}_{\geq 0.8}\ \Box^{[0,10]} \neg down)$$

is valid in those states that guarantee that, in equilibrium with probability at least 0.9, the probability that the system will not go down within 10 time units is at least 0.8. Conversely,

$$\mathcal{P}_{\geq 0.5}((\neg down)\mathcal{U}^{[10,20]}\mathcal{S}_{\geq 0.8}((up_3 \vee up_2)))$$

is valid for those states that,with probability at least 0.5, will reach a state s between 10 and 20 time units, which guarantees the system to be operational with at least two processors when the system is in equilibrium. Besides, prior to reaching state s, the system must be operational continuously. These measures are of interest from a practical point of view, but could not be expressed precisely before.

To summarize, Table 1 surveys some performance and dependability measures and their formulation in CSL, where up characterizes all states in which the system is operational.

There are two main benefits when using CSL for specifying constraints on measures-of-interest over CTMCs. First, the specification is entirely formal such that the interpretation is unambiguous. Whereas this is also the case for standard transient and steady-state measures (like

TABLE 1
Measures and Their Logical Specification

(a)	steady-state availability	$\mathcal{S}_{\trianglelefteq p}(up)$
(b)	instantaneous availability at time t	$\mathcal{P}_{\trianglelefteq p}(\Diamond^{[t,t]} up)$
(c)	conditional instantaneous availability at time t	$\mathcal{P}_{\trianglelefteq p}(\Phi\, \mathcal{U}^{[t,t]} up)$
(d)	interval availability	$\mathcal{P}_{\trianglelefteq p}(\Box^{[t,t']} up)$
(e)	steady-state interval availability	$\mathcal{S}_{\trianglelefteq p}(\mathcal{P}_{\trianglelefteq q}(\Box^{[t,t']} up))$
(f)	conditional time-bounded steady-state availability	$\mathcal{P}_{\trianglelefteq p}(\Phi\, \mathcal{U}^{[t,t']}\, \mathcal{S}_{\trianglelefteq q}(up))$

Cases a and b in Table 1), this often does not apply to measures that are derived from these elementary measures. Such measures are typically described in an informal manner. A rigorous specification of such more intricate measures is of the utmost importance for their automated analysis (as proposed in the sequel). Furthermore, an important aspect of CSL is the possibility of stating performance and dependability requirements over a selective set of paths through a model, which was not possible previously. Finally, the possibility of nesting steady-state and transient measures provides a means to specify complex, though important measures in a compact and flexible way.

3.4 Model-Checking CSL

Once we have formally specified the (constraint on the) measure-of-interest in CSL by a formula Φ and have obtained the model, i.e., CTMC $\mathcal{M}$, of the system under consideration, the next step is to model check the formula. To that end, we adapt the model-checking algorithm for CTL [23] to support the automated validation of Φ over a given state s in $\mathcal{M}$. The basic procedure is as for model checking CTL: In order to check whether state s satisfies formula Φ, we recursively compute the set $Sat(\Phi)$ of states that satisfy Φ and, finally, check whether s is a member of that set. For the nonprobabilistic state operators, this procedure is the same as for CTL. The only main remaining question is how to compute $Sat(\Phi)$ for the $\mathcal{S}$ and $\mathcal{P}$ operators. We deal with these operators separately.

Computing steady-state measures. Let G be the underlying directed graph of $\mathcal{M}$, where vertices represent states and where there is an edge from s to s' iff $\mathbf{R}(s,s') > 0$. Subgraph B is a *bottom strongly connected component* (BSCC) of G if it is a maximal strongly connected component such that it has no edges to outside its vertices, i.e., $Reach(s) = B$ for all $s \in B$. Let $B(\mathcal{M})$ denote the BSCCs of CTMC $\mathcal{M}$. From the semantics of CSL state formulas, it directly follows that $s \in Sat(\mathcal{S}_{\trianglelefteq p}(\Phi))$ iff $\pi(s, Sat(\Phi)) \trianglelefteq p$. In order to compute the probability $\pi(s, S')$ for some set of states S', we exploit the following result.

Proposition 2. *For CTMC $\mathcal{M} = (S, \mathbf{R}, L)$ with $s \in S$, $S' \subseteq S$:*

$$\pi(s, S') = \sum_{B \in B(\mathcal{M})} \left(Prob(s, \diamond\, at_B) \cdot \sum_{s' \in B \cap S'} \pi^B(s') \right),$$

where $\pi^B(s')$ is the steady-state probability of s' in BSCC B.

Proof. Follows directly from [53, Theorem 6.16]. □

Proposition 2 suggests the following algorithm for checking whether $s \models \mathcal{S}_{\trianglelefteq p}(\Phi)$: We first recursively determine the set of states that satisfy Φ. Then, the BSCCs are computed using (a slight variant of) an algorithm for computing strongly connected components [1], [70]. For each BSCC B that contains a Φ state, the steady-state probabilities are determined using standard means for solving the linear equation system (2). Formally, $\pi^B(s') = 1$ if $B = \{s'\}$; otherwise, $\underline{\pi}^B$ is a vector of size $|B|$ satisfying the linear system of equations for state s' in B:

$$\sum_{\substack{s \in B \\ s \neq s'}} \pi^B(s) \cdot \mathbf{R}(s,s') = \pi^B(s') \cdot \sum_{\substack{s \in B \\ s \neq s'}} \mathbf{R}(s',s)$$
$$\text{with } \sum_{s \in B} \pi^B(s) = 1.$$

States not contained in any BSCC have steady-state probability 0, independent of the initial state. The cumulative probability $\sum \pi^B(s')$ is weighted with the probability of eventually reaching BSCC B. These weights can be computed as the least solution in $[0,1]$ of the linear equation system:

$$Prob(s, \diamond at_B) = \begin{cases} 1 & \text{if } s \models at_B \\ \sum_{s'} \mathbf{P}(s,s') \cdot Prob(s', \diamond\, at_B) & \text{otherwise} \end{cases}$$

Note that for the—often encountered—case in which the CTMC $\mathcal{M}$ is strongly connected, i.e., $\mathcal{M}$ has a single BSCC consisting of all states, this entire procedure reduces to a standard steady-state analysis [69] and cumulating the steady-state probabilities for all Φ states.

Example 2. Consider CTMC $\mathcal{M}$ depicted in Fig. 2 and let us check $\mathcal{S}_{>0.75}(b)$ in state s_0. Clearly, $\mathcal{M}$ is not strongly connected and has three BSCCs as indicated by the gray shaded sets of states: $B_1 = \{s_3\}$, $B_2 = \{s_4\}$, and $B_3 = \{s_2, s_5\}$. As states s_3 and s_5 are the only b states, we have:

$$\pi(s_0, Sat(b)) = Prob(s_0, \diamond\, at_{B_1}) \cdot \pi^{B_1}(s_3) + Prob(s_0, \diamond\, at_{B_3}) \cdot \pi^{B_3}(s_5).$$

The probability of eventually reaching BSCC B_1 from state s_0 is obtained by solving:

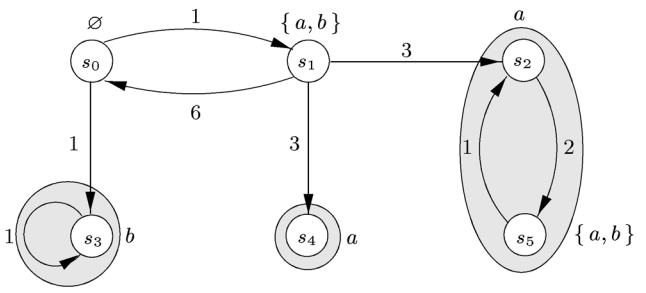


Fig. 2. An example nonstrongly connected CTMC.

$$Prob(s_0, \diamond\, at_{B_1}) = \frac{1}{2} + \frac{1}{2} \cdot Prob(s_1, \diamond\, at_{B_1}),$$
$$Prob(s_1, \diamond\, at_{B_1}) = \frac{1}{2} \cdot Prob(s_0, \diamond\, at_{B_1}).$$

which yields $\frac{1}{2} \cdot \sum_{k=0}^{\infty} (\frac{1}{4})^k = \frac{2}{3}$. Likewise, the probability of reaching B_3 from s_0 equals $\frac{1}{6}$. Trivially, $\pi^{B_1}(s_3) = 1$. $\pi^{B_3}(s_5)$ is obtained by solving the equation system

$$-2\pi^{B_3}(s_2) + \pi^{B_3}(s_5) = 0 \text{ and } \pi^{B_3}(s_2) + \pi^{B_3}(s_5) = 1.$$

This yields $\pi^{B_3}(s_5) = \frac{2}{3}$. Following Proposition 2, we have $\pi(s_0, Sat(b)) = \frac{7}{9}$. As this exceeds the bound 0.75, it follows $s_0 \models \mathcal{S}_{>0.75}(b)$.

Computation of probabilistic path-measures. The basis for model-checking probabilistic path-formulas is characterizations for $Prob(s, \varphi)$, i.e., the probability measure for the set of paths that fulfill φ and that start in s. We consider such characterizations for the timed-next and timed-until operators and show that the characterizations for their untimed variants coincide with those for model checking PCTL over discrete-time Markov chains [10], [27], [38]. We first observe that it suffices to consider time bounds specified by closed intervals since:

$$Prob(s, \Phi\, \mathcal{U}^I \Psi) = Prob(s, \Phi\, \mathcal{U}^{cl(I)} \Psi) \text{ and}$$
$$Prob(s, X^I \Phi) = Prob(s, X^{cl(I)} \Phi),$$

where $cl(I)$ denotes the closure of I. This follows from the fact that the probability measure of a basic cylinder set $C(s, I_0, \ldots, I_{k-1}, s_k)$ does not change when some of the intervals I_i $(0 \le i < k)$ are replaced by their closure. In the sequel, we assume that interval I is compact.

Proposition 3. *For $s \in S$, interval $I \subseteq \mathbb{R}_{\ge 0}$ and CSL state formula Φ:*

$$Prob(s, X^I \Phi) = \left(e^{-E(s) \cdot \inf I} - e^{-E(s) \cdot \sup I}\right) \cdot \sum_{s' \models \Phi} \mathbf{P}(s, s').$$

Proof. By straightforward verification from the Borel space construction. □

Proposition 3 suggests the following algorithm: First, set $Sat(\Phi)$ is computed. State s is added to $Sat(\mathcal{P}_{\trianglelefteq p}(X^I \Phi))$ if $Prob(s, X^I \Phi) \trianglelefteq p$. Vector

$$\underline{Prob}(X^I \Phi) = (\ldots, Prob(s, X^I \Phi), \ldots)$$

can be obtained by multiplying $\mathbf{P}$ with the vector $\underline{b}_I$, where $b_I(s) = e^{-E(s) \cdot \inf I} - e^{-E(s) \cdot \sup I}$ if $s \in Sat(\Phi)$ and $b_I(s) = 0$ otherwise.

For time-bounded until formula φ, $Prob(s, \varphi)$ is characterized by a fixed-point equation. This is similar to CTL [23] where appropriate fixed-point characterizations constitute the key toward model checking until formulas. In the sequel, let $I \ominus x$ denote $\{t - x \mid t \in I \wedge t \ge x\}$ and $\mathbf{T}(s, s', x)$ denote the density of moving from state s to s' in x time units, i.e., $\mathbf{T}(s, s', x) = \mathbf{P}(s, s') \cdot E(s) \cdot e^{-E(s) \cdot x} = \mathbf{R}(s, s') \cdot e^{-E(s) \cdot x}$. Recall that $E(s) \cdot e^{-E(s) \cdot x}$ is the probability density function of the residence time in state s at instant x. Let $\mathcal{I}$ denote the set of all (nonempty) intervals $I \subseteq \mathbb{R}_{\ge 0}$.

Theorem 1. *Let $s \in S$, interval $I \subseteq \mathbb{R}_{\ge 0}$ with $a = \inf I$ and $b = \sup I$ and Φ, Ψ be CSL state formulas. The function $S \times \mathcal{I} \to [0, 1]$, $(s, I) \mapsto Prob(s, \Phi\, \mathcal{U}^I \Psi)$ is the least fixed point of the higher-order operator*

$$\Omega : (S \times \mathcal{I} \to [0, 1]) \to (S \times \mathcal{I} \to [0, 1]),$$

where

$$\Omega(F)(s, I) = \begin{cases} 1 & \text{if } s \models \neg\Phi \wedge \Psi \text{ and } a = 0 \\ \int_0^b \sum_{s' \in S} \mathbf{T}(s, s', x) \cdot F(s', I \ominus x)\, dx & \text{if } s \models \Phi \wedge \neg\Psi \\ e^{-E(s) \cdot a} + \int_0^a \sum_{s' \in S} \mathbf{T}(s, s', x) \cdot F(s', I \ominus x)\, dx & \text{if } s \models \Phi \wedge \Psi \\ 0 & \text{otherwise.} \end{cases}$$

Proof. First, we show that the function $(s, I) \mapsto Prob(s, I) = Prob(s, \Phi\, \mathcal{U}^I \Psi)$ is a fixed point of Ω. Let $s \in S$ and I be an arbitrary nonempty interval in $\mathbb{R}_{\ge 0}$. We define $a = \inf I$, $b = \sup I$. $Path(s, I)$ denotes the collection of all paths σ that start in state s and fulfill the path formula $\Phi\, \mathcal{U}^I \Psi$. We may assume w.l.o.g. that I is closed, i.e., $I = [a, b]$ if $b < \infty$ and $I = [a, \infty[$ if $b = \infty$. We consider the following cases:

Case 1. $a = 0$ and $s \models \Psi$. Then, any path starting in s satisfies $\Phi\, \mathcal{U}^I \Psi$. Hence,

$$Prob(s, I) = 1 = \Omega(Prob)(s, I).$$

Case 2. $s \models \Phi \wedge \neg\Psi$. Then, $Path(s, I)$ consists of all paths σ which are of the form $s \xrightarrow{x} \sigma'$, where $0 \le x \le b$ and $\sigma' \in Path(s', I \ominus x)$ for some state s'. Hence,

$$Prob(s, I) = \int_0^b \sum_{s' \in S} \mathbf{T}(s, s', x) \cdot Prob(s', I \ominus x)\, dx.$$

Case 3. $a > 0$ and $s \models \Phi \wedge \Psi$. Then, $Path(s, I)$ consists of all paths σ of the form $s \xrightarrow{x} \sigma'$, where 1) either $0 \le x \le a$ and $\sigma' \in Path(s', I \ominus x)$ for some state s' or 2) $x > a$. Thus, $Prob(s, I)$ is the sum of the probability to stay for more than x time units in state s' plus the probability to take a transition from s to s' within x time units (where $x \le a$) and to fulfill $\Phi\, \mathcal{U}^{I \ominus x} \Psi$ along a path starting in s'. We obtain

$$Prob(s,I) = e^{-E(s)\cdot a} + \int_0^a \sum_{s'\in S} \mathbf{T}(s,s',x)\cdot Prob(s', I\ominus x)\, dx.$$

It is clear that $Prob(s,I)=0$ in all remaining cases (if $s \not\models \Phi \vee \Psi$ or $a>0$ and $s \models \neg\Phi \wedge \Psi$).

We now explain why the function $(s,I)\mapsto Prob(s,I)$ is the *least* fixed point of Ω. Let $Path_n(s,I)$ denote the set of all paths $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{n-1}} s_{n-1} \xrightarrow{t_n} s_n \xrightarrow{t_{n+1}} \ldots$, where $s_0 = s$ and n is minimal with the properties 1) $s_0,\ldots,s_{n-1} \models \Phi$ and $s_n \models \Psi$ and 2) $t_0+t_1+\ldots+t_n \in I$. Let $Path_{\le n}(s,I) = \bigcup_{0\le i\le n} Path_i(s,I)$ and $Prob_{\le n}(s,I)$ the probability measure of all paths $\sigma \in Path_{\le n}(s,I)$. It is easy to see that, for $n=1,2,3,\ldots$,

$$Prob_{\le n}(s,I) = \Omega(Prob_{\le n-1})(s,I).$$

Moreover, $\lim_{n\to\infty} Prob_{\le n}(s,I) = Prob(s,I)$. Let $F: S\times \mathcal{I} \to [0,1]$ be a fixed point of Ω. By induction on n, we obtain: $F(s,I) \ge Prob_{\le n}(s,I)$. This yields $F(s,I) \ge \lim_{n\to\infty} Prob_{\le n}(s,I) = Prob(s,I)$. □

A few remarks are in order. First, in the special case $I=[t,t]$ (where $t>0$) and $\Phi = \mathrm{tt}$, $\Psi = at_{s''}$ for some state s'' we obtain:

$$\pi^{\mathcal{M}}(s,s'',t) = Prob(s, \diamond^{[t,t]} at_{s''}) = \int_0^t \sum_{s'\in S} \mathbf{T}(s,s',x)\cdot \pi^{\mathcal{M}}(s',s'',t-x)\, dx.$$

From this, the Chapman-Kolmogorov differential equation system (see Section 2.4) can be derived. A second observation is that the recursive characterization for unbounded intervals, e.g., $I=[t,\infty[$, yields that $Prob(s,\Phi\,\mathcal{U}^{\ge t}\,\Psi)$ equals

$$\int_0^t \sum_{s'\in S} \mathbf{T}(s,s',x)\cdot Prob(s', I\ominus x)\, dx + \sum_{s'\models\Psi} \mathbf{P}(s,s')\Big(1-e^{E(s)\cdot t}\Big).$$

The characterization in Theorem 1 is informally justified as follows: If s satisfies Φ and $\neg\Psi$, the probability of reaching a Ψ state from s within the interval I equals the probability of reaching some direct successor s' of s in x time units ($x \le \sup I$), multiplied by the probability of reaching a Ψ state from s' in the remaining time interval $I\ominus x$ (along a Φ path). If s satisfies $\Phi\wedge\Psi$, the path formula φ is satisfied if no transition outgoing from s is taken for at least $\inf I$ time units (first summand). Alternatively, state s should be left before $\inf I$ in which case the probability is defined in a similar way as for the case $s\models\Phi\wedge\neg\Psi$ (second summand). Note that $\inf I = 0$ is possible. In this case, $s\models\Phi\wedge\Psi$ yields $Prob^{\mathcal{M}}(s,\Phi\,\mathcal{U}^I\Psi)=1$.

Example 3. Consider our running CTMC example again with $\lambda=0.01$ and $\nu=0.001$. We check $s_{3,1} \models \mathcal{P}_{\ge 0.2}(up_3\,\mathcal{U}^{[2,5]}\,up_2)$. According to Theorem 1, it follows:

$$Prob(s_{3,1}, up_3\,\mathcal{U}^{[2,5]}\,up_2) = \int_0^5 3\lambda\cdot e^{-(3\lambda+\nu)\cdot x}\cdot Prob(s_{3,1}, up_3\,\mathcal{U}^{[2-x,5-x]}\,up_2)\, dx.$$

Distinguishing the cases $x\le 2$ and $2\le x\le 5$ yields

$$Prob(s_{3,1}, up_3\,\mathcal{U}^{[2,5]}\,up_2) = \int_0^2 3\lambda\cdot e^{-(3\lambda+\nu)\cdot x}\cdot Prob(s_{2,1}, up_3\,\mathcal{U}^{[2-x,5-x]}\,up_2)\, dx + \int_2^5 3\lambda\cdot e^{-(3\lambda+\nu)\cdot x}\cdot Prob(s_{2,1}, up_3\,\mathcal{U}^{[0,5-x]}\,up_2)\, dx.$$

According to Theorem 1, $Prob(s_{2,1}, up_3\,\mathcal{U}^{[2-x,5-x]}\,up_2)$ is 0 for $x\le 2$ (case otherwise) and equals 1 for $2\le x\le 5$ so that we finally obtain:

$$Prob(s_{3,1}, up_3\,\mathcal{U}^{[2,5]}\,up_2) = \int_2^5 3\lambda\cdot e^{-(3\lambda+\nu)\cdot x}\, dx = \Big(e^{-(3\lambda+\nu)\cdot 2} - e^{-(3\lambda+\nu)\cdot 5}\Big)\cdot\frac{3\lambda}{3\lambda+\nu}.$$

This probability equals 0.2006 and as this exceeds 0.2, $\mathcal{P}_{\ge 0.2}(up_3\,\mathcal{U}^{[2,5]}up_2)$ is fulfilled by state $s_{3,1}$.

We discuss specific algorithms to compute $\underline{Prob}(\Phi\,\mathcal{U}^I\,\Psi)$ in Section 4.

Corollary 1. *For $s\in S$ and Φ,Ψ CSL state formulas:*

1. $Prob(s, X\,\Phi) = \sum_{s'\models\Phi} \mathbf{P}(s,s')$.
2. *The function $S\to[0,1]$, $s\mapsto Prob(s,\Phi\,\mathcal{U}\,\Psi)$ is the least fixed point of the higher-order operator $\Theta : (S\to[0,1]) \to (S\to[0,1])$ where:*

$$\Theta(F)(s) = \begin{cases} 1 & \text{if } s\models\Psi \\ \sum_{s'\in S} \mathbf{P}(s,s')\cdot F(s') & \text{if } s\models\Phi\wedge\neg\Psi \\ 0 & \text{otherwise.}\end{cases}$$

Proof. Directly from Proposition 3, Theorem 1, and the fact that $X = X^{[0,\infty)}$ and $\mathcal{U} = \mathcal{U}^{[0,\infty)}$. □

The results in Corollary 1 are identical to the discrete-time probabilistic case, i.e., the probabilities in DTMCs for satisfying next and until formulas in the logic PCTL are determined in the same way, cf. [10], [38]. This suggests the following algorithms: For the formulas $\mathcal{P}_{\trianglelefteq p}(X\,\Phi)$, the computation boils down to a simple matrix-vector multiplication, i.e., the vector $\underline{Prob}(X\,\Phi) = \mathbf{P}\cdot\underline{i}_\Phi$, where $\underline{i}_\Phi$ characterizes the set $Sat(\Phi)$ ($i_\Phi(s)=1$ if $s\models\Phi$ and $i_\Phi(s)=0$ otherwise). For $\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}\,\Psi)$, solving a system of linear equations suffices; the vector $\underline{Prob}(\Phi\,\mathcal{U}\,\Psi)$ is the least solution of the following set of equations:

$$\underline{x} = \widehat{\mathbf{P}}\cdot\underline{x} + \underline{i}_\Psi \text{ where } \widehat{\mathbf{P}}(s,s') = \mathbf{P}(s,s') \text{ if } s\models\Phi\wedge\neg\Psi \text{ and } 0 \text{ otherwise.}$$

This system of equations can, in general, have more than one solution. The least solution can be obtained by applying an iterative method or a graph analysis combined with standard methods (like Gaussian elimination or some iterative method [69]) to solve regular linear equation systems. As for the discrete-time probabilistic case, more efficient, tailored algorithms can be used to check $\mathcal{P}_{>0}(\Phi\,\mathcal{U}\,\Psi)$ and $\mathcal{P}_{\ge 1}(\Phi\,\mathcal{U}\,\Psi)$; see [27], [38].

4 Exploiting Transient Analysis

One of the main results of the previous section was the characterization of probability measures for time-bounded until formulas in terms of a Volterra integral equation system. This section first briefly discusses some numerical techniques to solve this equation system directly. To overcome the encountered problems in doing so, we propose a strategy that reduces the model-checking problem for time-bounded until properties to the problem of calculating transient probabilities in CTMCs. This is presented in Section 4.2. This strategy allows us to implement model checking of $\mathcal{U}^I$ by means of a well-established transient analysis techniques for CTMCs such as uniformization.

4.1 Numerically Solving the Integral Equation System

Two obvious techniques that could be applied to solve the recursive integral equation of Theorem 1 is to either use numerical integration or to solve the differential equation system that corresponds to the integrals directly. We briefly discuss both approaches for $\varphi = \Phi\, \mathcal{U}^{[0,t]}\, \Psi$ and argue that these techniques are not attractive for our purposes.

Theorem 1 suggests the following *iterative* method to approximate the probability $Prob(s, \varphi)$: let $F_0(s,t) = 0$ for all s, t, and $F_{k+1} = \Omega(F_k)$. Then,

$$\lim_{k\to\infty} F_k(s,t) = Prob(s, \Phi\, \mathcal{U}^{[0,t]}\, \Psi).$$

Each step in the iteration amounts to solve an integral of the following form:

$$F_{k+1}(s,t) = \int_0^t \sum_{s'\in S} \mathbf{R}(s,s')\cdot e^{-E(s)\cdot x}\cdot F_k(s', t-x)\, dx,$$

if $s \models \Phi \wedge \neg\Psi$. These integrals can be solved numerically using integration methods such as trapezoidal, Simpson and Romberg integration [63]. Experiments have shown that this approach is rather time consuming and that numerical stability is hard to achieve [43].

Alternatively, the recursive integral formula equation of Theorem 1 can be reformulated as a heterogeneous linear differential equation of the following form. With $\underline{y}(t)$ denoting the vector $\underline{Prob}(\Phi\, \mathcal{U}^{[0,t]}\, \Psi)$, we have:

$$\underline{y}'(t) = \widehat{\mathbf{R}} \cdot \underline{y}(t) + \underline{b}(t), \text{ where}$$

$$\widehat{\mathbf{R}}(s,s') = \begin{cases} \mathbf{R}(s,s') & \text{if } s, s' \models \Phi \wedge \neg\Psi \\ 0 & \text{otherwise} \end{cases} \text{ and}$$

$$b_s(t) = \begin{cases} \sum_{s'\models\Psi} \mathbf{R}(s,s') \cdot e^{-E(s)\cdot t} & \text{if } s \models \Phi \wedge \neg\Psi \\ 0 & \text{otherwise.} \end{cases}$$

The vector $\underline{y}(t)$ agrees with the following solution of the above heterogeneous linear differential equation:

$$\underline{y}(t) = e^{\widehat{\mathbf{R}}\cdot t}\cdot\left(\underline{i}_\Psi + \int_0^t e^{-\widehat{\mathbf{R}}\cdot x}\cdot \underline{b}(x)\, dx\right)$$

$$\text{where } e^{\widehat{\mathbf{R}}\cdot x} = \sum_{k=0}^{\infty} \frac{(\widehat{\mathbf{R}}\cdot x)^k}{k!}.$$

Unfortunately, there does not seem to exist a closed-form solution for this integral. In the sequel, we present transformations of the CTMC that avoid the integration, still resulting in a numerical solution for $\underline{y}(t)$.

4.2 Four Correctness-Preserving Transformations

We now propose a strategy that reduces the model-checking problem for time-bounded until to a transient analysis of the CTMC. This is inspired by the observation that determining the transient state probabilities of a CTMC at time t, say, corresponds to calculating the probabilities of the path-formula $\diamond^{[t,t]} at_{s'}$, for some initial state s:

$$\pi^{\mathcal{M}}(s, s', t) = Prob^{\mathcal{M}}(s, \diamond^{[t,t]} at_{s'}).$$

As a slight generalization, we obtain (cf. Section 3.3):

$$\pi^{\mathcal{M}}(s, Sat(\Phi), t) = Prob^{\mathcal{M}}(s, \diamond^{[t,t]}\Phi) = \sum_{s'\models\Phi} \pi^{\mathcal{M}}(s, s', t) \quad (3)$$

for arbitrary state-formula Φ.

We first observe that unbounded time intervals $[t, \infty)$ can be treated by combining time-bounded until and unbounded until since:

$$Prob(s, \Phi\, \mathcal{U}^{[t,\infty)}\, \Psi) = \sum_{s'\models\Phi} Prob(s, \Phi\, \mathcal{U}^{[t,t]}\, at_{s'}) \cdot Prob(s', \Phi\, \mathcal{U}\, \Psi).$$

In the sequel, we partition the problem into four types of time-bounded until formulas with a nonempty compact interval I and show how they all can be reduced to instances of two simple base cases. We first define a CSL state-formula-driven transformation on CTMCs.

Definition 7. *For CTMC $\mathcal{M} = (S, \mathbf{R}, L)$ and CSL state formula Φ let CTMC $\mathcal{M}[\Phi]$ result from $\mathcal{M}$ by making all Φ states in $\mathcal{M}$ absorbing, i.e., $\mathcal{M}[\Phi] = (S, R', L)$, where $R'(s,s') = R(s,s')$ if $s \not\models \Phi$ and 0 otherwise.*

Note that $\mathcal{M}[\Phi][\Psi] = \mathcal{M}[\Phi \vee \Psi]$.

Case A: Time-bounded until for absorbing goal states. Let $\varphi = \Phi\, \mathcal{U}^{[0,t]}\, \Psi$ and assume that all Ψ states are absorbing, i.e., once a Ψ state is reached it cannot be left anymore. We first observe that once a $(\neg\Phi \wedge \neg\Psi)$ state is reached, φ will be invalid, regardless of the future evolution of the system. As a result, we may switch from $\mathcal{M}$ to $\mathcal{M}[\neg\Phi \wedge \neg\Psi]$ and consider the property on the new CTMC. The assumption that all Ψ states are absorbing allows us to conclude that φ is satisfied if a Ψ state is occupied at time t. Thus,

Proposition 4. *If all Ψ states are absorbing in $\mathcal{M}$, i.e., $\mathcal{M} = \mathcal{M}[\Psi]$, then:*

$$Prob^{\mathcal{M}}(s, \Phi\, \mathcal{U}^{[0,t]}\, \Psi) = Prob^{\mathcal{M}[\neg\Phi\wedge\neg\Psi]}(s, \diamond^{[t,t]}\, \Psi) = \sum_{s''\models\Psi} \pi^{\mathcal{M}[\neg\Phi\wedge\neg\Psi]}(s, s'', t).$$

Proof. From the CSL semantics it follows that $Prob^{\mathcal{M}}(s, \Phi\, \mathcal{U}^{[0,t]}\, \Psi)$ equals

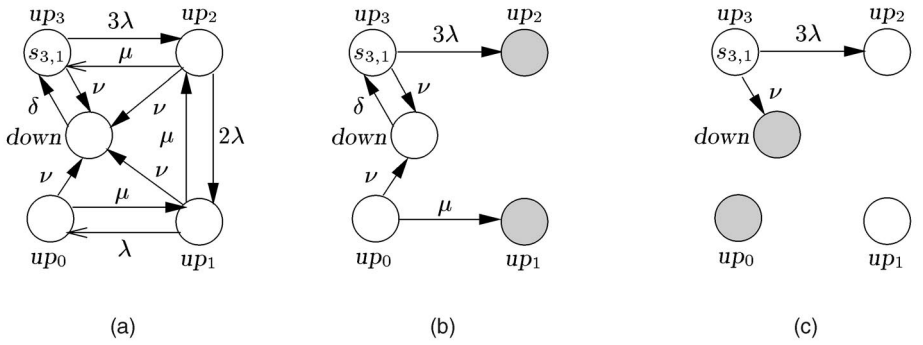


Fig. 3. Successive transformations on the TMR example for $(up_3 \vee up_2)\mathcal{U}^{[0,t]}(up_2 \vee up_1)$.

$$\Pr_s\{\sigma \in Path^{\mathcal{M}} \mid \exists x \in [0,t].\ \sigma@x \models \Psi \wedge (\forall y \in [0,x).\ \sigma@y \models \Phi)\}. \tag{4}$$

Since Ψ states are absorbing, it follows that once σ reaches a Ψ state at time instant x, then it will stay in Ψ states at all later time instants. This reduces (4) to:

$$\Pr_s\{\sigma \in Path^{\mathcal{M}} \mid \sigma@t \models \Psi \wedge (\forall y \in [0,t).\sigma@y \models \Phi \vee \Psi)\}. \tag{5}$$

There are no paths in $\mathcal{M}[\neg\Phi \wedge \neg\Psi]$ relevant for this probability measure that pass through $\neg(\Phi \vee \Psi)$ states. Thus, (5) equals:

$$\Pr_s\{\sigma \in Path^{\mathcal{M}[\neg\Phi\wedge\neg\Psi]} \mid \sigma@t \models \Psi\} = Prob^{\mathcal{M}[\neg\Phi\wedge\neg\Psi]}(s, \diamond^{[t,t]}\ \Psi),$$

which can be computed in a standard way, cf. (3). □

Case B: Time-bounded until. Let $\varphi = \Phi\ \mathcal{U}^{[0,t]}\ \Psi$ and consider an arbitrary CTMC $\mathcal{M}$. Property φ is fulfilled if a Ψ state is reached before (or at) time t via some Φ path. Once such a Ψ state has been reached, the future behavior of the CTMC is irrelevant for the validity of φ. Accordingly, the Ψ states can safely be made absorbing without affecting the validity of φ. As a result, it suffices to consider the probability of being in a Ψ state at time t for $\mathcal{M}[\Psi]$, thus reducing to the case in Proposition 4. As $\mathcal{M}[\Psi][\neg\Phi \wedge \neg\Psi] = \mathcal{M}[\neg\Phi \vee \Psi]$, we obtain:

Theorem 2. *For any CTMC $\mathcal{M}$:*

$$\begin{aligned} &Prob^{\mathcal{M}}(s, \Phi\ \mathcal{U}^{[0,t]}\ \Psi) \\ &= Prob^{\mathcal{M}[\Psi]}(s, \Phi\ \mathcal{U}^{[0,t]}\ \Psi) \\ &= \sum_{s'' \models \Psi} \pi^{\mathcal{M}[\neg\Phi\vee\Psi]}(s, s'', t). \end{aligned}$$

Proof. Straightforward verification. □

Example 4. Consider the TMR model with initial distribution $\alpha = \alpha^1_{s_{3,1}}$ and let $\Phi' = \mathcal{P}_{\geq 0.05}\ (\Phi\ \mathcal{U}^{[0,4]}\ \Psi)$ for $\Phi = up_3 \vee up_2$ and $\Psi = up_2 \vee up_1$ be the property under consideration for state $s_{3,1}$. According to the first part of Theorem 2, model checking Φ' on the original CTMC of Example 1, depicted in Fig. 3a, amounts to verifying this property on the CTMC depicted in Fig. 3b where the gray-shaded states indicate the (now absorbing) states satisfying Ψ. Proposition 4 now yields that it suffices to check the property $\mathcal{P}_{\geq 0.05}(\diamond^{[4,4]}\Psi)$ on the CTMC of Fig. 3c, where the gray-shaded states indicate the $(\neg\Phi \wedge \neg\Psi)$ states. The transient-state probability of the only remaining reachable state that satisfies Ψ, i.e., state $s_{2,1}$, is approximately 0.041 at $t = 4$ (for $\lambda = 0.01$ and $\nu = 0.001$); so, Φ' is invalid for state $s_{3,1}$.

Case C: Point-interval until. Let $\varphi = \Phi\ \mathcal{U}^{[t,t]}\ \Psi$ and assume $\Psi \Rightarrow \Phi$. Similar to Proposition 4, $(\neg\Phi \wedge \neg\Psi)$ states are made absorbing. Since $\Psi \Rightarrow \Phi$, it follows that $Prob(s, \varphi)$ equals the probability to occupy a Ψ state at time t in the obtained CTMC:

Proposition 5. *If $\Psi \Rightarrow \Phi$, we have for any CTMC $\mathcal{M}$:*

$$\begin{aligned} &Prob^{\mathcal{M}}(s, \Phi\ \mathcal{U}^{[t,t]}\ \Psi) \\ &= Prob^{\mathcal{M}[\neg\Phi\wedge\neg\Psi]}(s, \diamond^{[t,t]}\ \Psi) \\ &= \sum_{s'' \models \Psi} \pi^{\mathcal{M}[\neg\Phi\wedge\neg\Psi]}(s, s'', t). \end{aligned}$$

Proof. Straightforward verification. □

Case D: Interval until. Let $\varphi = \Phi\ \mathcal{U}^{[t,t']}\ \Psi$ with $0 < t \leq t'$ and let $\mathcal{M}$ be an arbitrary CTMC. First, we observe that

$$\begin{aligned} &Prob(s, \Phi\ \mathcal{U}^{[t,t']}\ \Psi) \\ &\neq Prob(s, \Phi\ \mathcal{U}^{[0,t']}\ \Psi) - Prob(s, \Phi\ \mathcal{U}^{[0,t]}\ \Psi), \end{aligned}$$

e.g., for a CTMC with just a single state s equipped with a self-loop, where s satisfies Φ and Ψ, the probability on the righthand side would be 0, whereas $Prob(s, \Phi\ \mathcal{U}^{[t,t']}\ \Psi) = 1$.

Theorem 3. *For any CTMC $\mathcal{M}$ and $0 < t \leq t'$:*

$$\begin{aligned} &Prob^{\mathcal{M}}(s, \Phi\ \mathcal{U}^{[t,t']}\ \Psi) \\ &= \sum_{s' \models \Phi} \sum_{s'' \models \Psi} \pi^{\mathcal{M}[\neg\Phi]}(s, s', t) \cdot \pi^{\mathcal{M}[\neg\Phi\vee\Psi]}(s', s'', t' - t). \end{aligned}$$

Proof. Let $\varphi = \Phi\ \mathcal{U}^{[t,t']}\ \Psi$ with $0 < t \leq t'$. For any path σ with $\sigma \models \varphi$, it follows directly from the semantics of $\mathcal{U}^{[t,t']}$ that Φ continuously holds in the interval $[0,t)$ (i.e., $\sigma \models \Box^{[0,t)}\ \Phi$), in particular, $\sigma@t = s' \in Sat(\Phi)$. Let $\sigma' \in Path(s')$ be the suffix of σ that starts at time t, i.e., σ' is the unique path with $\sigma'@x = \sigma@(t + x)$ for any positive real x. If $\sigma \models \varphi$, the suffix $\sigma' \models \Phi\ \mathcal{U}^{[0,t'-t]}\ \Psi$. Let

$$\Sigma(s') = \{\sigma \in Path^{\mathcal{M}}(s) \mid \sigma@t = s' \wedge \sigma \models \varphi\}$$

be the set of paths that start in state s, that satisfy φ, and that pass the intermediate state $\sigma@t = \sigma'@0 = s'$. Then,

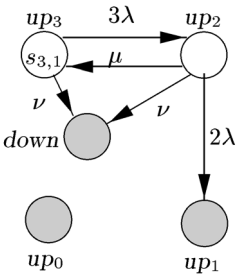


Fig. 4. The CTMC $\mathcal{M}[\neg\Phi]$.

$$\Pr_s(\Sigma(s')) = Prob^{\mathcal{M}}(s, \Phi\, \mathcal{U}^{[t,t]} at_{s'}) \cdot Prob(s', \Phi\, \mathcal{U}^{[0,t'-t]}\ \Psi).$$

As the sets $\Sigma(s')$ for $s' \in Sat(\Phi)$ are pairwise disjoint, we obtain:

$$Prob^{\mathcal{M}}(s, \Phi\, \mathcal{U}^{[t,t']}\ \Psi) = \sum_{s' \models \Phi} Prob^{\mathcal{M}}(s, \Phi\, \mathcal{U}^{[t,t]} at_{s'}) \cdot Prob^{\mathcal{M}}(s', \Phi\, \mathcal{U}^{[0,t'-t]}\ \Psi).$$

Applying Proposition 5 to the left factor and Theorem 2 to the right factor yields:

$$\sum_{s' \models \Phi} Prob^{\mathcal{M}[\neg\Phi]}(s, \diamond^{[t,t]} at_{s'}) \cdot Prob^{\mathcal{M}[\Psi]}(s', \Phi\, \mathcal{U}^{[0,t'-t]}\ \Psi).$$

Rewriting these probabilities to transient state probabilities—again using Proposition 5 and Theorem 2—finally results in:

$$Prob^{\mathcal{M}}(s, \Phi\, \mathcal{U}^{[t,t']}\ \Psi) = \sum_{s' \models \Phi} \left(\pi^{\mathcal{M}[\neg\Phi]}(s, s', t) \cdot \sum_{s'' \models \Psi} \pi^{\mathcal{M}[\neg\Phi \vee \Psi]}(s', s'', t'-t) \right).$$

□

Example 5. Consider the model of the TMR system with initial distribution $\alpha = \alpha^1_{s_{3,1}}$ and let $\Phi' = \mathcal{P}_{\geq 0.05}(\Phi\, \mathcal{U}^{[3,7]}\ \Psi)$ for $\Phi = up_3 \vee up_2$ and $\Psi = up_2 \vee up_1$ and $\varphi = \Phi\, \mathcal{U}^{[3,7]}\ \Psi$. Note that Theorem 3 indicates to compute the probability $Prob(s_{3,1}, \varphi)$ by considering two different CTMCs: $\mathcal{M}[\neg\Phi]$ and $\mathcal{M}[\neg\Phi \vee \Psi]$. According to Theorem 3, verifying Φ' boils down to first computing the transient probabilities at time 3, i.e., $\alpha' = \underline{\pi}^{\mathcal{M}_1}(s_{3,1}, 3)$ for all states in CTMC $\mathcal{M}_1$ of Fig. 4, where all $\neg\Phi$ states (indicated in gray) of the original model are made absorbing. We obtain $\alpha' = (0.968, 0.0272, 0.011, 0, 0.003)$ with a precision of $\varepsilon = 10^{-6}$ for $\lambda = 0.01$, $\nu = 0.001$, $\mu = 1.0$, and $\delta = 0.2$. In the second phase, Theorem 3 suggests computing the transient probabilities at time 4 in CTMC $\mathcal{M}_2$ of Fig. 3c starting from distribution α'. This yields $\sum_{s'' \models up_2} \pi^{\mathcal{M}_2}(\alpha', s'', 4) \approx 0.1365$. Thus, $s_{3,1} \models \Phi'$.

Corollary 2. *For CTMC $\mathcal{M}$:*

$$Prob^{\mathcal{M}}(s, \Phi\, \mathcal{U}^{[t,t]}\ \Psi) = \sum_{s' \models \Phi \wedge \Psi} \pi^{\mathcal{M}[\neg\Phi]}(s, s', t).$$

Proof. Directly from Theorem 3. □

Note that (3) is a simplified version of this corollary.

4.3 Uniformisation

In the previous sections, we have shown that the calculation of $Prob(s, \Phi\, \mathcal{U}^I \Psi)$ boils down to instances of transient analysis on CTMCs. Formally, transient state probabilities are obtained as a solution to the Chapman-Kolmogorov differential equations (cf. Section 2.4) and are given by the Taylor-MacLaurin series:

$$\underline{\pi}(\alpha, t) = \alpha \cdot e^{\mathbf{Q} \cdot t} = \alpha \cdot \sum_{i=0}^{\infty} \frac{(\mathbf{Q} \cdot t)^I}{i!} \quad \text{with } \mathbf{Q} = \mathbf{R} - diag(\underline{E}), \tag{6}$$

where we recall that $\underline{\pi}(\alpha, t)$ denotes the vector of state probabilities at time t. This characterization is not attractive for a numerical algorithm since [58], [69] 1) it suffers from numerical instability as $\mathbf{Q}$ contains both positive and negative entries and 2) it is difficult to find a proper truncation criterion for the infinite summation. Therefore, other algorithms to compute transient state probabilities for CTMCs have been developed of which uniformization [36], [37], [47] is currently regarded as the most attractive. Under special conditions, e.g., when the rates in $\mathbf{R}$ differ a large number of magnitudes, Runge-Kutta-like methods might perform better, see [66], [67]. For the sake of completeness, we briefly discuss the main aspects of uniformization here. These details will play a significant role in discussing the efficiency of our model-checking algorithms, cf. Section 6.

For CTMC $\mathcal{M} = (S, \mathbf{R}, L)$, the uniformized DTMC is $(S, \mathbf{U}, L)$, where $\mathbf{U}$ is defined by $\mathbf{U} = I + \mathbf{Q}/q$ with $q \geq \max_i\{E(s_i)\}$. The *uniformization rate* q can be chosen to be any value exceeding the shortest mean residence time. All rates in the CTMC are normalized with respect to q. For each state s with $E(s) = q$, one epoch in the uniformized DTMC corresponds to a single exponentially distributed delay with rate q, after which one of its successor states is selected probabilistically. As a result, such states have no additional self loop in the DTMC. If $E(s) < q$, i.e., state s has, on average, a longer state residence time than $\frac{1}{q}$, one epoch in the DTMC might not be "long enough"; hence, in the next epoch, these states might be revisited with some positive probability. This is represented by equipping these states with a self loop with probability $1 - \frac{E(s)}{q} + \frac{\mathbf{R}(s,s)}{q}$.

Transient state probabilities are now computed as follows: Substituting $\mathbf{Q} = q \cdot (\mathbf{U} - \mathbf{I})$ in (6) yields:

$$\underline{\pi}(\alpha, t) = \alpha \cdot \sum_{i=0}^{\infty} e^{-q \cdot t} \frac{(q \cdot t)^I}{i!} \mathbf{U}^i = \sum_{i=0}^{\infty} e^{-q \cdot t} \frac{(q \cdot t)^I}{i!} \underline{\pi}_i, \tag{7}$$

where $e^{-q \cdot t} \cdot ((q \cdot t)^i / i!)$ is the ith Poisson probability with parameter $q \cdot t$ and $\underline{\pi}_i$ is the state probability distribution vector after i epochs in the DTMC with transition matrix $\mathbf{U}$ determined recursively by $\underline{\pi}_i = \underline{\pi}_{i-1} \cdot \mathbf{U}$ with $\underline{\pi}_0 = \alpha$. The Poisson probabilities can be computed in a stable way with the Fox-Glynn algorithm [35]. The number of terms k_ϵ in (7) to be taken given a required accuracy ϵ is the smallest value satisfying:

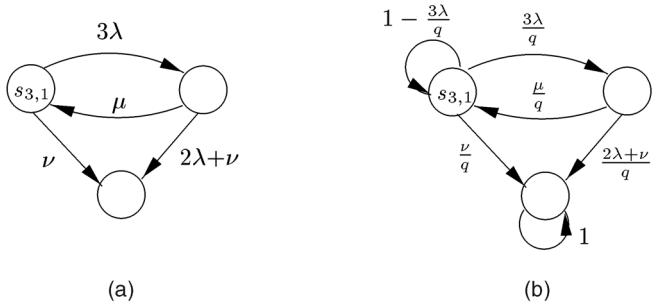


Fig. 5. (a) A CTMC and (b) the corresponding uniformized DTMC.

$$\sum_{n=0}^{k_\epsilon} \frac{(q{\cdot}t)^n}{n!} \geq \frac{1-\epsilon}{e^{-q\cdot t}} = (1-\epsilon){\cdot}e^{q\cdot t}. \tag{8}$$

If the product $q{\cdot}t$ is large, k_ϵ tends to be of order $O(q{\cdot}t)$. On the other hand, if $q{\cdot}t$ is large, the DTMC described by $\mathbf{U}$ might have reached steady state along the way. Such an "on-the-fly" steady-state detection can be integrated in the computational procedure, see [59]. This steady-state truncation point is often smaller than k_ϵ, making the trailing matrix-vector multiplications superfluous. For further details, see [39], [69].

Example 6. Consider the TMR example and $\Phi = \mathcal{P}_{\geq 0.99}(\Box^{[0,3]}(up_3 \vee up_2))$. As explained in Example 5, it suffices to verify Φ on the CTMC of Fig. 4. This CTMC is equivalent to the CTMC depicted in Fig. 5a, as will be justified in the next section. Assume $\mu > \lambda$. Checking Φ reduces to a transient analysis of the uniformized DTMC depicted in Fig. 5b, where $q = 2\lambda{+}\nu{+}\mu$. With a required accuracy $\epsilon = 10^{-5}$ and $\lambda = 0.01$, $\nu = 0.001$, and $\mu = 1.0$, we obtain (with five digits of precision) $q = 1.021$, $k_\epsilon = 13$, and $\underline{\pi}(s_{3,1}, 3) = (0.96856, 0.02724, 0.00148)$, where the last value corresponds to the state probability of the absorbing state. Summing the first two probabilities yields $s_{3,1} \models \Phi$.

5 Abstraction with Bisimulation

In this section, we discuss a technique to aggregate the state space of a CTMC. This technique is based on the observation that a slight variant of bisimulation, also known as lumping on CTMCs, preserves all CSL formulas. The result presented below is similar to that for relating bisimulation and CTL (and CTL* equivalence) [17] and probabilistic bisimulation and PCTL [8].

5.1 Bisimulation (Lumping) Equivalence

Lumpability is an important notion on CTMCs that allows their aggregation without affecting performance properties [18], [49]. We adapt the standard notion in order to deal with CTMCs with state labelings. Rather than considering a state labeling with atomic propositions, it is convenient for our purposes to consider labelings with more general sets of CSL formulas. Let $\mathcal{M} = (S, \mathbf{R}, L)$ be a CTMC, F a set of CSL formulas, and $L_F : S \to 2^F$ a labeling defined by $L_F(s) = \{\Phi \in F \mid s \models \Phi\}$.

Definition 8. *An F* bisimulation *on $\mathcal{M} = (S, \mathbf{R}, L)$ is an equivalence R on S such that, whenever $(s, s') \in R$, then*

$$L_F(s) = L_F(s') \text{ and } \mathbf{R}(s, C) = \mathbf{R}(s', C) \text{ for all } C \in S/R,$$

where S/R denotes the quotient space under R and $\mathbf{R}(s, C) = \sum_{s' \in C} \mathbf{R}(s, s')$. States s and s' are F bisimilar *iff there exists an F bisimulation R that contains (s, s').*

Thus, any two bisimilar states are equally labeled (with respect to F) and the cumulative rate of moving from these states to any equivalence class C is equal. The notion of F bisimulation is a slight variant of lumping equivalence [18] and Markovian bisimulation [19], [44]. For $s \in S$, let $[s]_R$ denote the equivalence class of s under R. For $\mathcal{M} = (S, \mathbf{R}, L)$, the CTMC $\mathcal{M}/R$ is defined by $\mathcal{M}/R = (S/R, \mathbf{R}_R, L_R)$ with $\mathbf{R}_R([s]_R, C) = \mathbf{R}(s, C)$ and $L_R([s]_R) = L_F(s)$. The performance measures of $\mathcal{M}$ and $\mathcal{M}/R$ are strongly related. The transient state probability of the lumped CTMC $\mathcal{M}/R$ being in state $[s]_R$ at time t given initial distribution α_R equals [18], [49]:

$$\pi^{\mathcal{M}/R}(\alpha_R, [s]_R, t) = \sum_{s' \in [s]_R} \pi^{\mathcal{M}}(\alpha, s', t),$$

where $\alpha_R([s]_R) = \sum_{s' \in [s]_R} \alpha(s')$. The same correspondence holds for steady-state probabilities.

Example 7. The reflexive, symmetric, and transitive closure of the relation

$$R = \{(0111, 1011), (1011, 1101), (0011, 0101), (0101, 1001)\}$$

is an AP bisimulation on the set of states of the CTMC $\mathcal{M}$ depicted in Fig. 6. For convenience, double arrows are used to indicate that there exists a transition from a state to another state and vice versa. The lumped CTMC $\mathcal{M}/R$ consists of five aggregated states: the singleton states $[1111]_R$, $[0000]_R$, and $[0001]_R$ and the states $[0111]_R = \{0111, 1011, 1101\}$ and $[0011]_R = \{0011, 0101, 1001\}$. In fact, this yields the CTMC of the TMR system of Example 1.

5.2 Bisimulation and CSL Equivalence

Let CSL_F denote the smallest set of CSL formulas that includes F and that is closed under all CSL operators. In the following, we write $\models_{\mathcal{M}}$ for the satisfaction relation $\models$ (on CSL) on $\mathcal{M}$.

Theorem 4. *Let R be an F bisimulation on $\mathcal{M}$ and s a state in $\mathcal{M}$. Then,*

1. *For all CSL_F formulas Φ, $s \models_{\mathcal{M}} \Phi$ iff $[s]_R \models_{\mathcal{M}/R} \Phi$.*
2. *For all CSL_F path formulas φ,*

$$Prob^{\mathcal{M}}(s, \varphi) = Prob^{\mathcal{M}/R}([s]_R, \varphi).$$

In particular, F-bisimilar states satisfy the same CSL_F formulas.

Proof. Straightforward by structural induction on Φ and φ.□

As the verification of $\mathcal{S}$ formulae amounts to carrying out a steady-state analysis (after a graph analysis) and the verification of $\mathcal{P}$ formulae boils down to a transient-state analysis (on a transformed CTMC), this result is not surprising given that bisimulation—ordinary lumping

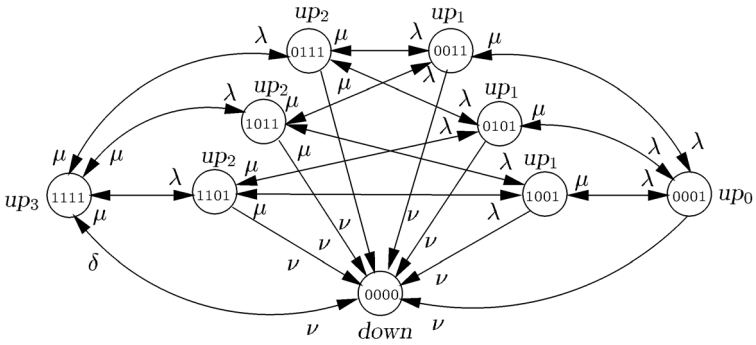


Fig. 6. A detailed version of the TMR model.

equivalence—preserves steady-state and transient-state probabilities. Theorem 4 allows us to verify CSL formulas on the possibly much smaller $\mathcal{M}/R$ rather than on $\mathcal{M}$, for AP bisimulation R.

Theorem 5. *$s \models_{\mathcal{M}} \Phi$ iff $s' \models_{\mathcal{M}} \Phi$ for all CSL formulas iff s and s' are AP bisimilar.*

Proof. The "if" part of the proof follows directly from Theorem 4. The "only if" part was recently shown in [30].□

We can exploit the above result and apply it to the transformations of the previous section by using the following observation: From 1 of Theorem 4, it follows:

$$\sum_{s' \models_{\mathcal{M}} \Phi} \pi^{\mathcal{M}}(s, s', t) = \sum_{S' \models_{\mathcal{M}/R} \Phi} \pi^{\mathcal{M}/R}([s]_R, S', t) \qquad (9)$$

for any CSL_F formula Φ and F bisimulation R. This observation allows us to simplify the CTMCs $\mathcal{M}[\ldots]$ that occur in Cases A-D of the model checking procedure presented in Section 4 in the following way. For Cases B and D, we compute the transient probabilities for Ψ states in the CTMC $\mathcal{M}' = \mathcal{M}[\neg\Phi \vee \Psi]$. Let $F = \{\neg\Phi \wedge \neg\Psi, \Psi\}$ and R be the smallest equivalence on the state space S of $\mathcal{M}'$ that identifies all Ψ states and all $(\neg\Phi \wedge \neg\Psi)$ states. Clearly, R is an F bisimulation on $\mathcal{M}'$. The state space of $\mathcal{M}'/R$ is

$$S/R = Sat(\Phi \wedge \neg\Psi) \cup [Sat(\Psi)]_R \cup [Sat(\neg\Phi \wedge \neg\Psi)]_R. \qquad (10)$$

Since Ψ is a CSL_F formula, (9) yields

$$\sum_{s'' \models \Psi} \pi^{\mathcal{M}'}(s, s'', t) = \pi^{\mathcal{M}'/R}(s, [Sat(\Psi)]_R, t)$$

for any state $s \in Sat(\Phi \wedge \neg\Psi)$. Similar arguments are applicable to Cases A and C. As a result, the sets $[Sat(\neg\Phi \wedge \neg\Psi)]_R$ and $[Sat(\Psi)]_R$ in Cases A-D can be considered as single states. This may yield a substantial reduction of the state space of the CTMC under consideration; more precisely, the resulting state space S/R has size $|Sat(\Phi \wedge \neg\Psi)| + 2$. From a computational point of view, the switch from $\mathcal{M}$ to the modified $\mathcal{M}[\ldots]/R$ is quite simple as we just collapse certain states into a single absorbing state. The rate matrix for $\mathcal{M}[\ldots]/R$ can be obtained by simple manipulations of the rate matrix $\mathbf{R}$ for $\mathcal{M}$.

Example 8. According to the above observations, in the CTMC of Fig. 4, we may aggregate states $[Sat(\Phi)]_R = \{s_{0,1}, s_{0,0}, s_{1,1}\}$ into a single state using a $\{\Phi, \Psi\}$ bisimulation, where Φ and Ψ are given in Example 4. This new state is reachable from $s_{3,1}$ with rate ν and from $s_{2,1}$ with rate $2\lambda+\nu$. This yields the CTMC depicted in Fig. 5a. As a second example, in the CTMC of Fig. 5c, we may collapse $[Sat(\Psi)]_R = \{s_{2,1}, s_{1,1}\}$ and $[Sat(\neg\Phi \wedge \neg\Psi)]_R = \{s_{0,0}, s_{0,1}\}$ into single states using a $\{\neg\Phi, \Psi\}$ bisimulation.

Note that the bisimulations constructed in this way have at most two nontrivial classes. We can exploit the above theorems further by aggregating the model $\mathcal{M}$ as far as possible during model checking or prior to model checking. For the latter purpose, we consider the coarsest AP bisimulation R on $\mathcal{M}$ and construct the quotient Markov model $\mathcal{M}/R$ prior to model checking $\mathcal{M}$. R can be computed by a modification of the standard partition refinement algorithm [29]. It now follows from Theorem 5 that any CSL formula can be equally well checked on $\mathcal{M}/R$ instead of on $\mathcal{M}$.

Furthermore, we can add a formula-specific aggregation. Let $AP(\Phi)$ denote the set of atomic propositions occurring in Φ. Note that Φ belongs to $\mathrm{CSL}_{AP(\Phi)}$. Due to Theorem 4, Φ can be model checked on $\mathcal{M}/R'$ instead of on $\mathcal{M}$, where R' is the coarsest $AP(\Phi)$ bisimulation on $\mathcal{M}$. R' can again be computed by partition refinement.

6 Efficiency Considerations

In this section, we summarize the results of the previous sections and discuss the space and time complexity of model checking CSL as well as some implementation considerations.

Let $\mathcal{M} = (S, \mathbf{R}, L)$ be a CTMC, M the number of nonzero entries in the rate matrix $\mathbf{R}$, and $N = |S|$ the number of states in $\mathcal{M}$. For a fully connected CTMC $\mathcal{M}$, we thus have $M = N^2$, but, in practice, we often find $M < kN$ for a small constant k. Without loss of generality, we assume $M \geq N$. Consider CSL state formula Φ. The general strategy is identical to the model checking procedure for CTL [23]. The set $Sat(\Phi)$ of states satisfying Φ is computed in an iterative way, starting with considering the subformulas of Φ of length 1, i.e., the atomic propositions in Φ. These subformulas correspond to the leaves in the parse tree of Φ. In the $(i+1)$th iteration of the algorithm, subformulas of length $i+1$ are considered using the results of all subformulas of length at most i, i.e., the results of all

subnodes in the parse tree. This computation continues until the formula Φ of length $|\Phi|$, i.e., the root of the parse tree, is considered. We consider the required computations for each kind of subformula (node). The computation for nodes in the parse tree of the form tt, $\neg\Phi$, or $\Phi \wedge \Psi$ is straightforward and takes $\mathcal{O}(N)$ time.

Steady-state operator. First, a graph analysis is carried out to determine the BSCCs of $\mathcal{M}$. This takes $\mathcal{O}(N+M)$ time [70]. In the worst case, for each identified BSCC B, a linear system of $|B|$ equations needs to be solved once. Ranging over all BSCCs, this leads to at most N equations since each state belongs to at most one BSCC. Finally, the probability of reaching a BSCC B needs to be computed for each BSCC. This requires solving a linear system of N equations, taking $\mathcal{O}(N^3)$ (or better) time [1]. In practice, it is often preferred to use iterative, numerical methods such as Power, Gauss-Seidel, or similar [69] for large N. Then, convergence depends on the structure of the equation system and is in principle not guaranteed.

Next formulas. The nodes that represent formulas of the form $\mathcal{P}_{\trianglelefteq p}(X^I\Phi)$ require $\mathcal{O}(M)$ time as $\mathcal{O}(M)$ scalar multiplications and additions are needed to perform the matrix-vector multiplication with $\mathbf{P}$ given a suitably chosen sparse matrix storage structure. The same applies for the unbounded next operator.

Unbounded until. Formulas of the form $\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}\,\Psi)$ require the solution of a linear system of N equations, taking $\mathcal{O}(N^3)$ time. The special case $\mathcal{P}_{>0}(\Phi\,\mathcal{U}\,\Psi)$ and $\mathcal{P}_{\geq 1}(\Phi\,\mathcal{U}\,\Psi)$ can be treated in $\mathcal{O}(N)$ time [38].

Time-bounded until. For formulas of the form $\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}^I\,\Psi)$, we distinguish two cases: $I=[0,t']$ and $I=[t,t']$ with $0<t\leq t'$. To model check the formula $\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}^{[0,t']}\,\Psi)$, the CTMC $\mathcal{M}$ is transformed into $\mathcal{M}[\neg\Phi\vee\Psi]$ (cf. Theorem 2) and transient analysis is carried out on $\mathcal{M}[\neg\Phi\vee\Psi]$ to compute

$$Prob(s,\Phi\,\mathcal{U}^{[0,t]}\,\Psi)=\sum_{s''\models\Psi}\pi^{\mathcal{M}[\neg\Phi\vee\Psi]}(s,s'',t').$$

The transformation takes $\mathcal{O}(M)$ time. To compute $\underline{\pi}(s,t')$ on $\mathcal{M}[\neg\Phi\vee\Psi]$ using uniformization, the sum of $\mathcal{O}(q'\cdot t')$ vectors is required, each of which is the result of a matrix-vector multiplication. Here, q' is the uniformization rate of $\mathcal{M}[\neg\Phi\vee\Psi]$. Given a sparse implementation of the latter, we require $\mathcal{O}(M)$ multiplications and additions for the matrix-vector product so that the overall computational complexity of computing $\underline{\pi}(s,t')$ is $\mathcal{O}(M\cdot q'\cdot t')$. A naive approach to model check $\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}^{[0,t']}\,\Psi)$ requires to perform this procedure for each state s as suggested in [12]. An improvement suggested in [48] cumulates the entire vector $\underline{Prob}(\Phi\,\mathcal{U}^{[0,t]}\,\Psi)$ for all states simultaneously, yielding a time complexity of $\mathcal{O}(M\cdot q'\cdot t')$.

For formulas of the form $\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}^{[t,t']}\,\Psi)$ with $0<t\leq t'$, the computation is split into two parts, according to Theorem 3. This means that transient analysis is needed two times (for t and for $t'-t$) on different transformed Markov chains, $\mathcal{M}[\neg\Phi]$ and $\mathcal{M}[\neg\Phi\vee\Psi]$. Each transformation takes $\mathcal{O}(M)$ time. The effort needed to carry out uniformization on either chain can be quantified as follows. Even though the uniformization rates in these two chains may differ, we can use the uniformization rate q of $\mathcal{M}$ as an upper bound for them and, hence, each transient analysis has a time complexity of $\mathcal{O}(M\cdot q\cdot t')$. Note that $t'-t\leq t'\geq t$. The method of [48] can be generalized such that two transient analyses suffice to compute the required probabilities for each s. In summary, we obtain that model checking the time bounded-until operator takes $\mathcal{O}(M\cdot q\cdot t')$ time.

Besides, the special case $\mathcal{P}_{>0}(\Phi\,\mathcal{U}^I\,\Psi)$, for nonempty I, can be treated in the same way as the CTL formula $\exists(\Phi\,\mathcal{U}\,\Psi)$. This yields a worst-case time complexity of $\mathcal{O}(N)$ for this case, cf. [23]. The timing constraint I is not relevant here: If there exists a path in $\mathcal{M}$ satisfying $\Phi\,\mathcal{U}\,\Psi$, then, with some positive probability, a Ψ state can be reached for some $t\in I$. Note, however, that $\mathcal{P}_{\geq 1}(\Phi\,\mathcal{U}^I\,\Psi)$ cannot be treated as the CTL formula $\forall(\Phi\,\mathcal{U}\,\Psi)$.

A few efficiency improvements are possible. First, for large t', the number of computation steps needed in practice might be much smaller than the above bound when using on-the-fly steady-state detection [59]. Furthermore, the uniformization rate is in practice determined chain specifically, i.e., after transformation. Also, it is favorable to reduce the size of the state spaces and, hence, of the probability vectors to be computed. For the $[0,t']$ case, for instance, we can exploit the fact that the CTMC $\mathcal{M}[\neg\Phi]$ can be aggregated to $\mathcal{M}[\neg\Phi]/R$ for R a $\{\neg\Phi\wedge\neg\Psi,\Psi\}$ bisimulation.

Bisimulation aggregation. Orthogonal to the model checking algorithm, we have the means to interweave abstraction steps whenever appropriate during model checking $\mathcal{M}$. A priori, we can compute the best possible formula-independent lumping by constructing the coarsest possible AP bisimulation R and considering the quotient $\mathcal{M}/R$ instead of $\mathcal{M}$. The computation of R and $\mathcal{M}/R$ takes $\mathcal{O}(M\log N)$ time, using an adapted version of the algorithm in [29]. With the same computational effort, we can also compute a formula-dependent quotient $\mathcal{M}/R'$ for R' the coarsest $AP(\Psi)$ bisimulation, in order to reduce the number of computation steps required for subsequent model checking of subformula Ψ.

Summary. The results for each operator are collected in Table 2, where the complexity results are based on a sparse storage structure for the rate and transition matrix and where Gaussian elimination is used for solving linear equation systems and uniformization is used for transient analysis. Cumulating over all nodes in the parse tree, i.e., all subformulas of Φ, we obtain, for the worst-case time complexity of model checking CSL:

$$\mathcal{O}(|\Phi|\cdot(M\cdot q\cdot t_{max}+N^3)),$$

where t_{max} is the maximum time bound of the time-bounded until subformulas occurring in Φ. Recall that q is the uniformization rate, which can be chosen as the maximum entry of $\underline{E}$. If we make the practically often justified assumption that $M<kN$ for a constant k, then the space complexity is linear in N using a sparse matrix data structure.

Theorem 6. *For $\mathcal{M}$, a finite state CTMC and CSL state formula Φ, the time and space complexity of the model checking algorithm described in Section 4 is polynomial in the size of $\mathcal{M}$ and linear in the length of the formula Φ.*

TABLE 2
Algorithms for Model Checking CSL and Their (Worst Case) Time Complexity

operator	algorithm(s)	time complexity
$\mathcal{S}_{\trianglelefteq p}(\Phi)$	BSCC detection + steady-state analysis per BSCC computation $Prob(s, \Diamond at_B)$ per BSCC	$\mathcal{O}(N^3)$
$\mathcal{P}_{\trianglelefteq p}(X\,\Phi)$	matrix-vector multiplication	$\mathcal{O}(M)$
$\mathcal{P}_{\trianglelefteq p}(X^I\,\Phi)$	matrix-vector multiplication	$\mathcal{O}(M)$
$\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}\,\Psi)$	solving linear equation system	$\mathcal{O}(N^3)$
$\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}^I\,\Psi)$	matrix-vector multiplication + transient analysis	$\mathcal{O}(M{\cdot}q{\cdot}t')$

7 Related Work

Model-checking probabilistic systems. Early work has concentrated on discrete-time models. Methods to verify a DTMC or a Markov decision process against a linear-time temporal logic (LTL) formula (or a Büchi automaton) have been considered, e.g., [28], [62], [71]. The basis of these works is the nontrivial reduction of the model-checking problem to the computation of the probabilities to reach certain sets of states (mostly, BSCCs). Courcoubetis and Yannakakis [27] describe an algorithm for checking whether a DTMC satisfies an LTL formula.

As stated in the introduction, PCTL model checking has been brought up by Hansson and Jonsson [38]. For the CSL operators that do not refer to the real-time behavior of the CTMC, the PCTL algorithms can be exploited. The logic PCTL* contains both LTL and PCTL. Its verification is studied in [8], [15], [16]. Its basic idea is the reduction to the verification of quantitative LTL properties.

Branching-time model checking of Markov decision processes is considered in [4], [14], [15], [16]. Here, nondeterminism is resolved by adversaries. The model checking of until formulas reduces to the computation of a minimum (or maximum) probability, depending whether one quantifies over all or some adversaries, respectively.

Model-checking real-time probabilistic systems. A qualitative model-checking algorithm for a continuous probabilistic variant of timed automata has been proposed in [7]. This technique is based on regions, finite partitions of the infinite continuous-time domain tailored to the property and model under consideration. Recently, this approach has been adopted for quantitative model checking [54].

Model-checking continuous-time Markov chains. A stochastic extension of CTL, also called CSL, was initially proposed in [9]. Using transcendental number theory, the elementary result that the model-checking problem for CSL is decidable for rational time bounds is proven. No concrete algorithms were provided, though.

In [11], we extended CSL with the steady-state operator presented here to reason about the stationary behavior of CTMCs. The first work on logics and model-checking algorithms for studying the stationary behavior of stochastic systems, in particular semi-Markov decision processes, has been reported in [4], [5]. Semi-Markov decision processes extend CTMCs with nondeterminism and non-exponential distributions. Apart from the fact that we are considering a more specific model, our approach differs in several aspects. To enable the specification of long-run average properties, [4], [5] uses experiments, automata that are intended to be traversed infinitely often. Experiments are used to either measure the probability with which an LTL formula holds or to measure the expected time to reach a given set of goal states. In contrast, steady-state properties are first-class citizens of CSL—they can be combined arbitrarily with other operators—whereas experiments can only occur as top-level "operator."

Another related approach is that of [60]. Here, automata are used to define path-based stochastic variables on a Markov model described as a stochastic activity network [57]. Analysis takes place by considering a synchronous product of the model and the specification automaton. Other recent works on CSL are the extension to continuous-space Markov processes [30], the use of discrete-event simulation and hypothesis testing [72], and the use of Kronecker algebra to exploit the structure of the CTMC [20].

8 Concluding Remarks

This paper proposed the use of the temporal logic CSL to specify performance and reliability measures for CTMCs and introduced automated verification algorithms. This yields:

- a flexible means to specify standard and complex measures succinctly,
- automated means to analyze these measures over CTMCs, and
- automated measure-driven aggregation (lumping) of CTMCs.

The automated verification hides specialized algorithms from the performance engineer.

The following algorithms are used: Next and (unbounded) until formulas can be treated using matrix-vector multiplication and solving a system of linear equations like in [38]. Checking steady-state properties amounts to solving a system of linear equations combined with standard graph analysis methods. We showed that checking the time-bounded until operator can be reduced to the problem of computing transient state probabilities for CTMCs. This allows us to adopt efficient and numerically stable

techniques for model-checking CTMCs. The time and space complexity of our model-checking algorithms is polynomial in the size of the model and linear in the length of the formula. A prototype implementation and experimental results have been reported in [43].

In addition, we showed that AP bisimulation preserves the validity of all CSL formulas. This allows us to switch from the original state space to the (possibly much smaller) quotient space under AP bisimulation prior to carrying out the model checking.

Acknowledgments

The authors would like to thank Joachim Meyer-Kayser and Markus Siegle (both from the University of Erlangen-Nürnberg) for valuable discussions. H. Hermanns is supported by the Netherlands Organization for Scientific Research (NWO) and J.-P. Katoen is partially supported by the Dutch Technology Foundation (STW). The cooperation between the research groups in Aachen, Bonn, Erlangen-Nürnberg, and Twente takes place as part of the Validation of Stochastic Systems (VOSS) project, funded by the NWO and the German Research Council DFG. This work was performed while B. Haverkort was at RWTH Aachen, Germany.

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullmann, *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[2] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modeling with Generalized Stochastic Petri Nets.* John Wiley & Sons, 1995.

[3] L. de Alfaro, "Formal Verification of Probabilistic Systems," PhD dissertation, Stanford Univ., 1997.

[4] L. de Alfaro, "Temporal Logics for the Specification of Performance and Reliability," *Proc. Fourth Ann. Symp. Theoretical Aspects of Computer Science,* 1997.

[5] L. de Alfaro, "How to Specify and Verify the Long-Run Average Behavior of Probabilistic Systems," *Proc. IEEE 13th Symp. Logic in Computer Science,* pp. 174-183, 1998.

[6] R. Alur and D. Dill, "A Theory of Timed Automata," *Theoretical Computer Science,* vol. 126, pp. 183-235, 1994.

[7] R. Alur, C. Courcoubetis, and D. Dill, "Model-Checking for Probabilistic Real-Time Systems," *Automata, Languages, and Programming,* 1991.

[8] A. Aziz, V. Singhal, F. Balarin, R. Brayton, and A. Sangiovanni-Vincentelli, "It Usually Works: The Temporal Logic of Stochastic Systems," *Computer-Aided Verification,* 1995.

[9] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, "Model Checking Continuous Time Markov Chains," *ACM Trans. Computational Logic,* vol. 1, no. 1, pp. 162-170, 2000.

[10] C. Baier, "On Algorithmic Verification Methods for Probabilistic Systems," habilitation thesis, Univ. of Mannheim, Germany, 1999, avaliable at web.informatik.uni-bonn.de/I/papers/haupt.ps.

[11] C. Baier, J.-P. Katoen, and H. Hermanns, "Approximate Symbolic Model Checking of Continuous-Time Markov Chains," *Concurrency Theory,* 1999.

[12] C. Baier, B.R. Haverkort, and H. Hermanns, J.-P. Katoen, "Model Checking Continuous-Time Markov Chains by Transient Analysis," *Computer Aided Verification,* 2000.

[13] C. Baier, B.R. Haverkort, H. Hermanns, and J.-P. Katoen, "On the Logical Characterisation of Performability Properties," *Automata, Languages, and Programming,* 2000.

[14] C. Baier and M. Kwiatkowska, "On the Verification of Qualitative Properties of Probabilistic Processes Under Fairness Constraints," *Information Processing Letters,* vol. 66, no. 2, pp. 71-79, 1998.

[15] C. Baier and M.Z. Kwiatkowska, "Model Checking for a Probabilistic Branching Time Logic with Fairness," *Distributed Computing,* vol. 11, pp. 125-155, 1998.

[16] A. Bianco and L. de Alfaro, "Model Checking of Probabilistic and Nondeterministic Systems," *Foundations of Software Technology and Theoretical Computer Science,* 1995.

[17] M. Brown, E. Clarke, and O. Grumberg, "Characterizing Finite Kripke Structures in Propositional Temporal Logic," *Theoretical Computer Science,* vol. 59, pp. 115-131, 1988.

[18] P. Buchholz, "Exact and Ordinary Lumpability in Finite Markov Chains," *J. Applied Probability,* vol. 31, pp. 59-75, 1994.

[19] P. Buchholz, "Markovian Process Algebra," Technical Report 500, Fachbereich Informatik, Univ. of Dortmund, 1994.

[20] P. Buchholz, J.-P. Katoen, P. Kemper, and C. Tepper, "Model-Checking Large Structured Markov Chains," *J. Logic and Algebraic Programming,* to appear.

[21] G. Ciardo, J.K. Muppala, and K.S. Trivedi, "SPNP: Stochastic Petri Net Package," *Proc. Third Int'l Workshop Petri Nets and Performance Models,* pp. 142-151, 1989.

[22] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, "NuSMV: A New Symbolic Model Checker," *J. Software Tools for Technology Transfer,* vol. 2, pp. 410-425, 2000.

[23] E. Clarke, E. Emerson, and A. Sistla, "Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications," *ACM Trans. Programming Languages and Systems,* vol. 8, pp. 244-263, 1986.

[24] E. Clarke, M. Fujita, P.C. McGeer, and J.C-Y. Yang, "Multi-Terminal Binary Decision Diagrams: An Efficient Data Structure for Matrix Representation," *Formal Methods in System Design,* vol. 10, nos. 2/3, pp. 149-169, 1997.

[25] E. Clarke, O. Grumberg, and D. Peled, *Model Checking.* MIT Press, 1999.

[26] A.E. Conway and N.D. Georganas, *Queueing Networks: Exact Computational Algorithms.* MIT Press, 1989.

[27] C. Courcoubetis and M. Yannakakis, "Verifying Temporal Properties of Finite-State Probabilistic Programs," *Proc. Ann. Symp. Foundations of Computer Science,* pp. 338-345, 1988.

[28] C. Courcoubetis and M. Yannakakis, "The Complexity of Probabilistic Verification," *J. ACM,* vol. 42, no. 4, pp. 857-907, 1995.

[29] S. Derisavi, H. Hermanns, and W.H. Sanders, "Optimal State-Space Lumping in Markov Chains," *Information Processing Letters,* to appear.

[30] J. Desharnais and P. Panangaden, "Continuous Stochastic Logic Characterizes Bisimulation of Continuous-Time Markov Processes," *J. Logic and Algebraic Programming,* to appear.

[31] D.L. Dill, "The Murphi Verification System," *Computer-Aided Verification,* 1996.

[32] E.A. Emerson and E.M. Clarke, "Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons," *Science of Computer Programming,* vol. 2, pp. 241-266, 1982.

[33] E.A. Emerson and C.-L. Lei, "Modalities for Model Checking: Branching Time Logic Strikes Back," *Science of Computer Programming,* vol. 8, no. 3, pp. 275-306, 1987.

[34] W. Feller, *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, 1968.

[35] B.L. Fox and P.W. Glynn, "Computing Poisson Probabilities," *Comm. ACM,* vol. 31, no. 4, pp. 440-445, 1988.

[36] W.K. Grassmann, "Finding Transient Solutions in Markovian Event Systems through Randomization," *Numerical Solution of Markov Chains,* pp. 357-371, 1991.

[37] D. Gross and D.R. Miller, "The Randomization Technique as a Modeling Tool and Solution Procedure for Transient Markov Chains," *Operations Research,* vol. 32, no. 2, pp. 343-361, 1984.

[38] H. Hansson and B. Jonsson, "A Logic for Reasoning about Time and Reliability," *Formal Aspects of Computing,* vol. 6, pp. 512-535, 1994.

[39] B.R. Haverkort, *Performance of Computer Communication Systems: A Model-Based Approach.* John Wiley & Sons, 1998.

[40] B. Haverkort and H. Hermanns, J.-P. Katoen, "On the Use of Model Checking Techniques for Quantitative Dependability Evaluation," *Proc. IEEE Symp. Reliable Distributed Systems,* pp. 228-238, 2000.

[41] H. Hermanns, U. Herzog, and J.-P. Katoen, "Process Algebra for Performance Evaluation," *Theoretical Computer Science,* vol. 274, nos. 1-2, pp. 43-87, 2002.

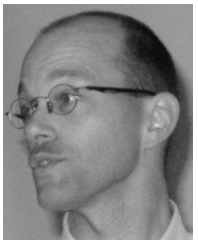
[42] H. Hermanns, U. Herzog, U. Klehmet, V. Mertsiotakis, and M. Siegle, "Compositional Performance Modeling with the TIPPtool," *Performance Evaluation,* vol. 39, nos. 1-4, pp. 5-35, 2000.

[43] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle, "A Markov Chain Model Checker," *J. Software Tools and Technology Transfer,* vol. 4, no. 2, pp. 153-172, 2003.
[44] J. Hillston, *A Compositional Approach to Performance Modeling.* Cambridge Univ. Press, 1996.
[45] G.J. Holzmann, "The Model Checker Spin," *IEEE Trans. Software Eng.,* vol. 23, no. 5, pp. 279-295, May 1997.
[46] R.A. Howard, *Dynamic Probabilistic Systems: Markov Models,* vol. 1. John Wiley & Sons, 1971.
[47] A. Jensen, "Markov Chains as an Aid in the Study of Markov Processes," *Skand. Aktuarietidskrift,* vol. 3, pp. 87-91, 1953.
[48] J.-P. Katoen, M.Z. Kwiatkowska, G. Norman, and D. Parker, "Faster and Symbolic CTMC Model Checking," *Process Algebra and Probabilistic Methods,* 2001.
[49] J.G. Kemeny and J.L. Snell, *Finite Markov Chains.* Van Nostrand, 1960.
[50] A.N. Kolmogorov, "Über die analytische Methoden in der Wahrscheinlichkeitsrechnung," *Mathematische Annalen,* vol. 104, pp. 415-458, 1931.
[51] A.N. Kolmogorov, "Anfangsgründe der Theorie der Markoffschen Ketten mit unendlichen vielen möglichen Zuständen," *Matematicheskii Sbornik N. S.,* pp. 607-610, 1936.
[52] U. Krieger, B. Müller-Clostermann, and M. Sczittnick, "Modeling and Analysis of Communication Systems Based on Computational Methods for Markov Chains," *IEEE Trans. Selected Areas in Comm.,* vol. 8, no. 9, pp. 1630-1648, 1990.
[53] V.G. Kulkarni, *Modeling and Analysis of Stochastic Systems.* Chapman Hall, 1995.
[54] M.Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston, "Automatic Verification of Real-Time Systems with Discrete Probability Distributions," *Theoretical Computer Science,* vol. 282, no. 1, pp. 101-150, 2002.
[55] K.G. Larsen and A. Skou, "Bisimulation through Probabilistic Testing," *Information and Computation,* vol. 94, no. 1, pp. 1-28, 1992.
[56] A.A. Markov, "Investigations of an Important Case of Dependent Trials," *Izvestia Acad.,* Nauk VI, Series I, vol. 61 (in Russian), 1907.
[57] J.F. Meyer, A. Movaghar, and W.H. Sanders, "Stochastic Activity Networks: Structure, Behavior, and Application," *Proc. Int'l Workshop Timed Petri Nets,* pp. 106-115, 1985.
[58] C. Moler and C.F. vanLoan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Rev.,* vol. 20, no. 4, pp. 801-835, 1978.
[59] J.K. Muppala and K.S. Trivedi, "Numerical Transient Solution of Finite Markovian Queueing Systems," *Queueing and Related Models,* 1992.
[60] W.D. Obal II and W.H. Sanders, "State-Space Support for Path-Based Reward Variables," *Performance Evaluation,* vol. 35, pp. 233-251, 1999.
[61] B. Plateau and K. Atif, "Stochastic Automata Networks for Modeling Parallel Systems," *IEEE Trans. Software Eng.,* vol. 17, no. 10, pp. 1093-1108, Oct. 1991.
[62] A. Pnueli and L. Zuck, "Probabilistic Verification," *Information and Computation,* vol. 103, pp. 1-29, 1993.
[63] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge Univ. Press, 1989.
[64] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, 1994.
[65] J.-P. Quielle and J. Sifakis, "Specification and Verification of Concurrent Systems in CESAR," *Proc. Int'l Symp. Programming,* 1982.
[66] A.L. Reibman, R. Smith, and K.S. Trivedi, "Markov and Markov Reward Models Transient Analysis: An Overview of Numerical Approaches," *European J. Operational Research,* vol. 4, pp. 257-267, 1989.
[67] A.L. Reibman and K.S. Trivedi, " Numerical Transient Analysis of Markov Models," *Computers and Operations Research,* vol. 15, no. 1, pp. 19-36, 1988.
[68] W.H. Sanders, W.D. Obal II, M.A. Qureshi, and F.K. Widnajarko, "The UltraSAN Modeling Environment," *Performance Evaluation,* vol. 24, pp. 89-115, 1995.
[69] W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains.* Princeton Univ. Press, 1994.
[70] R.E. Tarjan, "Depth-First Search and Linear Graph Algorithms," *SIAM J. Compting,* vol. 1, pp. 146-160, 1972.
[71] M.Y. Vardi, "Automatic Verification of Probabilistic Concurrent Finite State Programs," *Proc. Ann. Symp. Foundations of Computer Science,* pp. 327-338, 1985.
[72] H. Younes and R. Simmons, "Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling," *Computer-Aided Verification,* 2002.

Christel Baier received the master's degree in mathematics in 1990 from the University of Mannheim, Germany. She received the PhD degree (1994) and the venia legendi (1999), both from the Department of Computer Science at the University of Mannheim. Since the autumn of 1999, she has been a professor of computer science at the Rheinische Friedrich-Wilhelms Universität Bonn. Her research interests are the theory of concurrent and probabilistic systems, verification, semantics of programming languages, and process calculi and mathematical logic.

Boudewijn Haverkort received the engineering and PhD degrees in computer science, both from the University of Twente, in 1986 and 1991, respectively. He is the chair for the "design and analysis of communication systems" at the University of Twente, The Netherlands, in both the Department of Computer Science and Electrical Engineering since 2003. Prior to that, he was a professor of performance evaluation and distributed systems at the RWTH Aachen, Germany, a lecturer in computer science at the University of Twente in The Netherlands, and visiting researcher in the Teletraffic Research Centre at the University of Adelaide. His research interests encompass the design and performance and dependability evaluation of computer-communication systems, model checking, parallel and distributed computing, and fault-tolerant computer systems. He has published more than 75 papers in international journals and conference proceedings, edited several books and conference proceedings, and wrote a monograph on model-based performance evaluation of computer and communication systems. He is a senior member of the IEEE.

Holger Hermanns studied applied mathematics at the University of Bordeaux I, France, and computer science at the University of Erlangen-Nürnberg, Germany, where he received the diploma degree in 1993 (with honors) and the PhD degree from the Department of Computer Science in 1998 (with honors). Since 1998, he has been with the Faculty of Computer Science, University of Twente, The Netherlands, holding an associate professor position since October 2001. In 2003, he also became a professor of computer science at Saarland University, Germany. His research interests include distributed systems engineering, compositional performance and dependability modeling, model checking, and state space compression.



Joost-Pieter Katoen received the master's degree (with honors, 1987) and PhD degree (1996) in computer science, both from the University of Twente, The Netherlands. He is currently an associate professor in the Formal Methods and Tools Group at Twente and is a visiting research fellow at the University of Birmingham, United Kingdom. Prior to this, he held positions at the University of Erlangen-Nürnberg, Germany, Eindhoven University of Technology, and Philips Research Laboratories. His research interests include specification and verification of distributed and embedded systems, semantics, model checking, object-based systems, and the integration of formal methods and performance evaluation techniques. He is a member of the IEEE Computer Society.