Model-Checking Algorithms for Continuous-Time Markov Chains

Christel Baier[†], Boudewijn Haverkort[‡], Holger Hermanns[*] and Joost-Pieter Katoen[*1]

[†] *Institut für Informatik I, University of Bonn*
Römerstraße 164, D-53117 Bonn, Germany

[‡] *Laboratory for Performance Evaluation and Distributed Systems,*
Department of Computer Science, RWTH Aachen, D-52056 Aachen, Germany

[*] *Formal Methods and Tools Group, Faculty of Computer Science,*
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

E-mail: baier@cs.uni-bonn.de, haverkort@cs.rwth-aachen.de, {hermanns,katoen}@cs.utwente.nl

---

Continuous-time Markov chains (CTMCs) are 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.* [10, 11], contains a time-bounded until operator to express probabilistic timing properties over paths and 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. To solve the integral equation system by numerical means, we show that 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.

Key Words: continuous-time Markov chain, lumping, model checking, temporal logic, steady-state analysis, transient analysis, uniformization

[1] Corresponding author.

1. INTRODUCTION

Continuous-time Markov chains (CTMCs) [37, 51, 55, 58, 79] 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 queueing networks [29, 62], stochastic Petri nets [2], stochastic activity networks [66, 71] and stochastic process algebras [44, 49]. With appropriate software tools supporting these specification methods, such as e.g. provided by MACOM [57], SPNP [24], UltraSAN [78], or TIPPtool [45], it is relatively comfortable to specify performance and dependability models of which the underlying CTMCs have millions of states, cf. [79]. 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, utilisation. A measure of interest determines the kind of model analysis that is to be carried out in order to calculate, i.e., quantify 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 behaviour on 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 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) [35] 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. 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* [35, 75]; for an overview see [28]. The basis of model checking CTL is a systematic, usually exhaustive, state-space exploration to check whether

the 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 amongst others hardware and software systems, security protocols and e-commerce systems. With appropriate tools such as SMV [25, 64], SPIN [50] and Murφ [34] 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 extends the (equally named) logic by Aziz *et al.* [10, 11] with an operator to reason about steady-state probabilities. As an example of the latter, the formula $\mathcal{S}_{\leqslant 0.02}(error)$ asserts that the steady-state probability for an error-state is at most 0.02. Apart from the usual path-formulas like next and until, a time-bounded next X^I and a time-bounded until $\mathcal{U}^I$, with I a time interval, are incorporated, together with standard derivatives, such as a time-bounded eventually $\Diamond^I$. Typical properties expressible using these operators are that along a path a set of goal-states is reached at some time $t \in I$ while only visiting particular states before. As in the logic PCTL [41], a probabilistic variant of CTL interpreted over discrete-time Markov chains (DTMCs), the operator $\mathcal{P}_{\trianglelefteq p}(\varphi)$ replaces the usual CTL path-quantifiers $\forall$ and $\exists$ and refers to the probability for the event specified by the path formula φ. The subscript $\trianglelefteq p$ (where $\trianglelefteq$ is a comparison operator and $p \in [0,1]$) specifies a lower or upper bound for the "allowed" probabilities. The combination of the probabilistic operator with the temporal operator $\Diamond^{[t,t]}$ analyses the quantitative behaviour at time instant t and can be used to reason about transient-state probabilities. For instance, the formula $\mathcal{P}_{\leqslant 0.001}(\Diamond^{[4,4]} error)$ asserts that the probability for a system error at time instant 4 is at most 10^{-3}.

The model checking problem for CSL is known to be decidable [10, 11] (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 as models. We show that next and (unbounded) until-formulas 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 [41]. Checking steady-state properties reduces to solving a system of linear equations combined with standard graph analysis methods, while checking the time-bounded until 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 [26].

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 *uniformisation* [39, 40, 52, 68], 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 [22, 49] that can be viewed as a continuous variant of probabilistic bisimulation [61] — 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 [20] 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 as well as path-based performance measures,
- the characterization of the probability measure for time-bounded until-formulas in terms of a Volterra integral equation system,
- the computation of probability measures for time-bounded until-formulas by transient analysis, and
- the preservation of the validity of CSL-formulas under lumping.

This paper is based on the extended abstract [14] and the paper [15].

Organization of the paper. Section 2 introduces the basic concepts of CTMCs that are needed for the rest of the paper. 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 and presents suggestions for further work.

2. CONTINUOUS-TIME MARKOV CHAINS

This section recalls the basic concepts of continuous-time Markov chains (CTMCs) as originally developed by Markov [63] for finite state spaces and by Kolmogorov [56] for denumerable and continuous state spaces. The presentation is focussed on the concepts needed for the understanding of the rest of this paper; for a more elaborate treatment we refer to e.g. [37, 51, 54, 58, 79].

2.1. Labelled CTMCs

To ease the definition of the semantics of the logic CSL, we slightly depart from the standard notations for CTMCs and 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 2.1. A (labelled) CTMC $\mathcal{M}$ is a tuple $(S, \mathbf{R}, L)$ with

- S, a finite set of *states*

- $\mathbf{R} : S \times S \to \mathbb{R}_{\geqslant 0}$, the *rate matrix*, and
- $L : S \to 2^{AP}$, the *labelling* 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 this definition 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. In our setting, self-loops at state s are possible and are modelled 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 does neither alter the transient nor the steady-state behaviour 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, amongst others, the textbook [74].

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 non-absorbing state s to a particular state s' within t time units, i.e., the transition $s \to s'$ wins the race, is given by:

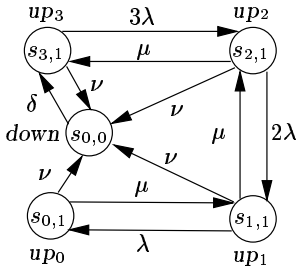
$$\mathbf{P}(s,s',t) \;=\; \frac{\mathbf{R}(s,s')}{E(s)} \cdot \left(1 - e^{-E(s) \cdot t}\right),$$

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 non-absorbing 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 2.2. 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^1_s, where $\alpha^1_s(s) = 1$ and $\alpha^1_s(s') = 0$ for any $s' \neq s$.

EXAMPLE 2.1. As a running example we address a *triple modular redundant system* (TMR) taken from [42], 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 $(0 \leqslant i < 4)$ processors and j $(0 \leqslant j \leqslant 1)$ voters are operational. As atomic propositions we use $AP = \{ up_i \mid 0 \leqslant i < 4 \} \cup \{ down \}$. 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., $\alpha = \alpha^1_{s_{3,1}}$. 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 modelling this system are (with a clock-wise ordering of states for the matrix/vector-representation, starting with $s_{3,1}$):



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

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 labelling is defined by $L(s_{i,1}) = \{ up_i \}$ for $0 \leqslant i < 4$ 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 2.3. Let $\mathcal{M} = (S, \mathbf{R}, L)$ be a CTMC. An infinite *path* σ is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \ldots$ 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 $i < l$.[3] ■

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 $t \in \mathbb{R}_{\geqslant 0}$ and i the smallest index with $t \leqslant \sum_{j=0}^{i} t_j$ 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 $i \leqslant l$; they are defined for $i < l$ in the above way, and

[3] 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.

$\delta(\sigma, l) = \infty$. 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 $\delta(\sigma, 1) = \sqrt{2}$, $\sigma[1] = s_0 = \sigma@1.7$, $\sigma[2] = s_1 = \sigma@2.4$, $\sigma[3] = s_2 = \sigma@5$, 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 in the notation whenever convenient.

2.3. Borel space

An initial distribution α yields a probability measure $\Pr_\alpha$ on paths as follows. Let $s_0, \ldots, s_k \in S$ with $\mathbf{R}(s_i, s_{i+1}) > 0$, $(0 \leqslant i < k)$, and $I_0, \ldots, I_{k-1}$ non-empty intervals in $\mathbb{R}_{\geqslant 0}$. 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 $\sigma[i] = s_i$ $(i \leqslant k)$, 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 $s = s_0$, $\mathbf{R}(s_i, s_{i+1}) > 0$ $(0 \leqslant i < k)$, and $I_0, \ldots, I_{k-1}$ ranges over all sequences of non-empty intervals in $\mathbb{R}_{\geqslant 0}$. 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 $k \geqslant 0$:

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

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 [7], we do not assume time divergence for infinite paths $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$. Although $\sum_{j \geqslant 0} t_j$ 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 2.1. *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 $\sum_{i \geqslant 0} t_i$ 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

$$\Pr\left(ConvPath(s)\right) = 0$$

Consider the set $B(s)$ of all paths σ that start in state s for which the delay of the transitions is never exceeding one time unit. Formally, $B(s)$ consists of

all paths $\sigma = s \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \ldots$ such that $t_i \leqslant 1$ 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 $\sigma = s \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \ldots$ where $t_i \leqslant 1$ for all $i < n$. Clearly,

$$B(s) = \bigcap_{n \geqslant 1} B_n(s)$$

By induction on n we obtain

$$\Pr(B_n(s)) \leqslant \left(1 - e^{-\Lambda}\right)^n$$

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 infinite sum $\sum_{i=0}^{\infty} t_i$ converges. In particular, the sequence $(t_i)_{i \geqslant 1}$ converges to 0. Thus, there exists some natural number $n \geqslant 1$ with $t_i \leqslant 1$ 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 $B(s_n)$. We conclude that $\mathit{ConvPath}(s)$ is a subset of

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

This yields:

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

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 non-negative reals) is at most 1, as it equals $\mathbf{P}(s, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \ldots \cdot \mathbf{P}(s_{n-1}, s_n)$. ■

This proposition can also be deduced, though not in a straighforward 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

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

stands for the probability to be in state s' at time t given the initial distribution α. 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 characterised by a system of linear differential equations, also known as the forward Chapman-Kolmogorov differential equations [37, 55, 58, 79]:

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

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

Steady-state probabilities are given by [37, 55, 58, 79]:

$$\pi^{\mathcal{M}}(\alpha, s') = \lim_{t\to\infty} \pi^{\mathcal{M}}(\alpha, s', t).$$

This limit always exists for finite CTMCs [58]. 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.,

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

We let $\pi^{\mathcal{M}}(\alpha, \varnothing) = 0$. Steady-state probabilities are computed from a system of linear equations

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

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, 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 [70], suitable mechanisms to express such measures have not been considered. In Section 3, we will introduce a logic-based approach that allows 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. These characterizations serve as the basis for the model checking algorithms for CSL.

3.1. Syntax of CSL

CSL is a branching-time temporal logic à la CTL [35] with state- and path-formulas based on [10, 11]. 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 behaviour 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) on the long run, the transient operator allows us to refer to the probability of the occurrence of particular *paths* in the CTMC, similar to [41]. In order to express the time-span of a certain path, the path-operators until ($\mathcal{U}$) and next (X) are extended with a parameter that specifies a time-interval.

DEFINITION 3.1. Let $p \in [0,1]$ a real number, $\trianglelefteq \in \{\leqslant, \geqslant\}$ a comparison operator, and $I \subseteq \mathbb{R}_{\geqslant 0}$ a non-empty 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, an informal explanation of the CSL-formulas is given. $\mathcal{S}_{\trianglelefteq p}(\Phi)$ 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 $\mathcal{P}_{\geqslant 1}(\varphi)$. For instance, $\mathcal{P}_{>0}(\Diamond a)$ is equivalent to $\exists\Diamond a$, and $\mathcal{P}_{\geqslant 1}(\Diamond a)$ stands for $\forall\Diamond a$ given a fair interpretation [36] of the CTL-formula $\forall\Diamond a$. An elaborate discussion about the relation between fairness and probabilities goes beyond the scope of this paper; we refer the interested reader to [17]. 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 3.2. Let $Sat(\Phi) = \{\, s \in S \mid s \models \Phi \,\}$. The relation $\models$ for CSL state-formulas is defined by:

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

■

Here, $Prob^{\mathcal{M}}(s, \varphi)$ 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 3.3. The relation $\models$ for CSL path-formulas is defined by:

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

■

We note that for $I = \varnothing$ 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:

$$X\Phi = X^{[0,\infty)}\, \Phi \quad \text{and} \quad \Phi\, \mathcal{U}\, \Psi = \Phi\, \mathcal{U}^{[0,\infty)}\, \Psi$$

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 $\Diamond^I$ is 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:

$$\mathcal{P}_{\trianglerighteq p}(\Box^I\, \Phi) = \mathcal{P}_{\trianglelefteq 1-p}(\Diamond^I\, \neg\Phi) \text{ and } \mathcal{P}_{\trianglerighteq p}(\Box\, \Phi) = \mathcal{P}_{\trianglelefteq 1-p}(\Diamond\, \neg\Phi).$$

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. Recall that atomic proposition at_s is valid in state s and invalid in any other state.

Steady-state measures. The formula $\mathcal{S}_{\trianglelefteq p}(at_s)$ imposes a requirement on the steady-state probability to be in state s. For instance, $\mathcal{S}_{\leqslant 10^{-5}}(at_{s_{2,1}})$ 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 towards 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 $\mathcal{S}_{\geqslant 0.99}(up_3 \vee up_2)$ states that on the long run, for at least 99% of the time at least 2 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, $\mathcal{P}_{\leqslant 0.2}(\Diamond^{[5,5]}\ at_{s_{2,1}})$ 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 $\mathcal{P}_{\geqslant 0.99}(\Diamond^{[t,t]}\, up_3 \vee up_2)$ requires that the probability to have at least 2 processors running at time t is at least 0.99. For specification convenience, a transient-state operator

$$\mathcal{T}^{@t}_{\trianglelefteq p}(\Phi) = \mathcal{P}_{\trianglelefteq p}(\Diamond^{[t,t]}\, \Phi)$$

could be defined.

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,

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

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 2 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 measures of interest can be specified. These are useful to obtain a more detailed insight into the system's behaviour. We provide two examples. The property

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

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,

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

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	$\mathcal{P}_{\trianglelefteq p}(\Phi\, \mathcal{U}^{[t,t']}\, \mathcal{S}_{\trianglelefteq q}(up))$

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 2 processors when the system is in equilibrium. Besides, prior to reaching state s the system must be operational continuously.

Summary and conclusion. Table 1 surveys some performance and dependability measures and their formulation in CSL, where *up* characterises 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 (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 utmost importance for their automated analysis (as proposed in the sequel). Furthermore, an important aspect of CSL is the possibility to state performance and dependability requirements over a selective set of paths through a model, which was not possible previously. Finally, the possibility to nest 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 [26] 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 non-probabilistic state-operators this procedure is the same as for CTL. The main remaining problem 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 $\mathbf{R}(s,s') > 0$. Sub-graph B is a *bottom strongly connected component* (BSCC) of G if it is a strongly connected component such that once it is entered, it cannot be left. Formally, a BSCC B is a maximal strongly connected component set B of states

such that $Reach(s) \subseteq B$ for all $s \in B$. 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 3.1. *Let $\mathcal{M}$ be a CTMC $(S, \mathbf{R}, L)$ with $s \in S$, $S' \subseteq S$ and set of BSCCs $B(\mathcal{M})$. Then*

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

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

Proof. Follows directly from [58, Theorem 6.16]. ■

This result 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 of the CTMC under consideration are computed using (a slight variant of) an algorithm for computing strongly connected components [1, 80]. For each BSCC B that contains some Φ-state, we compute the steady-state probabilities in B to determine $\pi^B(s')$ over all states $s' \in B$ that satisfy Φ. These probabilities are determined using standard means for solving the linear equation system that characterizes steady-state probabilities, cf. Section 2.4. Formally, $\pi^B(s') = 1$ if $B = \{ s' \}$, and otherwise $\boldsymbol{\pi}^B$ 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. As we will show later, 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 and cumulating the steady-state probabilities for all Φ-states.

EXAMPLE 3.1. Consider CTMC $\mathcal{M}$ depicted in Fig. 1 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 grey 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_3 from state s_0 equals $\frac{1}{2} \cdot \sum_{k=0}^{\infty} (\frac{1}{4})^k = \frac{2}{3}$. Likewise, the probability of reaching B_1 from s_0 equals $\frac{1}{6}$. Trivially, $\pi^{B_1}(s_3) = 1$.

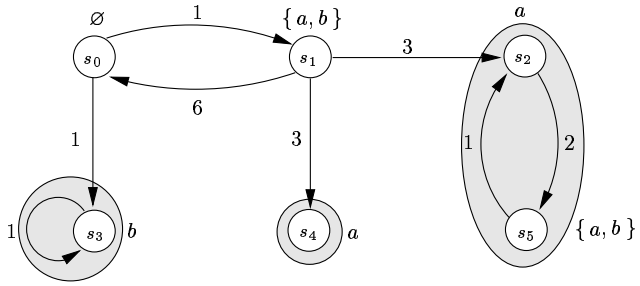


FIG. 1. An example non-strongly connected CTMC

$\pi^{B_3}(s_5)$ 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 3.1, 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 are 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 [13, 31, 41]. 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 by 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 $(0 \leqslant i < k)$ are replaced by their closure. In the sequel, we assume that interval I is compact.

PROPOSITION 3.2. *For $s \in S$, interval $I \subseteq \mathbb{R}_{\geqslant 0}$ 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. As we have mentioned in the definition of the Borel space over the paths, the probability to take some transition outgoing from state s at some time instant in interval I equals

$$\int_I E(s) \cdot e^{-E(s)\cdot t}\, dt \;=\; e^{-E(s)\cdot \inf I} - e^{-E(s)\cdot \sup I} \tag{3}$$

The probability to move from state s to a Φ-state at some time instant in I now equals the product of (3) and the probability of reaching a Φ-state in one step, i.e., $\sum_{s' \models \Phi} \mathbf{P}(s, s')$. ■

This result suggests the following algorithm. First the set $Sat(\Phi)$ is computed. State s is added to $Sat(\mathcal{P}_{\trianglelefteq p}(X^I\,\Phi))$ if $Prob(s, X^I\,\Phi) \trianglelefteq p$. The 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 [26] where appropriate fixed-point characterizations constitute the key towards model checking until-formulas. In the sequel, let $I \ominus x$ denote $\{\, t{-}x \mid t \in I \wedge t \geqslant x \,\}$ and $\mathbf{T}(s, s', x)$ denotes 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 $I \subseteq \mathbb{R}_{\geqslant 0}$.

THEOREM 3.1. *Let $s \in S$, interval $I \subseteq \mathbb{R}_{\geqslant 0}$ 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

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

Proof. First, we show that the function $Prob(s, I) = Prob(s, \Phi\,\mathcal{U}^I\,\Psi)$, is a fixed point of Ω. Let $s \in S$ and I be an arbitrary nonempty interval in $\mathbb{R}_{\geqslant 0}$. 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 $0 \leqslant x \leqslant b$ 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

- either $0 \leqslant x \leqslant a$ and $\sigma' \in Path(s', I \ominus x)$ for some state s'
- or $x > a$.

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 $x \leqslant a$) 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

- $s_0, \ldots, s_{n-1} \models \Phi$ and $s_n \models \Psi$
- $t_0 + t_1 + \ldots + t_n \in I$.

Let $Path_{\leqslant n}(s, I) \;=\; \bigcup_{0 \leqslant i \leqslant n} Path_i(s, I)$ and $Prob_{\leqslant n}(s, I)$ the probability measure of all paths $\sigma \in Path_{\leqslant n}(s, I)$. It is easy to see that for $n = 1, 2, 3, \ldots$

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

Moreover,

$$\lim_{n \to \infty} \; Prob_{\leqslant n}(s, I) \;=\; Prob(s, I).$$

Let $F : S \times \mathcal{I} \to [0, 1]$ be a fixed point of Ω. By induction on n we obtain:

$$F(s, I) \;\geqslant\; Prob_{\leqslant n}(s, I).$$

This yields $F(s, I) \;\geqslant\; \lim_{n \to \infty} \; Prob_{\leqslant n}(s, I) \;=\; Prob(s, I)$. ■

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 $Prob(s, \Phi \, \mathcal{U}^{\geqslant t} \, \Psi)$ equals

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

The characterization in Theorem 3.1 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 ($x \leqslant \sup I$), 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 3.2. Consider our running CTMC example again with $\lambda = 0.01$ and $\nu = 0.001$. We check $s_{3,1} \models \mathcal{P}_{\geqslant 0.2}(up_3\,\mathcal{U}^{[2,5]}\,up_2)$. According to Theorem 3.1 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 $x \leqslant 2$ and $2 \leqslant x \leqslant 5$ yields

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

According to Theorem 3.1, $Prob(s_{2,1}, up_3\,\mathcal{U}^{[2-x,5-x]}\,up_2)$ is 0 for $x \leqslant 2$ (case otherwise), and equals 1 for $2 \leqslant x \leqslant 5$, so that we finally obtain:

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

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

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

COROLLARY 3.1. *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 3.2, Theorem 3.1 and the fact that $X = X^{[0,\infty)}$ and $\mathcal{U} = \mathcal{U}^{[0,\infty)}$. ■

These results 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. [13, 41]. 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 $\underline{i}_{\Phi}$ characterises 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 [79]) 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 $\mathcal{P}_{\geqslant 1}(\Phi\,\mathcal{U}\,\Psi)$. Details can be found in [31, 41].

4. MODEL-CHECKING TIME-BOUNDED UNTIL

One of the main results of the previous section was the characterisation 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 the equation system directly. To overcome the encountered problems, 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.

4.1. Numerically solving the integral equation system

Two obvious techniques that could be applied to solve the recursive integral equation of Theorem 3.1 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 3.1 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 [73]. Experiments have shown that this approach is rather time-consuming and that numerical stability is hard to achieve [47].

Alternatively, the recursive integral formula equation of Theorem 3.1 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:

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

where

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

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

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

$$\underline{y}(t) \;=\; e^{\widehat{\mathbf{R}}\cdot t}\cdot\left(\underline{i}_{\Psi} \;+\; \int_0^t e^{-\widehat{\mathbf{R}}\cdot x}\cdot\underline{b}(x)\;dx\right) \text{ where } \quad e^{\widehat{\mathbf{R}}\cdot x} \;=\; \sum_{k=0}^{\infty}\frac{(\widehat{\mathbf{R}}\cdot x)^k}{k!}$$

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 operator 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):

$$\pi^{\mathcal{M}}(s,Sat(\Phi),t) = Prob^{\mathcal{M}}(s,\Diamond^{[t,t]}\Phi) = \sum_{s'\models\Phi}\pi^{\mathcal{M}}(s,s',t) \tag{4}$$

for arbitrary state-formula Φ.

In this section, we show how the calculation of the general case, $Prob(s,\Phi\,\mathcal{U}^{I}\,\Psi)$, can be reduced to various instances of classical transient analysis on CTMCs. The key to our approach is to perform transient analysis on a CTMC that – depending on the property under consideration, i.e., the measure-of-interest – is appropriately transformed.

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 4 types of time-bounded until-formulas with a non-empty compact interval I and show how they all can be reduced to instances of two simple base cases. We first define a transformation on CTMCs that is parameterized with a CSL state-formula.

DEFINITION 4.1. 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., $\mathcal{M}[\Phi] = (S, \mathbf{R}', L)$ where $\mathbf{R}'(s, s') = \mathbf{R}(s, s')$ 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 will not 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 obtained 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 4.1. *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

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

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 (5) to:

$$\Pr_s\{\, \sigma \in Path^{\mathcal{M}} \mid \sigma@t \models \Psi \wedge (\forall y \in [0,t).\, \sigma@y \models \Phi \vee \Psi)\,\} \tag{6}$$

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, (6) 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. equation (4). ■

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 Ψ-state has been reached, the future behaviour 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 4.1. As $\mathcal{M}[\Psi][\neg\Phi \wedge \neg\Psi] = \mathcal{M}[\neg\Phi \vee \Psi]$ we obtain:

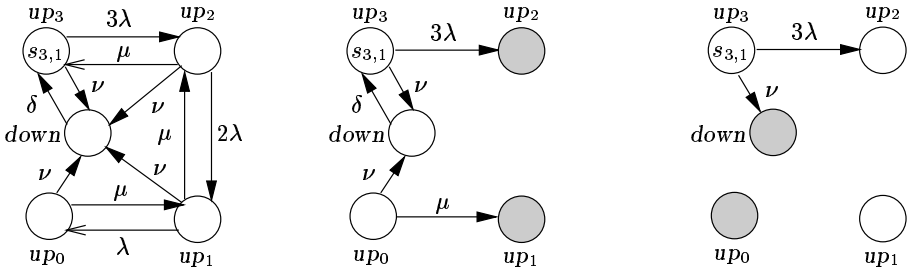


FIG. 2. Successive transformations on the TMR example for $(up_3 \vee up_2)\,\mathcal{U}^{[0,t]}\,(up_2 \vee up_1)$

THEOREM 4.1. *For any CTMC $\mathcal{M}$:*

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

Proof. Straightforward verification using the observation that

$$\Pr_s\{\,\sigma \in Path^{\mathcal{M}} \mid \sigma \models \Phi\,\mathcal{U}^{[0,t]}\,\Psi\,\} = \Pr_s\{\,\sigma \in Path^{\mathcal{M}[\Psi]} \mid \sigma \models \Phi\,\mathcal{U}^{[0,t]}\,\Psi\,\}$$

and using Proposition 4.1. ■

EXAMPLE 4.1. Consider the TMR-model with initial distribution $\alpha = \alpha^1_{s_{3,1}}$, and let $\Phi' = \mathcal{P}_{\geqslant 0.05}(\Phi\,\mathcal{U}^{[0,4]}\,\Psi)$ 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 4.1, model checking Φ' on the original CTMC of Example 2.1, depicted in Fig. 2(a), amounts to verify this property on the CTMC depicted in Fig. 2(b) where the grey-shaded states indicate the (now absorbing) states satisfying Ψ. Proposition 4.1 now yields that it suffices to check the property $\mathcal{P}_{\geqslant 0.05}(\Diamond^{[4,4]}\Psi)$ on the CTMC of Fig. 2(c), where the grey-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 (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 $\Psi \Rightarrow \Phi$ is valid. With the same motivation as for Proposition 4.1, $(\neg\Phi \wedge \neg\Psi)$-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 4.2. *If $\Psi \Rightarrow \Phi$ we have for any CTMC $\mathcal{M}$:*

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

Proof. For $\Psi \Rightarrow \Phi$ it follows that $Prob^{\mathcal{M}}(s, \Phi\,\mathcal{U}^{[t,t]}\,\Psi)$ equals

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

As all $\neg(\Phi \vee \Psi)$-states are absorbing in $\mathcal{M}[\neg\Phi \wedge \neg\Psi]$ it now directly follows that this equals $Prob^{\mathcal{M}[\neg\Phi\wedge\neg\Psi]}(s, \Diamond^{[t,t]}\,\Psi)$. Using (4) we obtain the summation. ■

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

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

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

THEOREM 4.2. *For any CTMC $\mathcal{M}$ and $0 < t \leqslant t'$:*

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

Proof. Let $\varphi = \Phi\,\mathcal{U}^{[t,t']}\,\Psi$ with $0 < t \leqslant t'$. 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

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

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 4.2 to the left factor and Theorem 4.1 to the right 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 – using again Proposition 4.2 and Theorem 4.1 – 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 4.2. Consider the model of the TMR system with initial distribution $\alpha = \alpha^1_{s_{3,1}}$ and let $\Phi' = \mathcal{P}_{\geqslant 0.05}(\Phi\,\mathcal{U}^{[3,7]}\,\Psi)$ for $\Phi = up_3 \vee up_2$ and $\Psi = up_2 \vee up_1$

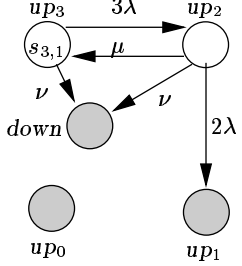


FIG. 3. The CTMC $\mathcal{M}[\neg\Phi]$

and $\varphi = \Phi\,\mathcal{U}^{[3,7]}\,\Psi$. Note that Theorem 4.2 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 4.2, model checking 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. 3 where all $\neg\Phi$-states (indicated in grey) 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 4.2 suggests to compute the transient probabilities at time 4 in CTMC $\mathcal{M}_2$ of Fig. 2(c) starting from initial 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 4.1. *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 4.2. ■

Note that equation (4) is a simplified version of this corollary.

4.3. Uniformisation

In the previous subsections, 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 solution to the Chapman-Kolmogorov differential equations (cf. Section 2.4), and are given by the Taylor-MacLaurin series:

$$\underline{\pi}(\alpha, t) = \alpha \cdot e^{\mathbf{Q} \cdot t} = \alpha \cdot \sum_{i=0}^{\infty} \frac{(\mathbf{Q} \cdot t)^i}{i!} \quad \text{with} \quad \mathbf{Q} = \mathbf{R} - diag(\underline{E}), \tag{7}$$

where we recall that $\underline{\pi}(\alpha, t)$ denotes the vector of state probabilities at time t. This characterisation is not attractive for a numerical algorithm since [67, 79]: (i) it suffers from numerical instability as $\mathbf{Q}$ contains both positive and negative entries and (ii) 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 uniformisation [39, 40, 52] is currently regarded as the most attractive. Under special conditions, e.g., when the rates in $\mathbf{R}$ differ a large number of magnitudes (5 to 6), Runge-Kutta-like methods might perform better, see [76, 77]. For the sake of completeness, we briefly discuss the main aspects of uniformisation here. These details will play a significant role in discussing the

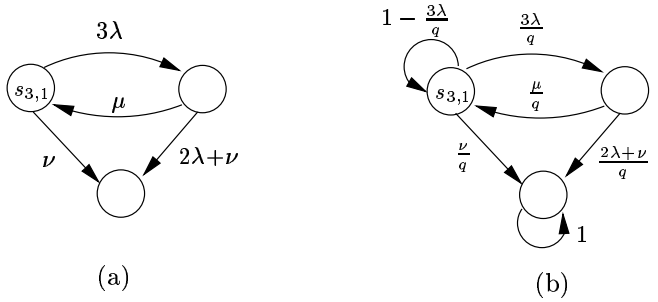


FIG. 4. A (a) CTMC and its uniformised (b) DTMC

efficiency of our model-checking algorithms, cf. Section 6. Uniformisation is used for the implementation of CSL model checking [47].

For CTMC $\mathcal{M} = (S, \mathbf{R}, L)$ the uniformised DTMC is $(S, \mathbf{U}, L)$ where $\mathbf{U}$ is defined by: $\mathbf{U} = \mathbf{I} + \mathbf{Q}/q$ for $q \geqslant \max_i\{ E(s_i) \}$. Transient state probabilities are now computed as follows. Substituting $\mathbf{Q} = q{\cdot}(\mathbf{U} - \mathbf{I})$ in (7) yields:

$$\underline{\pi}(\alpha, t) = \alpha{\cdot} \sum_{i=0}^{\infty} e^{-q\cdot t} \frac{(q{\cdot}t)^i}{i!} \mathbf{U}^i \tag{8}$$

where $e^{-q\cdot t}{\cdot}((q{\cdot}t)^i/i!)$ is the i-th Poisson probability with parameter qt, 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 [38]. The number of terms k_ϵ in (8) to be taken given a required accuracy ϵ is the smallest value satisfying:

$$\sum_{n=0}^{k_\epsilon} \frac{(q{\cdot}t)^n}{n!} \geqslant \frac{1-\epsilon}{e^{-q\cdot t}} = (1-\epsilon){\cdot}e^{q\cdot t} \tag{9}$$

If the product qt is large, k_ϵ tends to be of order $O(q{\cdot}t)$. On the other hand, if qt 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 [69]. This steady-state truncation point is often smaller than k_ϵ, making the trailing matrix-vector multiplications superfluous. For further details, see [42, 79].

EXAMPLE 4.3. Consider the TMR example and $\Phi = \mathcal{P}_{\geqslant 0.99}(\Box^{[0,3]}(up_3 \vee up_2))$. As explained in Example 4.2, it suffices to verify Φ on the CTMC of Fig. 3. This CTMC is equivalent to the CTMC depicted in Fig. 4(a), as will be justified in the next section. Assume $\mu > \lambda$. Checking Φ reduces to a transient analysis of the uniformised DTMC depicted in Fig. 4(b) 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 5 digits of precision): $q = 1.021$, $k_\epsilon = 13$, and $\underline{\pi}(s_{3,1}, 3) = (.96856, .02724, .00148)$, 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

An important ingredient for verifying probabilistic path-properties is a property-driven transformation of the CTMC under consideration, as being used in Section 4. This transformation makes certain sets of states absorbing. As a result, the size of the CTMC at hand is typically reduced. In this section, we discuss a technique to aggregate the state space of a CTMC even further. 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) [20] and probabilistic bisimulation and PCTL [9].

5.1. Bisimulation (lumping) equivalence

Lumpability is an important notion on CTMCs that allows their aggregation without affecting performance properties [22, 54]. We adapt the standard notion in order to deal with CTMCs with state-labellings. Rather than considering a state-labelling with atomic propositions, it is convenient for our purposes to consider labellings 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 labelling defined by $L_F(s) = \{ \Phi \in F \mid s \models \Phi \}$.

DEFINITION 5.1. 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 [22] and Markovian bisimulation [23, 49]. 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 $L_R([s]_R) = L_F(s)$. That is, $\mathcal{M}/R$ results from $\mathcal{M}$ by building the quotient space under R and labelling the states according to F (rather than AP). 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 [22, 54]:

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

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

EXAMPLE 5.1. The reflexive, symmetric and transitive closure of the relation

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

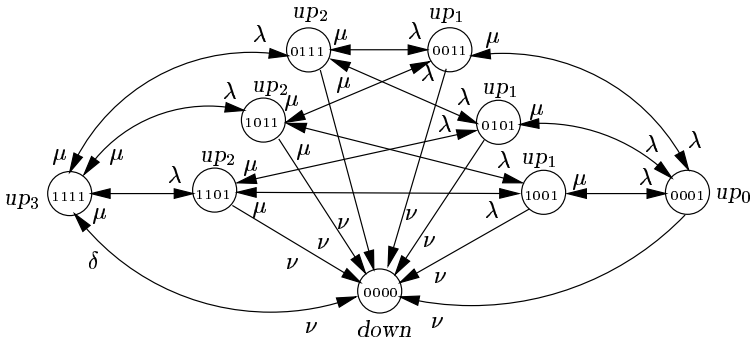


FIG. 5. A detailed version of the TMR model

is an AP-bisimulation on the set of states of the CTMC $\mathcal{M}$ depicted in Fig. 5. 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 2.1. For instance, state $s_{2,1}$ equals $[0111]_R$ and can be considered as the lumped state representing the three possible configurations in which, out of three, a single processor has failed. For initial distribution $\alpha = \alpha^1_{1111}$, that is, the system starts with all components running, the transient probability to be in state $s_{2,1}$ in $\mathcal{M}/R$ at time t equals $\pi^{\mathcal{M}}(\alpha, 0111, t) + \pi^{\mathcal{M}}(\alpha, 1011, t) + \pi^{\mathcal{M}}(\alpha, 1101, t)$.

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 5.1. *Let R be an F-bisimulation on $\mathcal{M}$ and s a state in $\mathcal{M}$. Then:*

(a) For all CSL_F-formulas Φ: $s \models_{\mathcal{M}} \Phi$ iff $[s]_R \models_{\mathcal{M}/R} \Phi$

(b) 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 $\mathcal{S}$-properties amounts to carrying out a steady-state analysis (after a graph analysis) and the verification of $\mathcal{P}$-properties boils down to a transient-state analysis (on a transformed CTMC), this result is not surprising given that bisimulation – ordinary lumping equivalence – preserves steady-state and transient-state probabilities. Theorem 5.1 allows us to verify **CSL**-formulas on the possibly much smaller $\mathcal{M}/R$ rather than on $\mathcal{M}$, for AP-bisimulation R.

The following result states that CSL-equivalence agrees with lumping, i.e., AP-bisimulation:

THEOREM 5.2. *$s \models_{\mathcal{M}} \Phi$ iff $s' \models_{\mathcal{M}} \Phi$ for all CSL-formulas if and only if s and s' are AP-bisimilar.*

Proof. The "only if" part of the proof follows directly from Theorem 5.1. The "if" part was recently shown in [33]. ■

We can exploit the above result and apply it to our transformations of the previous section by using the following observation. From Theorem 5.1(b) it follows:

$$\sum_{s' \models_{\mathcal{M}} \Phi} \pi^{\mathcal{M}}(s, s', t) \;=\; \sum_{S' \models_{\mathcal{M}/R} \Phi} \pi^{\mathcal{M}/R}([s]_R, S', t) \tag{10}$$

for any CSL_F-formula Φ and F-bisimulation R. This observation allows us to simplify the CTMCs $\mathcal{M}[\ldots]$ that occur in the cases A–D of our 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

$$S/R = \mathit{Sat}(\Phi \wedge \neg\Psi) \cup [\mathit{Sat}(\Psi)]_R \cup [\mathit{Sat}(\neg\Phi \wedge \neg\Psi)]_R \tag{11}$$

Since Ψ is a CSL_F-formula, equation (10) yields

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

for any state $s \in \mathit{Sat}(\Phi \wedge \neg\Psi)$. Similar arguments are applicable to case A and C. As a result, the sets $[\mathit{Sat}(\neg\Phi \wedge \neg\Psi)]_R$ and $[\mathit{Sat}(\Psi)]_R$ 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 $|\mathit{Sat}(\Phi \wedge \neg\Psi)| + 2$. 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 5.2. According to the above observations, in the CTMC of Fig. 3 we may aggregate states $[\mathit{Sat}(\Phi)]_R = \{\, s_{0,1}, s_{0,0}, s_{1,1} \,\}$ into a single state using an $\{\, \Phi, \Psi \,\}$-bisimulation where Φ and Ψ are given in Example 4.1. 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. 4(a). As a second example, in the CTMC of Fig. 2(c) we may collapse $[\mathit{Sat}(\Psi)]_R = \{\, s_{2,1}, s_{1,1} \,\}$ and $[\mathit{Sat}(\neg\Phi \wedge \neg\Psi)]_R = \{\, s_{0,0}, s_{0,1} \,\}$ into single states using an $\{\, \neg\Phi, \Psi \,\}$-bisimulation.

Note that the bisimulations constructed in this way have at most two non-trivial 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 [48]. It now follows from Theorem 5.2 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 5.1, 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 non-zero 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 $M \geqslant N$. Consider CSL state-formula Φ. The general strategy is identical to the model checking procedure for CTL [26]. The set $Sat(\Phi)$ of states satisfying Φ is computed in an iterative way, starting with considering the sub-formulas of Φ of length 1, i.e., the atomic propositions in Φ. These sub-formulas correspond to the leaves in the parse tree of Φ. In the $(i+1)$-th iteration of the algorithm sub-formulas of length $i+1$ are considered using the results of all sub-formulas of length at most i, i.e., the results of all sub-nodes 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 sub-formula (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. To model-check the steady-state operator, first a graph analysis is carried out to determine the BSCCs of $\mathcal{M}$. This takes $\mathcal{O}(N+M)$ time [80]. In 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. So the complexity of the steady-state operator is determined by the time it takes to solve a linear equation system. Using the method in [1] this takes $\mathcal{O}(N^{2.81})$ time. In practice it is often preferred to use iterative, numerical methods such as Power, Gauss-Seidel or similar [79] for large N. Then convergence depends on the structure of the equation system and is in principle not guaranteed (except for the Power method).

Next-formulas. The nodes that represent formulas whose outermost operator is the probabilistic-operator combined with time-bounded next step, i.e., formulas of the from $\mathcal{P}_{\trianglelefteq p}(X^I\,\Phi)$, 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, i.e., formulas of the form $\mathcal{P}_{\trianglelefteq p}(X\,\Phi)$.

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 $\mathcal{O}(N^{2.81})$ time. The special case $\mathcal{P}_{>0}(\Phi\,\mathcal{U}\,\Psi)$ can be treated in the same way as the CTL-formula $\exists(\Phi\,\mathcal{U}\,\Psi)$, whereas $\mathcal{P}_{\geqslant 1}(\Phi\,\mathcal{U}\,\Psi)$ corresponds to $\forall(\Phi\,\mathcal{U}\,\Psi)$ under a fair CTL-interpretation [36]. This yields a worst-case time complexity of $\mathcal{O}(N)$ for these special cases, cf. [41].

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 $0 < t \leqslant t'$. 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 4.1) 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 uniformisation, 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 uniformisation 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 [15]. An improvement suggested in [53] 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 this case.

For formulas of the form $\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}^{[t,t']}\,\Psi)$ with $0 < t \leqslant t'$, the computation is split into two parts, according to Theorem 4.2. 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 uniformisation on either chain can be quantified as follows. Even though the uniformisation rates in these two chains may differ, we can use the uniformisation 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 $t'-t \leqslant t' \geqslant t$. The method of [53] 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 non-empty 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. [26]. 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 $\mathcal{P}_{\geqslant 1}(\Phi\,\mathcal{U}^I\,\Psi)$ 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 will be much smaller than the above bound due to a built-in on-the-fly steady-state detection [69]. Furthermore, the uniformisation 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 means to interweave abstraction steps whenever appropriate during model checking $\mathcal{M}$. A priori, we can compute the best possible formula-independent lumping by

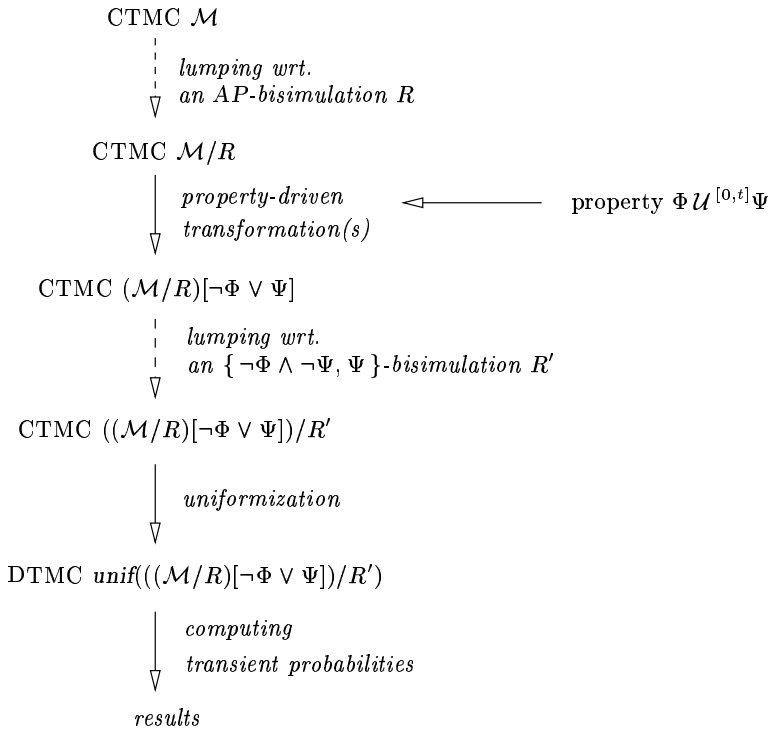


FIG. 6. Model checking time-bounded until-formulas

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 [48]. 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 sub-formula Ψ. For the time-bounded until such a recipe is illustrated in Fig. 6.

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 uniformisation is used for transient analysis. Cumulating over all nodes in the parse tree, i.e., all sub-formulas of Φ, we obtain that the worst-case time complexity of model checking CSL is

$$\mathcal{O}(|\Phi| \cdot (M \cdot q \cdot t_{max} + N^{2.81})),$$

where t_{max} is the maximum time bound of the time-bounded until sub-formulas occurring in Φ. Recall that q is the uniformisation 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. In conclusion, we have the following theorem.

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	$\mathcal{O}(N^{2.81})$
$\mathcal{P}_{\trianglelefteq p}(X\,\Phi)$	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	$\mathcal{O}(N^{2.81})$
$\mathcal{P}_{\trianglelefteq p}(\Phi\,\mathcal{U}^I\,\Psi)$	matrix-vector multiplication transient analysis	$\mathcal{O}(M{\cdot}q{\cdot}t')$

THEOREM 6.1. *For $\mathcal{M} = (S, \mathbf{R}, L)$ 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.*

7. RELATED WORK

Model-checking probabilistic systems. Early work on stochastic verification has concentrated on discrete-time models. Methods to verify a DTMC or a Markov decision process against a linear-time temporal logic (LTL) formula (sometimes specified as a Büchi automaton) have been considered, e.g., [32, 72, 81]. The basis of these works is the non-trivial reduction of the model-checking problem to the computation of the probabilities to reach certain sets of states (mostly, BSCCs). [31] describes 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 [41]. We have seen that for the CSL-operators that do not refer to the real-time behaviour of the CTMC, the PCTL algorithms can be exploited. In a similar way, as CTL* contains both LTL and CTL, the logic PCTL* contains both LTL and PCTL. PCTL* model checking is studied in [9, 18, 19]. Its basic idea is the reduction to the verification of quantitative LTL properties.

For work on the branching-time model checking of Markov decision processes we refer to [4, 17, 18, 19]. Here, non-determinism 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. Another research strand addressing similar verification issues is based on extending timed automata [7] with probabilities. A qualitative model-checking algorithm for a continuous probabilistic variant of timed automata has been proposed in [8]. 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 of discrete probabilistic timed automata [59] and a continuous variant thereof [60]. Our approach for CTMCs is not based on region-like constructions.

Model-checking continuous-time Markov chains. Model-checking CTMCs has received scant attention so far. A stochastic extension of CTL, also called CSL, was initially proposed in [10, 11]. Using transcendental number theory, these works prove the elementary result that the model-checking problem for CSL is decidable for rational time-bounds. No concrete algorithms were provided, though.

In [14] we extended CSL with the steady-state operator presented here to reason about the stationary behaviour of CTMCs. The first work on logics and model-checking algorithms for studying the stationary behaviour of stochastic systems, in particular semi-Markov decision processes, has been reported in [3, 4, 6]. Semi-Markov decision processes extend CTMCs with non-determinism 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, 6] uses experiments, a kind of 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. Verification of such properties takes place by constructing a synchronous product of the system model and the experiment, in a similar style as model-checking LTL. 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 [70]. Here, automata are used to define path-based stochastic variables on a Markov model described as a stochastic activity network [66]. Analysis takes place by considering a synchronous product of the model and the specification automaton, like in LTL model checking. Finally, the logic CSL has recently been extended to *continuous-space* Markov processes [33].

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 based on model checking for their analysis. This yields:

- a flexible means to specify standard and complex measures succinctly,
- automated means to analyse these measures over CTMCs,
- 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 [41]. 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 has recently been reported in [47].

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. Likewise, a formula-specific refinement of AP-bisimulation can further reduce the effort needed to model check a specific CSL-formula.

ACKNOWLEDGMENT

The authors thank Luca de Alfaro, Joachim Meyer-Kayser and Markus Siegle for valuable discussions. Holger Hermanns is supported by the Netherlands Organization for Scientific Research (NWO) and Joost-Pieter Katoen is partially supported by the Dutch Technology Foundation (STW). The co-operation 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 Dutch NWO and the German Research Council DFG.

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.* Ph.D dissertation, Stanford University, 1997.
4. L. de Alfaro. Temporal logics for the specification of performance and reliability. R. Reischuk and M. Morvan (eds), *4th Ann. Symposium on Theoretical Aspects of Computer Science*, LNCS 1200, pp. 165–176, Springer-Verlag, 1997.
5. L. de Alfaro. Stochastic transition systems. In D. Sangiorgi and R. de Simone (eds), *Concurrency Theory*, LNCS 1466, pp. 423–438, Springer-Verlag, 1998.
6. L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *Proc. IEEE 13th Symposium on Logic in Computer Science*, pp. 174–183, IEEE CS Press, 1998.
7. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, **126**: 183–235, 1994.
8. R. Alur, C. Courcoubetis and D. Dill. Model-checking for probabilistic real-time systems. In J.L. Albert, B. Monien, and M. Rodriguez-Artalejo (eds), *Automata, Languages and Programming*, LNCS 510: 115–127, Springer-Verlag, 1991.
9. A. Aziz, V. Singhal, F. Balarin, R. Brayton and A. Sangiovanni-Vincentelli. It usually works: the temporal logic of stochastic systems. In P. Wolper (ed), *Computer-Aided Verification*, LNCS 939, pp. 155–165, Springer-Verlag, 1995.
10. A. Aziz, K. Sanwal, V. Singhal and R. Brayton. Verifying continuous time Markov chains. In R. Alur and T.A. Henzinger (eds), *Computer-Aided Verification*, LNCS 1102, pp. 269–276, Springer-Verlag, 1996.
11. A. Aziz, K. Sanwal, V. Singhal and R. Brayton. Model checking continuous time Markov chains. *ACM Transactions on Computational Logic*, **1**(1): 162–170, 2000.
12. I. Bahar, E. Frohm, C. Gaona, G. Hachtel, E. Macii, A. Pardo and F. Somenzi. Algebraic decision diagrams and their applications. *Formal Methods in System Design*, **10**(2/3): 171–206, 1997.
13. C. Baier. On algorithmic verification methods for probabilistic systems. Habilitation thesis, University of Mannheim, Germany, 1999. (avaliable at `web.informatik.uni-bonn.de/I/papers/haupt.ps`).
14. C. Baier, J.-P. Katoen and H. Hermanns. Approximate symbolic model checking of continuous-time Markov chains. In J.C.M. Baeten and S. Mauw (eds), *Concurrency Theory*, LNCS 1664, pp. 146–162, Springer-Verlag, 1999.

15. C. Baier, B.R. Haverkort, H. Hermanns and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In E.A Emerson and A.P. Sistla (eds), *Computer Aided Verification*, LNCS 1855, pp. 358–372, Springer-Verlag, 2000.

16. C. Baier, B.R. Haverkort, H. Hermanns and J.-P. Katoen. On the logical characterisation of performability properties. In U. Montanari, J.D.P. Rolim, and E. Welzl (eds.), *Automata, Languages, and Programming (ICALP)*, LNCS 1853, pp. 780–792, Springer-Verlag, 2000.

17. C. Baier and M. Kwiatkowska. On the verification of qualitative properties of probabilistic processes under fairness constraints. *Information Processing Letters*, **66**(2): 71–79, 1998.

18. C. Baier and M.Z. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, **11**: 125–155, 1998.

19. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P.S. Thiagarajan (ed), *Foundations of Software Technology and Theoretical Computer Science*, LNCS 1026, pp. 499–513, Springer-Verlag, 1995.

20. M. Brown, E. Clarke, O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, **59**: 115–131, 1988.

21. R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, **C-35**(8): 677–691, 1986.

22. P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, **31**: 59–75, 1994.

23. P. Buchholz. Markovian process algebra. Technical Report 500, Fachbereich Informatik, University of Dortmund, 1994.

24. G. Ciardo, J.K. Muppala and K.S. Trivedi. SPNP: Stochastic Petri Net Package. In *Proc. 3rd Int. Workshop on Petri Nets and Performance Models*, pp. 142–151, IEEE CS Press, 1989.

25. A. Cimatti, E. Clarke, F. Giunchiglia and M. Roveri. NuSMV: a new symbolic model checker. *Int. Journal on Software Tools for Technology Transfer*, **2**: 410–425, 2000.

26. E. Clarke, E. Emerson and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, **8**: 244–263, 1986.

27. 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*, **10**(2/3): 149–169, 1997.

28. E. Clarke, O. Grumberg and D. Peled. *Model Checking*. MIT Press, 1999.

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

30. C. Courcoubetis and S. Tripakis. Probabilistic model checking: formalisms and algorithms for discrete and real-time systems. In *Lecture Notes NATO Summerschool on the Verification of Digital and Hybrid Systems*, 1997 (to appear).

31. C. Courcoubetis and M. Yannakakis. Verifying temporal properties of finite-state probabilistic programs. In *Proc. IEEE Symposium on Foundations of Computer Science*, pp. 338–345, IEEE CS-Press, 1988.

32. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, **42**(4): 857–907, 1995.

33. J. Desharnais and P. Panangaden. Continuous stochastic logic characterizes bisimulation of continuous-time Markov processes, 2001 (submitted for publication). (avaliable at `http://www-acaps.cs.mcgill.ca/~prakash/csl.ps`).

34. D.L. Dill. The Murphi verification system. In R. Alur and T.A. Henzinger (eds), *Computer-Aided Verification*, LNCS 1102, pp. 390–393, 1996.

35. E.A. Emerson and E.M. Clarke. Using branching time temporal logic to syntesize synchronization skeletons. *Science of Computer Programming*, **2**: 241–266, 1982.

36. E.A. Emerson and C.-L. Lei. Modalities for model checking: branching time logic strikes back. *Science of Computer Programming*, **8**(3): 275–306, 1987.

37. W. Feller. *An Introduction to Probability Theory and its Applications*. John Wiley & Sons, 1968.

38. B.L. Fox and P.W. Glynn. Computing Poisson probabilities. *Communications of the ACM* **31**(4): 440–445, 1988.

39. W.K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In W.J. Stewart (ed), *Numerical Solution of Markov Chains*, pp. 357–371, Marcel Dekker Inc, 1991.

40. D. Gross and D.R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov chains. *Operations Research* **32**(2): 343–361, 1984.

41. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing* **6**: 512–535, 1994.

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

43. B.R. Haverkort and I. Niemegeers. Performability modelling tools and techniques. *Performance Evaluation*, **25**: 17–40, 1996.

44. H. Hermanns, U. Herzog and J.-P. Katoen. Process algebra for performance evaluation. *Theoretical Computer Science*, **272**(1-2), 2001 (to appear).

45. H. Hermanns, U. Herzog, U. Klehmet, V. Mertsiotakis and M. Siegle. Compositional performance modelling with the TIPPTOOL. *Performance Evaluation*, **39**(1-4): 5–35, 2000.

46. H. Hermanns and J.-P. Katoen. Automated compositional Markov chain generation for a plain-old telephone system. *Science of Computer Programming*, **36**(1): 97–127, 2000.

47. H. Hermanns, J.-P. Katoen, J. Meyer-Kayser and M. Siegle. A Markov chain model checker. In S. Graf and M. Schwartzbach (eds), *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS 1785, pp. 347–362, Springer-Verlag, 2000.

48. H. Hermanns and M. Siegle. Bisimulation algorithms for stochastic process algebras and their BDD-based implementation. In J.-P. Katoen (ed), *Formal Methods for Real-Time and Probabilistic Systems*, LNCS 1601, pp. 244–265, Springer-Verlag, 1999.

49. J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

50. G.J. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, **23**(5): 279–295, 1997.

51. R.A. Howard. *Dynamic Probabilistic Systems; Volume 1: Markov Models*. John Wiley & Sons, 1971.

52. A. Jensen. Markov chains as an aid in the study of Markov processes. *Skand. Aktuarietidskrift* **3**: 87–91, 1953.

53. J.-P. Katoen, M.Z. Kwiatkowska, G. Norman and D. Parker. Faster and symbolic CTMC model checking. In L. de Alfaro and S. Gilmore (eds), *Process Algebra and Probabilistic Methods*, LNCS 2165, pp. 23–38, Springer-Verlag, 2001.

54. J.G. Kemeny and J.L. Snell. *Finite Markov Chains*. Van Nostrand, 1960.

55. A.N. Kolmogorov. Über die analytische Methoden in der Wahrscheinlichkeitsrechnung. *Math. Ann.*, **104**: 415–458, 1931.

56. A.N. Kolmogorov. Anfangsgründe der Theorie der Markoffschen Ketten mit unendlichen vielen möglichen Zuständen. *Mat. Sbornik N.S.*, pp. 607–610, 1936.

57. U. Krieger, B. Müller-Clostermann and M. Sczittnick. Modelling and analysis of communication systems based on computational methods for Markov chains. *IEEE Transactions on Selected Areas in Communications*, **8**(9): 1630–1648, 1990.

58. V.G. Kulkarni. *Modeling and Analysis of Stochastic Systems*. Chapman & Hall, 1995.

59. M.Z. Kwiatkowska, G. Norman, R. Segala and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. In J.-P. Katoen (ed), *Formal Methods for Real-Time and Probabilistic Sys.*, LNCS 1601, pp. 75–95, Springer-Verlag, 1999.

60. M.Z. Kwiatkowska, G. Norman, R. Segala and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In C. Palamidessi (ed), *Concurrency Theory*, LNCS 1877, pp. 123–137, Springer-Verlag, 2000.

61. K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, **94**(1): 1–28, 1992.

62. E.D. Lazowka, J.L. Zahorjan, G.S. Graham and K.C. Sevcik. *Quantitative System Performance: Computer System Analysis using Queueing Network Models*. Prentice-Hall, 1982.

63. A.A. Markov. Investigations of an important case of dependent trials. *Izvestia Acad. Nauk VI, Series I*, **61**, 1907 (in Russian).

64. K.L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.

65. J.F. Meyer. On evaluating the performability of degradable computing systems. *IEEE Transaction on Computers*, **29**(8): 720–731, 1980.

66. J.F. Meyer, A. Movaghar and W.H. Sanders. Stochastic activity networks: structure, behavior and application. In *Proc. Int. Workshop on Timed Petri Nets*, pp. 106–115, IEEE CS Press, 1985.

67. C. Moler and C.F. van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review* **20**(4): 801–835, 1978.

68. A.P.A. van Moorsel and B.R. Haverkort. Probabilistic evaluation for the analytical solution of large Markov models. *Microelectronics and Reliability* **36**(6): 733–755, 1996.

69. J.K. Muppala and K.S. Trivedi. Numerical transient solution of finite Markovian queueing systems. In U. Bhat (ed), *Queueing and Related Models*, Oxford University Press, 1992.

70. W.D. Obal II and W.H. Sanders. State-space support for path-based reward variables. *Performance Evaluation*, **35**: 233–251, 1999.

71. B. Plateau and K. Atif, Stochastic automata networks for modeling parallel systems. *IEEE Transactions on Software Engineering*, **17**(10): 1093–1108, 1991.

72. A. Pnueli and L. Zuck. Probabilistic verification. *Information and Computation*, **103**: 1–29, 1993.

73. W. Press, B. Flannery, S. Teukolsky and W. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1989.

74. M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

75. J.-P. Quielle and J. Sifakis. Specification and verification of concurrent systems in CESAR. In M. Dezani-Ciancaglini and U. Montanari (eds), *Proc. of the Int. Symposium on Programming*, LNCS 137, pp. 337–351, Springer-Verlag, 1982.

76. A.L. Reibman, R. Smith, and K.S. Trivedi. Markov and Markov reward models transient analysis: An overview of numerical approaches. *European Journal of Operations Research*, **4**: 257–267, 1989.

77. A.L. Reibman and K.S. Trivedi. Numerical transient analysis of Markov models. *Computers and Operations Research*, **15**(1): 19–36, 1988.

78. W.H. Sanders, W.D. Obal II, M.A. Qureshi and F.K. Widnajarko. The UltraSAN Modeling Environment. *Performance Evaluation*, **24**: 89–115, 1995.

79. W.J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

80. R.E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal of Computing*, **1**: 146–160, 1972.

81. M.Y. Vardi. Automatic verification of probabilistic concurrent finite state programs. In *Proc. IEEE Symposium on Foundations of Computer Science*, pp. 327–338, IEEE CS Press, 1985.