Three-Valued Abstraction for Continuous-Time Markov Chains*

Joost-Pieter Katoen[1], Daniel Klink[1], Martin Leucker[2], and Verena Wolf[3]

RWTH Aachen University[1], TU Munich[2], University of Mannheim[3], Germany

Abstract. This paper proposes a novel abstraction technique for continuous-time Markov chains (CTMCs). Our technique fits within the realm of three-valued abstraction methods that have been used successfully for traditional model checking. The key idea is to apply abstraction on uniform CTMCs that are readily obtained from general CTMCs, and to abstract transition probabilities by intervals. It is shown that this provides a conservative abstraction for both *true* and *false* for a three-valued semantics of the branching-time logic CSL (Continuous Stochastic Logic). Experiments on an infinite-state CTMC indicate the feasibility of our abstraction technique.

1 Introduction

Continuous-time Markov chains (CTMCs) are an important class of stochastic processes that are extensively used in a wide range of application domains ranging from planning of production lines and safety-critical systems to systems biology. Model checking of CTMCs has been proved to extend and complement long-standing analysis techniques for Markov processes. Tools for stochastic Petri nets such as SMART [8] and GreatSPN [9], the PEPA Workbench [12] (a stochastic variant of the CWB), and Statemate [7] have adopted model checkers to analyse CTMCs, and temporal logics for Markov chains became prominent property specification techniques in performance and dependability analysis.

Like for traditional model checking, one of the main challenges in the automated verification of CTMCs is the state-space explosion problem. This paper proposes a novel *abstraction* technique for CTMCs. Abstraction amounts to obtain smaller models by collapsing sets of concrete states to abstract states. In two-valued semantics, abstraction is typically conservative in the sense that affirmative verification results for abstract models carry over to concrete models. False negatives may occur due to overapproximation. Promising results in traditional model checking have been obtained for a three-valued semantics of temporal logic formulae, i.e., an interpretation in which a formula evaluates to either true, false or indefinite. In this setting, abstraction is conservative for both positive and negative verification results. Only if the verification of the abstract

* The research has been partially funded by the DFG Research Training Group 1298 (AlgoSyn).

model yields an indefinite answer, the validity in the concrete model is unknown. The abstraction technique proposed here follows this three-valued approach.

We consider abstractions for the branching-time logic CSL [3], a real-time probabilistic variant of CTL. CSL is a powerful logic for expressing quantitative time-bounded constrained reachability properties such as the probability to reach a set of goal states (by avoiding bad states) within a maximal time span exceeds $\frac{7}{8}$. Existing abstraction techniques in this setting that have been applied in practice consider either bisimulation [16], matrix bounding [6], simulation [24] or symmetry reduction [19]. (Due to the absence of nondeterminism, techniques such as partial-order reduction do not yield substantial reductions.) Despite the fact that fairly large reductions have recently been reported, more aggressive abstraction techniques are needed. Such techniques would also be useful to obtain finite abstractions for a larger class of infinite-state CTMCs.

In traditional model checking, abstract models contain may and must transitions as over- and under-approximation, respectively of the concrete transition relation. This concept can be lifted to discrete-time Markov chains (DTMCs) in a rather natural way [11, 14, 15] by replacing transition probabilities by *intervals* where lower and upper bounds act as under- and over-approximation, respectively. In this paper we investigate such techniques for CTMCs. The main technical complication is that besides transition probabilities, one has to determine the residence time of an abstract state that results from concrete states with distinct residence times. We show that intervals of transition probabilities, intervals on residence times (or combinations thereof) are not satisfactory in terms of precision. Instead, we suggest to overcome this imprecision by using *uniform* CTMCs, i.e., CTMCs in which all states have equal residence times and use transition probability intervals. Note that this is not a restriction, as any CTMC can be transformed into a weak bisimilar uniform CTMC in linear time. The abstraction is shown to preserve simulation: concrete states are simulated by their abstract counterparts. Then we show that extreme schedulers suffice, i.e., schedulers that only consider lower- and upper bounds. This allows to compute reachability probabilities up to a given tolerance ε rather efficiently [2]. Using a three-valued semantics of CSL it is shown that the abstraction is indeed conservative for affirmative and negative verification results. Besides, we show the relationship with the approach in [11] for DTMCs. The feasibility of the approach is shown by considering abstractions of different granularity for an unbounded stochastic Petri net.

Related work. Abstraction-refinement has been applied to reachability problems in MDPs [10], partial-order reduction techniques using Peled's ample-set method have been generalised to MDPs [13], abstract interpretation has been applied to MDPs [20], and various bisimulation equivalences and simulation preorders allow model aggregation prior to model checking, see e. g., [4, 23]. Recent techniques that have been proposed include abstraction of MDPs by two-player stochastic games [18], and symmetry reduction [19]. To our knowledge, three-valued abstraction of continuous-time stochastic models has not been considered.

2 Preliminaries

Let X be a finite set. For $Y, Y' \subseteq X$ and function $Q : X \times X \to \mathbb{R}_{\geq 0}$ let $Q(Y, Y') = \sum_{y \in Y, y' \in Y'} Q(y, y')$. The function $Q(x, \cdot)$ is given by $x' \mapsto Q(x, x')$ for all $x' \in X$. Furthermore a function f is called a *distribution on* X iff $f : X \to [0, 1]$ and $f(X) := \sum_{x \in X} f(x) = 1$. Let AP be a fixed, finite set of atomic propositions and $\mathbb{B}_2 = \{\bot, \top\}$ the two-valued truth domain.

Definition 1 (DTMC). *A DTMC is a tuple $(S, \mathbf{P}, L)$ with a finite non-empty set of states S, transition probability function $\mathbf{P} : S \times S \to [0, 1]$ satisfying $\mathbf{P}(s, S) = 1$ for all $s \in S$, and labeling function $L : S \times AP \to \mathbb{B}_2$.*

$\mathbf{P}(s, s')$ is the one-step probability to move from s to s' and $L(s, a)$ states if atomic proposition a holds in s. A DTMC is time-abstract; in contrast, CTMCs are time-aware, as they have an explicit reference to time, in the form of exit rates which determine, together with the transition probabilities, the stochastic evolution of the system in time.

Definition 2 (CTMC). *A CTMC $\mathcal{M}$ is a tuple $(S, \mathbf{P}, E, L)$ with S, $\mathbf{P}$ and L as before, and* exit rate $E : S \to \mathbb{R}_{>0}$.

The quantity $E(s)$ determines the random residence time of s, i.e. $1 - e^{-E(s) \cdot t}$ is the probability to take a transition emanating from s within the next t time units. (Note that self-loops are admitted.) The probability to move from s to s' within t time units is now given by $\mathbf{P}(s, s', t) := \mathbf{P}(s, s') \cdot (1 - e^{-E(s) \cdot t})$.

The time-abstract probabilistic behaviour of CTMC $\mathcal{M}$ is described by its embedded DTMC. The *embedded* DTMC of CTMC $\mathcal{M} = (S, \mathbf{P}, E, L)$ is simply given by $emb(\mathcal{M}) = (S, \mathbf{P}, L)$. A CTMC is *uniform* if all its states have the same exit rate, i.e., $E(s) = E(s') = e$ for all states $s, s' \in S$. Each CTMC can be transformed into a uniformized CTMC by adding self-loops:

Definition 3 (Uniformisation). *Let $\mathcal{M} = (S, \mathbf{P}, E, L)$ be a CTMC and let (uniformisation rate) $e \in \mathbb{R}_{>0}$ such that $e \geq \max_{s \in S} E(s)$. Then, $unif(\mathcal{M}) = (S, \overline{\mathbf{P}}, \overline{E}, L)$ is a* uniform *CTMC with $\overline{E}(s) = e$ for all $s \in S$ and*

$$\overline{\mathbf{P}}(s, s') = \mathbf{P}(s, s') \cdot \tfrac{E(s)}{e} \text{ for } s' \neq s \quad \text{and} \quad \overline{\mathbf{P}}(s, s) = 1 - \overline{\mathbf{P}}(s, S \backslash \{s\}).$$

The minimal appropriate value of e is determined by the state in $\mathcal{M}$ with the shortest mean residence time[1]. In $unif(\mathcal{M})$ all rates of self-loops are "normalized" with respect to e. Thus, transitions occur with an average "pace" of e, uniform for all states. A CTMC is weak bisimilar to its uniformized CTMC [4].

Continuous Stochastic Logic. CSL [1, 3] extends CTL by replacing existential and universal path quantification by a probability operator (as in PCTL) and by equipping the until-operator with a time bound (as in timed CTL):

$$\varphi ::= true \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathcal{P}_{\bowtie p}(\Psi) \qquad \Psi ::= \varphi \, \mathcal{U}^I \varphi$$

[1] Strictly speaking, we should write $unif_e(\mathcal{M})$ as the uniformization depends on e.

$[\![s, \mathit{true}]\!] = \top$	$[\![s, a]\!] = L(s, a)$
$[\![s, \varphi_1 \wedge \varphi_2]\!] = [\![s, \varphi_1]\!] \sqcap [\![s, \varphi_2]\!]$	$[\![s, \neg\varphi]\!] = [\![s, \varphi]\!]^c$
$[\![s, \mathcal{P}_{\bowtie p}(\Psi)]\!] = \top$, iff $\mathit{Prob}(\{\sigma \in \mathit{Paths}_s^{\mathcal{M}} \mid [\![\sigma, \Psi]\!] = \top\}) \bowtie p$	
$[\![\sigma, \varphi_1 \mathcal{U}^I \varphi_2]\!] = \top$, iff $\exists\, t \in I : ([\![\sigma@t, \varphi_2]\!] = \top \wedge\ \forall\, t' \in [0, t) : [\![\sigma@t', \varphi_1]\!] = \top)$	

Table 1. Semantics of CSL

where $\bowtie \in \{<, \leq, \geq, >\}, p \in [0, 1], I = [0, t) \,|\, [0, t] \,|\, [0, \infty)$ for $t \in \mathbb{R}_{>0}$ and $a \in AP$. φ is a *state-formula*, whereas Ψ is a *path-formula*. State formulas are ranged over by $\varphi, \psi, \ldots$ and path formulas are ranged over by $\Psi, \Phi, \ldots$.

A *path* in a CTMC is an alternating sequence $\sigma = s_0\, t_0\, s_1\, t_1\, s_2 \ldots$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ and $t_i \in \mathbb{R}_{>0}$ for all i. The time stamps t_i denote the amount of time spent in state s_i. $\sigma@t$ denotes the state of σ occupied at time t, i.e. $\sigma@t = s_i$ with i the smallest index such that $t < \sum_{j=0}^{i} t_j$. Let *Prob* denote the unique probability measure on sets of paths and let $\mathit{Paths}_s^{\mathcal{M}}$ denote the set of all paths of $\mathcal{M}$, starting in s. The subscript s is omitted when s is clear from the context; the same applies to superscript $\mathcal{M}$. Note that the probability measure of the set of infinite paths $s_0 t_0 s_1 t_1 \ldots$ with $\sum_{i=0}^{\infty} t_i$ is converging, is zero [3].

The semantics of CSL is given in Table 1. $\top$ and $\bot$ form a complete lattice such that $\bot < \top$ and *meet* $\sqcap$ as well as *complement* $\cdot^c$ are defined as usual.

Measures of interest can now be expressed as CSL formula in a convenient way. For example, the liveness property to reach a *down* state in a system within 52 time units, via *premium* states, with probability at most 0.01 would be formulated as $\mathcal{P}_{\leq 0.01}(\mathit{premium}\, \mathcal{U}^{[0,52]}\, \mathit{down})$. Another typical example would be to check, if some designated *goal* state is reachable at all times: $\mathcal{P}_{>0}(\mathit{true} \mathcal{U}^{[0,\infty)}\, \mathit{goal})$.

As in our abstraction, states may be grouped that satisfy distinct atomic propositions, we resort to a three-valued interpretation. Let $\mathbb{B}_3 = \{\bot, ?, \top\}$ with ordering $\bot < ? < \top$ and let $?^c = ?$. When a formula evaluates to $\bot$ or $\top$, the result is *definitely* true or false respectively, otherwise it is *indefinite*.

3 Abstraction

Our aim is to provide an abstraction of CTMCs which is conservative for both positive and negative verification results of CSL formulas. This is established by adopting a three-valued interpretation. The basic principle is to collapse sets of concrete states into single abstract states such that concrete states are simulated by abstract ones. As opposed to abstract interpretation only disjoint sets of concrete states are collapsed. That is, we consider a partitioning $\mathcal{A} = \{A_1, \ldots, A_n\}$ of the state space S of a CTMC $\mathcal{M} = (S, \mathbf{P}, E, L)$. The probability to evolve from abstract state A_i to A_j, $i, j \in \{1, \ldots, n\}$ within some time interval is represented by the functions: $\mathbf{P}(A_i, A_j) = \{\mathbf{P}(s, s', \cdot) \mid s \in A_i, s' \in A_j\}$.

Taking minimal and maximal probabilities as under- and over-approximation, respectively, suggests to define

$$\mathbf{P}^l(A_i, A_j, t) = \inf_{f \in \mathbf{P}(A_i, A_j)} f(t) \quad \text{and} \quad \mathbf{P}^u(A_i, A_j, t) = \sup_{f \in \mathbf{P}(A_i, A_j)} f(t).$$

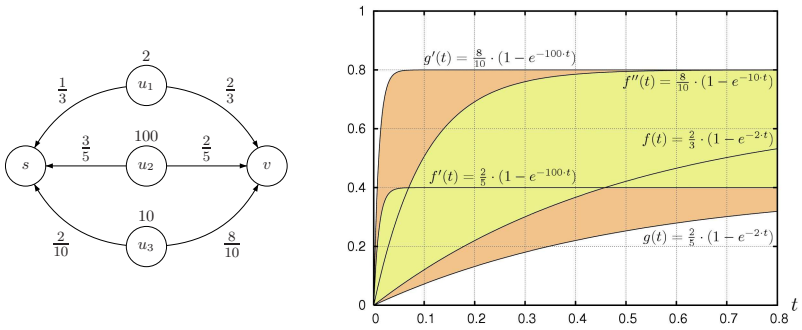

Fig. 1. Abstraction for non-uniform CTMCs

The functions $\mathbf{P}^l(A_i, A_j, t)$ and $\mathbf{P}^u(A_i, A_j, t)$ (considered as functions ranging over t) are in general not of the form $p{\cdot}(1 - e^{-E \cdot t})$ for fixed $p \in [0, 1]$ and $E > 0$.

Example 1. Consider the non-uniform CTMC $\mathcal{M} = (\{s, u_1, u_2, u_3, v\}, \mathbf{P}, E, L)$ in Fig. 1 (left). We focus on the transition probabilities of the states u_1, u_2, u_3 (indicated as labeled edges) and their exit rates which appear above the corresponding vertices. Further details of $\mathcal{M}$ are omitted in Fig. 1. Let $\mathcal{A} = \{A_s, A_u, A_v\}$ with $A_u = \{u_1, u_2, u_3\}$, $A_s = \{s\}$ and $A_v = \{v\}$. The set $\mathbf{P}(A_u, A_v) = \{f, f', f''\}$ is plotted in Fig. 1 (right). Note that $\mathbf{P}^l(A_u, A_v, t)$ and $\mathbf{P}^u(A_u, A_v, t)$ are not of the form $p{\cdot}(1 - e^{-E \cdot t})$. In general, the complexity of these functions grows when the number of transitions between states aggretated to A_u and A_v increases.

One might combine the infimum (supremum) of an abstract state's exit rates with the infimum (supremum) of the one-step transition probabilities to define an appropriate under- and over-approximation, yielding a rather coarse abstraction as indicated in Fig. 1 (right) which shows the plot of the functions g and g' resulting from this approach. But increasing the number of parameters to obtain a more accurate approximation results in a far too complex abstraction.

Therefore, we propose to abstract a CTMC by generating its uniformised CTMC (cf. Def. 3), and apply abstraction on the uniform CTMC, i.e., CTMCs in which all exit rates are equal to, say, E_{unif}. The advantage of uniform CTMCs is that $p_l{\cdot}(1 - e^{-E_{unif} \cdot t}) \leq p_u{\cdot}(1 - e^{-E_{unif} \cdot t})$ iff $p_l \leq p_u$ where p_l, p_u are the lower and upper bounds of time-abstract transition probabilities. Note that CTMC $\mathcal{M}$ and *unif*($\mathcal{M}$) are weak bisimilar, and as weak bisimulation preserves CSL equivalence[2] [4], the shift to the uniformized CTMC is correct for CSL. Our abstract model now becomes:

Definition 4 (Abstract CTMC). *An* abstract *CTMC (ACTMC for short) is a tuple* $\mathcal{M} = (S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$ *with a non-empty finite set of states* S*, transition probability functions* $\mathbf{P}^l, \mathbf{P}^u : S \times S \mapsto [0, 1]$ *such that* $\mathbf{P}^l(s, S) \leq 1 \leq \mathbf{P}^u(s, S)$ *componentwise for all* $s \in S$*.* $E_{unif} \in \mathbb{R}_{>0}$ *is the (global) exit rate for all states, and* $L : S \times AP \mapsto \mathbb{B}_3$ *is a labeling function.*

[2] Recall that we consider the fragment of CSL without the next-step operator.

An ACTMC $\mathcal{M}$ has a finite state space and is equipped with a pair of functions describing the lower and upper bound, respectively for the one-step transition probabilities. In contrast to CTMCs, states in an ACTMC may be labeled with ?. The set of transition probability functions is given by

$$\mathbf{P}_{\mathcal{M}} = \{\bar{\mathbf{P}} : S \times S \mapsto [0,1] \mid \mathbf{P}^l \leq \bar{\mathbf{P}} \leq \mathbf{P}^u \text{ and } \bar{\mathbf{P}}(s,S) = 1 \text{ for all } s \in S\},$$

where $\leq$ is to be interpreted element-wise. We may drop subscript $\mathcal{M}$ if $\mathcal{M}$ is clear from the context and write $\mathbf{P}(s,\cdot)$ for the set $\{\bar{\mathbf{P}}(s,\cdot) \mid \bar{\mathbf{P}} \in \mathbf{P}\}$.

An ACTMC $(S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$ with $\mathbf{P}^l = \mathbf{P}^u$ and $L(s,a) \in \mathbb{B}_2$ for any $s \in S$ and $a \in AP$ is a uniform CTMC.

Definition 5 (Abstraction). *For ACTMC $\mathcal{M} = (S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$, partitioning $\mathcal{A} = \{A_1, \ldots, A_n\}$ of S and $1 \leq i, j \leq n$, the* abstraction *of $\mathcal{M}$ induced by $\mathcal{A}$ is the ACTMC $abstr(\mathcal{A}, \mathcal{M}) := (\mathcal{A}, \tilde{\mathbf{P}}^l, \tilde{\mathbf{P}}^u, E_{unif}, \tilde{L})$ given by:*

- $\tilde{\mathbf{P}}^l(A_i, A_j) = \min_{s \in A_i} \mathbf{P}^l(s, A_j)$ *and* $\tilde{\mathbf{P}}^u(A_i, A_j) = \min\{1, \max_{s \in A_i} \mathbf{P}^u(s, A_j)\}$,
- $\tilde{L}(A_i, a) = \begin{cases} \top & \text{if } L(s,a) = \top \quad \text{for all } s \in A_i, a \in AP, \\ \bot & \text{if } L(s,a) = \bot \quad \text{for all } s \in A_i, a \in AP, \\ ? & \text{otherwise.} \end{cases}$

Example 2. Consider the CTMC in Fig. 2 (left) with exit rate 12, $AP = \{a\}$, $L(s_0,a) = L(s_1,a) = \top$ and $L(s_0',a) = L(s_2,a) = \bot$. The ACTMC induced by partition $\{\underbrace{\{s_0, s_0'\}}_{=A_0}, \underbrace{\{s_1\}}_{=A_1}, \underbrace{\{s_2\}}_{=A_2}\}$ is depicted in Fig. 2 (right) with $L(A_0,a) = ?$, $L(A_1,a) = \top$, $L(A_2,a) = \bot$.

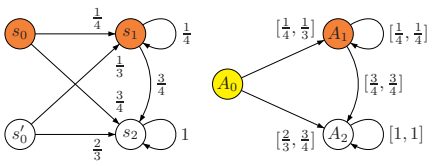

Fig. 2. Abstracting a CTMC

The probability to move from s to s' in an ACTMC may be any probability in $[\mathbf{P}^l(s,s'), \mathbf{P}^u(s,s')]$ and is chosen nondeterministically. As for Markov decision processes, *schedulers* are used to resolve nondeterminism. We consider (time-abstract) history-dependent schedulers that given a time-abstract path nondeterministically select a transition probability function from the set $\mathbf{P}$.

Definition 6 (Scheduler). *A* history-dependent scheduler *for ACTMC $\mathcal{M}$ is a function $D : Paths_{abs}(\mathcal{M}) \mapsto \mathbf{P}_{\mathcal{M}}$.*

Here, $Paths_{abs}(\mathcal{M})$ denotes the set of time-abstract paths in $\mathcal{M}$. A time-abstract path in $\mathcal{M}$ is a finite sequence of states $s_0 s_1 s_2 \ldots s_n$ such that $\bar{\mathbf{P}}(s_i, s_{i+1}) > 0$ for some $\bar{\mathbf{P}} \in \mathbf{P}$ for all $i \in \{0, 1, \ldots, n\}$. The set of history-dependent schedulers for ACTMC $\mathcal{M}$ is denoted by $Sched^{\mathcal{M}}$.

If only lower and upper bounds on transition probabilities are given, it may happen that not every combination is possible. For instance, in Example 2, a possible choice in state A_0 is to select A_1 with $\frac{1}{4}$ and A_2 with $\frac{2}{3}$, but $\frac{1}{4} + \frac{2}{3} < 1$.

Definition 7 (Delimited ACTMC). *An ACTMC $\mathcal{M} = (S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$ is* delimited *iff for any $s, s' \in S$ and $p \in [\mathbf{P}^l(s,s'), \mathbf{P}^u(s,s')]$, there exists $\bar{\mathbf{P}} \in \mathbf{P}$ with $\bar{\mathbf{P}}(s,s') = p$.*

In the following, we define an operation, called *cut*, that transforms a given ACTMC into a delimited one. It basically strips off combinations of probabilities in the intervals that do not yield transition probabilities. A similar function has been defined for abstractions of DTMCs (see Def. 11) in [11].

Definition 8 (Cut). *Let $\mathcal{M} = (S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$ be an ACTMC. We define the functions cut$(\mathbf{P}^l, \mathbf{P}^u) = (\tilde{\mathbf{P}}^l, \tilde{\mathbf{P}}^u)$ by $\tilde{\mathbf{P}}^l(s, s') = \max\{\mathbf{P}^l(s, s'), 1 - \mathbf{P}^u(s, S \setminus \{s'\})\}$ and $\tilde{\mathbf{P}}^u(s, s') = \min\{\mathbf{P}^u(s, s'), 1 - \mathbf{P}^l(s, S \setminus \{s'\})\}$ for all $s, s' \in S$.*

The cut *of $\mathcal{M}$ is defined as cut$(\mathcal{M}) = (S, \tilde{\mathbf{P}}^l, \tilde{\mathbf{P}}^u, E_{unif}, L)$.*

Lemma 1. *For ACTMC $\mathcal{M}$, cut$(\mathcal{M})$ is delimited and $Sched^{\mathcal{M}} = Sched^{cut(\mathcal{M})}$.*

A finite subset of the transition probability distributions, which will prove useful when considering lower and upper bounds of reachability properties, is the set of extreme distributions. Intuitively they result from a one by one minimisation/maximisation of transition probabilities. Note that different priorities for minimising/maximising yield different minimal/maximal probabilities. Actually, the number of extreme distributions grows exponentially in the state space size.

Definition 9 (Extreme distributions). *Let $s \in S$ and $S' \subseteq S$. We define $extr(\mathbf{P}^l, \mathbf{P}^u, S', s) \subseteq \mathbf{P}$ such that $\mu \in extr(\mathbf{P}^l, \mathbf{P}^u, S', s)$ iff either $S' = \emptyset$ and $\mu = \mathbf{P}^l(s, \cdot) = \mathbf{P}^u(s, \cdot)$ or one of the following conditions is true*[3]

$$\exists s' \in S' : \mu(s') = \mathbf{P}^l(s, s') \text{ and } \mu \in extr(\mathbf{P}^l, \mathbf{P}^u[s' \mapsto \mu(s')], S' \setminus \{s'\}, s)$$
$$\exists s' \in S' : \mu(s') = \mathbf{P}^u(s, s') \text{ and } \mu \in extr(\mathbf{P}^l[s' \mapsto \mu(s')], \mathbf{P}^u, S' \setminus \{s'\}, s)$$

We call $\mu \in \mathbf{P}(s, \cdot)$ an extreme distribution *if $\mu \in extr(\mathbf{P}^l, \mathbf{P}^u, S, s)$.*

To compare the behavior described by two ACTMCs, we introduce the notion of probabilistic simulation which is a variant of probabilistic simulation for CTMCs as it can be found in [4].

Definition 10 (Probabilistic simulation). *Let $\mathcal{M} = (S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$ be an ACTMC. We call $\mathcal{R} \subseteq S \times S$ a* probabilistic simulation *iff $s\mathcal{R}s'$ implies:*

1. *$L(s', a) \neq ? \Rightarrow L(s', a) = L(s, a)$ for all $a \in AP$.*
2. *For all distributions $\mu \in \mathbf{P}(s, \cdot)$, there is a distribution $\mu' \in \mathbf{P}(s', \cdot)$ and a weight function $\Delta : S \times S \to [0, 1]$ with:*
   (a) *$\Delta(u, v) > 0 \Rightarrow u\mathcal{R}v$,* (b) *$\Delta(u, S) = \mu(u)$,* (c) *$\Delta(S, v) = \mu'(v)$.*

State s is simulated by s' (written $s \preceq s'$) if there exists a probabilistic simulation $\mathcal{R}$ with $(s, s') \in \mathcal{R}$. We lift $\preceq$ to the union of two ACTMCs in the usual way.

Theorem 1. *For ACTMC $\mathcal{M}$ with state space S, and $\mathcal{A}$ a partitioning on S inducing the ACTMC abstr$(\mathcal{A}, \mathcal{M})$ with state space $\mathcal{A}$*

$$s \in A \Rightarrow s \preceq A \text{ for all } s \in S, A \in \mathcal{A}$$

[3] Here, $f[s \mapsto x]$ denotes the function that agrees everywhere with f except at s where it is equal to x.

Example 3. Consider the CTMC in Fig. 2(a), the partitioning leading to 2(b) (see Ex. 2) with $\mathcal{R} = \{(s_0, A_0), (s_0', A_0), (s_1, A_1), (s_2, A_2)\}$. Note that A_i should be considered as a single abstract state. We have $s_0 \mathcal{R} A_0$ because condition 1 of Def. 10 is trivially fulfilled since $L(A_0, a) = ?$. For condition 2 we observe that in s_0 there is only one possible distribution $\mu = (0, 0, \frac{1}{4}, \frac{3}{4})$ to choose. The only distribution in $\mathbf{P}(A_0, \cdot)$, for which there is a weight function Δ fulfilling condition 2, is $\mu' = (0, \frac{1}{4}, \frac{3}{4})$ with $\Delta(s_1, A_1) = \frac{1}{4}$, $\Delta(s_2, A_2) = \frac{3}{4}$ and 0 otherwise. The conditions of Def. 10 can be checked for the remaining elements of $\mathcal{R}$ similarly.

In the following we show that our abstraction of CTMCs can be regarded as a conservative extension of abstraction of DTMCs as recently proposed in [11].

Definition 11 (Abstract DTMC). *An abstract DTMC (ADTMC) is a tuple $(S, \mathbf{P}^l, \mathbf{P}^u, L)$ with S, $\mathbf{P}^l$, $\mathbf{P}^u$, and L as before.*

Abstract DTMCs are thus abstract CTMCs without exit rates. The theorem below shows that the following diagram commutes:

$$\begin{array}{ccccccc} \mathcal{M} & \xrightarrow{abstr.} & \mathcal{M}_{abstr} & \xrightarrow{cut} & \mathcal{M}_{del} & & \} \text{ (A)CTMCs} \\ \downarrow embedded & & & & \downarrow embedded & & \\ \mathcal{N} & \xrightarrow{abstr.} & \mathcal{N}_{abstr} & \xrightarrow{cut} & \mathcal{N}_{del} & & \} \text{ (A)DTMCs} \end{array}$$

Theorem 2. *For delimited uniform CTMC $\mathcal{M}$ and partitioning $\mathcal{A}$:*

$$emb(cut(abstr(\mathcal{A}, \mathcal{M}))) = cut_{ADTMC}(abstr_{DTMC}(\mathcal{A}, emb(\mathcal{M})))$$

where cut_{ADTMC} and $abstr_{\text{DTMC}}$ are the counterparts of *cut* and *abstr* in the setting of (A)DTMCs [11].

4 Model Checking Three-valued CSL

Now, we develop a three-valued version of CSL which is interpreted over ACTMCs. The simulation relation allows us to reason about more concrete systems.

For an ACTMC $\mathcal{M}$, every scheduler $D \in Sched^{\mathcal{M}}$ induces a probability space with a probability measure $Prob^D$ in the same manner as for CTMCs (see [3] for details). When interested in the infimum of probabilities of measurable sets with regard to all schedulers, it suffices to consider only extreme distributions. A scheduler which only chooses such distributions is an *extreme* scheduler. The set of all extreme schedulers for $\mathcal{M}$ is denoted as $Sched^{\mathcal{M}}_{extr}$.

Theorem 3. *Let $\mathcal{M} = (S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$ be an ACTMC. For every measurable set Q of the induced probability space:*

$$\inf_{D \in Sched^{\mathcal{M}}_{extr}} Prob^D(Q) = \inf_{D \in Sched^{\mathcal{M}}} Prob^D(Q).$$

The proof for the above theorem is rather technical and goes along the structure of the generated Borel field of the induced probability space. Note that the number of choices at a state is finite for extreme schedulers, whereas this is uncountable for arbitrary schedulers.

Before discussing CSL, let us first consider *time-dependent reachability probabilities* in ACTMCs, i. e., the probabilities to reach some state in set B within t time units, formally $Reach_{\leq t}(s,B) = \{\sigma \in Paths_s^{\mathcal{M}} \mid \sigma@t' \in B \text{ for some } t' \in [0,t]\}$. When computing the semantics of CSL formulas, the main challenge is to determine lower bounds of reachability properties, as we will see. Therefore, we will now analyse how to compute $\inf_{D \in Sched^{\mathcal{M}}} Prob^D(Reach_{\leq t}(s,B))$. $Prob^l(Q)$ will be used as abbreviation for $\inf_{D \in Sched^{\mathcal{M}}} Prob^D(Q)$.

We start with an algorithm for the approximation of probability bounds for timed reachability properties in uniform CTMDPs (see [2]). By Theorem 3, it suffices to consider extreme schedulers if one is interested in lower bounds. We interpret an ACTMC as a CTMDP, where each extreme distribution can be chosen by some action. From [2], we know that an ε-approximation of transient probabilities q_0 can efficiently be computed in an iterative way[4]:

$$
\begin{aligned}
q_0 &= \psi_{E_{unif},t}(0) \cdot i_B + q_1 \\
q_i &= \psi_{E_{unif},t}(i) \cdot i_B + \mathbf{P}_i \cdot q_{i+1}, \text{ for } i \in \{1, ..., k(\varepsilon, E_{unif}, t)\}, \\
q_{k(\varepsilon,E_{unif},t)+1} &= \underline{0}, \text{ where } k(\varepsilon, E_{unif}, t) \text{ is a proper truncation point,} \\
&\quad \text{and } \psi_{E_{unif,t}}(n) \text{ is the probability that} \\
&\quad n \text{ events occur in a Poisson process of rate } E_{unif \cdot t}
\end{aligned}
$$

Therefore, instead of checking for all extreme distributions in each iteration, we can find a minimizing distribution in polynomial time, by minimizing the vector-product $\mathbf{P}_i(s,\cdot) \cdot q_{i+1}$ with additional constraint $q_{i+1}(S) = 1$. This can be done by successively assigning as much proportion as possible to the transition leading to the successor s' for which $q_{i+1}(s')$ is minimal. For $N := |S|$, sorting the q-vector can be done in $\mathcal{O}(N \log(N))$ and assertion of probabilities takes $O(N^3)$ since the cut has to be applied N times and the *cut* itself has a complexity of $\mathcal{O}(N^2)$. This yields a worst-case complexity of $\mathcal{O}(N^2 \cdot (N \log(N) + N^3) + N) = \mathcal{O}(N^5)$ for every iteration step.

The following theorem, which states that the above algorithm yields an ε-accurate approximation of reachability properties, follows directly from [2].

Theorem 4. *For ACTMC $\mathcal{M} = (S, \mathbf{P}^l, \mathbf{P}^u, E_{unif}, L)$, $s \in S$, $B \subseteq S$, $t \in \mathbb{R}_{>0}$ and error margin ε:*

$$Prob^l(Reach_{\leq t}(s,B)) - \varepsilon \leq q_0(s) \leq Prob^l(Reach_{\leq t}(s,B))$$

Three-valued CSL-semantics. We define the satisfaction function $[\![\cdot]\!] : (S \cup Paths^{\mathcal{M}}) \times \text{CSL} \to \mathbb{B}_3$ inductively as shown in Table 2, where $Prob^l(s, \Phi, \alpha) = Prob^l(\{\sigma \in Paths_s^{\mathcal{M}} \mid [\![\sigma, \Phi]\!] = \alpha\})$ for $\alpha \in \mathbb{B}_3$.

Let us have a closer look at the semantics. For the propositional fragment the semantics is clear. A path σ satisfies until-formula $\varphi_1 \mathcal{U}^{[0,t]} \varphi_2$ if φ_1 holds for sure until φ_2 holds for sure at the latest at time t. The until-formula $\varphi_1 \mathcal{U}^{[0,t]} \varphi_2$ is violated, if either before φ_2 holds, φ_1 is violated, or if φ_2 is violated for sure. Otherwise, the result is indefinite.

[4] The truncation point $k(\varepsilon, E_{unif}, t)$ depends linearly on E_{unif} and t and can easily be computed on-the-fly.

$$[\![s, true]\!] = \top \qquad [\![s, false]\!] = \bot \qquad [\![s, a]\!] = L(s,a)$$
$$[\![s, \varphi_1 \wedge \varphi_2]\!] = [\![s, \varphi_1]\!] \sqcap [\![s, \varphi_2]\!] \qquad [\![s, \neg\varphi]\!] = [\![s, \varphi]\!]^c$$

$$[\![\sigma, \varphi_1 \mathcal{U}^I \varphi_2]\!] = \begin{cases} \top & \text{if } \exists\, t \in I : ([\![\sigma@t, \varphi_2]\!] = \top \wedge\ \forall\, t' \in [0,t) : [\![\sigma@t', \varphi_1]\!] = \top) \\ \bot & \text{if } \forall\, t \in I : ([\![\sigma@t, \varphi_2]\!] = \bot \vee\ \exists\, t' \in [0,t) : [\![\sigma@t', \varphi_1]\!] = \bot) \\ ? & \text{otherwise} \end{cases}$$

$$[\![s, \mathcal{P}_{\trianglerighteq p}(\Psi)]\!] = \begin{cases} \top & \text{if } \mathit{Prob}^l(s, \Psi, \top) \trianglerighteq p \\ \bot & \text{if } \mathit{Prob}^l(s, \Psi, \bot) \rhd 1-p \\ ? & \text{otherwise} \end{cases} \qquad \trianglerighteq \in \{>, \geq\}, \rhd = \begin{cases} > \text{ if } \trianglerighteq = \geq \\ \geq \text{ if } \trianglerighteq = > \end{cases}$$

$$[\![s, \mathcal{P}_{\trianglelefteq p}(\Psi)]\!] = \begin{cases} \top & \text{if } 1-p \trianglelefteq \mathit{Prob}^l(s, \Psi, \bot) \\ \bot & \text{if } \quad p \lhd \mathit{Prob}^l(s, \Psi, \top) \\ ? & \text{otherwise} \end{cases} \qquad \trianglelefteq \in \{<, \leq\}, \lhd = \begin{cases} < \text{ if } \trianglelefteq = \leq \\ \leq \text{ if } \trianglelefteq = < \end{cases}$$

Table 2. Three-valued semantics of CSL.

To determine the satisfaction of $\mathcal{P}_{\leq p}(\Psi)$ we consider the probability of the paths for which Ψ is surely violated. If this probability is greater than $1-p$, then paths where Ψ holds may have measure at most p. Similarly, to show that $\mathcal{P}_{\leq p}(\Psi)$ is violated, we have to consider the measure of all paths surely satisfying Ψ. If this measure is greater than p, then obviously $\mathcal{P}_{\leq p}(\Psi)$ is violated. The semantics of $\mathcal{P}_{\trianglelefteq p}(\Psi)$ for $\trianglelefteq \in \{<, >, \geq\}$ follows from a similar argumentation.

Example 4. Consider the CTMC in Fig. 2(a). Starting in s_0 (s_1), the probability to reach a non-a-state in 0.3 time units is about 0.9037 (0.9328, respectively). Thus, formula $\varphi = a \rightarrow \mathcal{P}_{\geq 0.9}(true\,\mathcal{U}^{\leq 0.3}\neg a)$ is true in all states. Consider the abstraction in Fig. 2(b): The lower and upper probability bounds to reach a non-a-state in 0.3 time units from A_0 are about 0.8807 respectively 0.9037. Hence, $[\![A_0, a \rightarrow \mathcal{P}_{\geq 0.9}(true\,\mathcal{U}^{\leq 0.3}\neg a)]\!] = ? \sqcup [\![t_0, \mathcal{P}_{\geq 0.9}(true\,\mathcal{U}^{\leq 0.3}\neg a)]\!] = ? \sqcup ? = ?$. For $\mathcal{P}_{\geq 0.88}$ instead of $\mathcal{P}_{\geq 0.9}$, the formula would have been satisfied in the abstraction as well, while for $\mathcal{P}_{\geq 0.91}$ the result would still be ? since $? \sqcup \bot = ?$.

The following theorem states that our framework developed so far can indeed be used for abstraction based model checking. It can be shown by structural induction on the CSL formulas. Intuitively, the theorem asserts that the result of checking a CSL formula in the abstract CTMC agrees with the one for the more concrete model, unless it is indefinite.

Theorem 5 (Preservation of CSL). *Let s and s' be two states of an ACTMC $\mathcal{M}$ with $s \preceq s'$. Then for all CSL formulas φ:*

$$[\![s', \varphi]\!] \neq ? \ \textit{ implies } [\![s, \varphi]\!] = [\![s', \varphi]\!].$$

Observe that the 3-valued CSL semantics on a CTMC (viewed as ACTMC) coincides with the 2-valued CSL semantics for CTMCs (see Section 2), showing that our abstraction is *conservative* for positive and negative verification results.

Model checking. As for CTL, model checking works bottom-up the parse tree of the CSL formula φ. Boolean combinations of formulas as well as the $\mathcal{P}$-formulas are evaluated, as expected. For the latter, however, we need the lower probability

bounds for the satisfaction/violation of an until-formula, which remains the only operator to discuss.

The idea of dealing with until-operators is similar as in [11]: For getting the measure of paths surely satisfying $\Psi = \varphi_1 \, \mathcal{U}^{[0,t]} \varphi_2$, it suffices to compute the measure of reaching states satisfying φ_2 in time bounded by t along paths of states satisfying φ_1. By induction, we know which states do not satisfy φ_1. Removing these from the CTMC, a path satisfies $\varphi_1 \mathcal{U}^{[0,t]} \varphi_2$ iff a state φ_2 is reached within time bound t. In other words, it remains to solve a time-bounded reachability problem in the reduced graph. Getting the measure of paths violating Ψ for sure, is done similarly by exchanging $\top$ and $\bot$ in the argumentation above.

Recall that the given algorithm for computing time-bounded reachability approximates only with error margin ε. However, it can easily be guaranteed that the error due to approximation only yields `?` in cases where a definite value could be obtained given a smaller error margin.

Theorem 6. *Given an ACTMC $\mathcal{M}$, a CSL formula φ, and an error margin ε, we can approximate $[\![\mathcal{M}, \varphi]\!]$ in time polynomial in size of $\mathcal{M}$ and linear in size of φ, E_{unif} and the highest time bound t occurring in φ (dependency on ε is omitted as ε is linear in $E_{unif} \cdot t$). In case the approximation yields $\top$ or $\bot$, the result is correct, while ? is correct with an error of at most ε.*

5 Case study: Quasi-Birth-Death Processes

Let us consider a simple system with a fixed number m of available processors and an infinite queue for storing job requests. The processing speed of the processors is described by an exponential distribution with rate γ and λ is the incoming rate of jobs. When all processors are being utilized, new jobs are added to the infinite queue. As soon as processors are getting available again, jobs from the queue are processed. To model these spontaneous transitions, we choose a high rate $\epsilon \gg \lambda$. In our experiments about 10 times the incoming rate for tasks has been a sufficiently precise approximation. The system initially has no job to process, i.e. all three processors are available and the queue is empty. For $m = 3$, this is being formally described by the *stochastic Petri net* (SPN) [5] in Fig. 3(a). Numbers at edges denote that the corresponding transition consumes or produces the given number of tokens and can not be fired until there are enough token to consume. The semantics of this SPN is equal to an infinite CTMC. Uniformization with rate E results in the infinite uniform CTMC (Fig. 3(b)). For $E, x, y, z \in \mathbb{R}_{\geq 0}$, we shortly write $E^x_{yz} = E^x_y - z, E^x_y = E^x - y$ and $E^x = E - x$. State $s_{i,j}$ represents

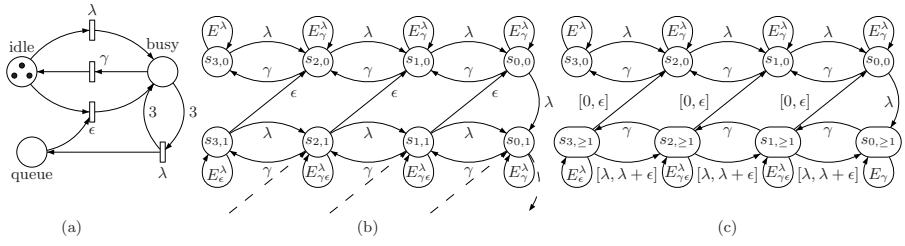

Fig. 3. (a) SPN, (b) uniformized underlying infinite CTMC, (c) finite abstraction

the marking of the SPN, where i tokens are at *idle*, $m-i$ at *busy* and j at *queue*. Aggregating $\{s_{i,j} \mid j \geq n\}$ by $s_{i,\geq n}$ for all $i \in \{0,...,m\}$ yields Fig. 3(c) ($n = 1$).

Consider $\varphi = (\langle l_1 = 0\rangle \wedge \langle l_2 = 0\rangle) \rightarrow \mathcal{P}_{\leq p}(true\, \mathcal{U}^{[0,t]}(\langle l_1 = m\rangle \wedge \langle l_2 = 0\rangle))$ where $\langle l_1 = i\rangle$, $\langle l_2 = j\rangle \in AP$ hold in all states $s_{i,j}$ of the infinite CTMCs.

In Fig. 4, for $\lambda \in \{1,...,6\}$, lower and upper probability bounds for φ for abstractions with $n \in \{1,...,9\}$ are plotted. As expected, by increasing n, lower and upper bounds are closer, i.e. the accuracy of the abstraction improves.

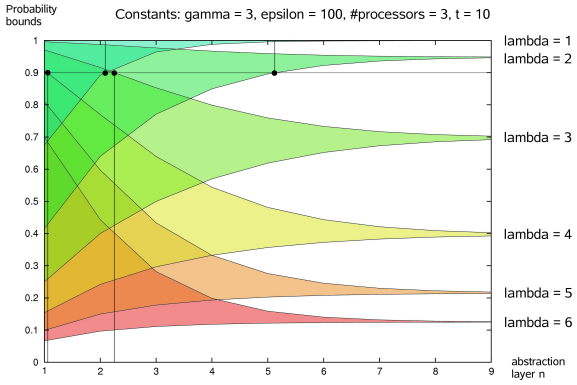

Fig. 4. Probability bounds for φ.

Increasing m improves the system performance. The probability for which φ holds decreases for increasing m. If the system is upgraded with m' additional processors then the requirement is not about m jobs anymore, but about $m + m'$. Note that CSL model-checking algorithms for quasi-birth-death processes have also been considered in [21]. Our abstraction technique, though, is not restricted to these (regular) infinite CTMCs.

6 Conclusion

This paper presented a three-valued abstraction technique for CTMCs that is conservative for true and false results of CSL. The idea is to abstract uniform CTMCs and replace transition probabilities by intervals. A polynomial-time approximative model-checking algorithm for 3-valued CSL has been provided.

Although our approach intends to combat the state-space explosion problem, model checking of probabilistic interval models is of interest in its own, when the exact values are not known and e.g., estimated by experiments, cf. [22].

Our experiments indicate that—like for most other abstraction techniques—the partitioning of the state space determines the accuracy of the abstraction; e.g., merging "slow" and "fast" states typically yields too coarse abstractions. To conduct more experiments, we currently incorporate the abstraction into the model checker MRMC [17]. Besides we plan to work on refinement techniques to improve abstractions when the verification yields indefinite results.

References

1. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model-checking continuous time Markov chains. *ACM TOCL* **1** (2000) 162–170
2. Baier, C., Hermanns, H., Katoen, J. P., Haverkort, B. R. H. M.: Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *TCS* **345** (2005) 2–26

3. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. *IEEE TSE* **29** (2003) 524–541
4. Baier, C., Katoen, J.-P., Hermanns, H., Wolf, V.: Comparative branching-time semantics for Markov chains. *Information and Computation* **200** (2005) 149–214
5. Bause, F., Kritzinger, P. S.: Stochastic Petri nets: An introduction to the theory. *SIGMETRICS Performance Evaluation Review* **26** (1998)
6. Ben Mamoun, M., Pekergin, N., Younès, S.: Model checking of continuous-time Markov chains by closed-form bounding distributions. In: *QEST*. IEEE CS (2006) 189–198
7. Böde, E., Herbstritt, M., Hermanns, H., Johr, S., Peikenkamp, T., Pulungan, R., Wimmer, R., Becker, B.: Compositional performability evaluation for STATEMATE. In: *QEST*. IEEE CS (2006) 167–178
8. Ciardo, G., III, R. L. J., Miner, A., Siminiceanu, R.: Logical and stochastic modeling with SMART. In: *Comp. Perf. Ev. LNCS*, Vol. 2794. (2003) 78–97
9. D'Aprile, D., Donatelli, S., Sproston, J.: CSL model checking for the GreatSPN tool. In: *Computer and Information Sc., ISCIS. LNCS*, Vol. 3280. (2004) 543–553
10. D'Argenio, P. R., Jeannet, B., Jensen, H. E., Larsen, K. G.: Reachability analysis of probabilistic systems by successive refinements. In: *PAPM–PROBMIV. LNCS*, Vol. 2165., Berlin (2001) 39–56
11. Fecher, H., Leucker, M., Wolf, V.: Don't know in probabilistic systems. In: *Model Checking Software. LNCS*, Vol. 3925., Berlin (2006) 71–88
12. Gilmore, S., Hillston, J.: The PEPA workbench: A tool to support a process algebra-based approach to performance modelling. In: *Computer Performance Evaluation. LNCS*, Vol. 794. (1994) 353–368
13. Groesser, M., Baier, C.: Partial order reduction for Markov decision processes: a survey. In de Boer, F. S., Bonsangue, M. M., Graf, S., de Roever, W.-P. (eds.): *FMCO. LNCS*, Vol. 4111. (2006) 408–427
14. Huth, M.: An abstraction framework for mixed non-deterministic and probabilistic systems. In: *Validation of Stoch. Systems. LNCS*, Vol. 2925. (2004) 419–444
15. Huth, M.: On finite-state approximants for probabilistic computation tree logic. *TCS* **346** (2005) 113–134
16. Katoen, J.-P., Kemna, T., Zapreev, I., Jansen, D. N.: Bisimulation minimisation mostly speeds up probabilistic model checking. In: *TACAS. LNCS*, Vol. 4424. (2007) 87–102
17. Katoen, J.-P., Khattri, M., Zapreev, I. S.: A Markov reward model checker. In: *QEST*. IEEE CS (2005) 243–244
18. Kwiatkowska, M., Norman, G., Parker, D.: Game-based abstraction for Markov decision processes. In: *QEST*. IEEE CS (2006) 157–166
19. Kwiatkowska, M., Norman, G., Parker, D.: Symmetry reduction for probabilistic model checking. In: *CAV. LNCS*, Vol. 4144., Berlin (2006) 234–248
20. Monniaux, D.: Abstract interpretation of programs as Markov decision processes. *Science of Computer Programming* **58** (2005) 179–205
21. Remke, A., Haverkort, B. R., Cloth, L.: Model checking infinite-state Markov chains. In: *TACAS. LNCS*, Vol. 3440. (2005) 237–252
22. Sen, K., Viswanathan, M., Agha, G.: Model-checking Markov chains in the presence of uncertainties. In: *TACAS. LNCS*, Vol. 3920., Berlin (2006) 394–410
23. Sproston, J., Donatelli, S.: Backward bisimulation in Markov chain model checking. *IEEE TSE* **32** (2006) 531–546
24. Zhang, L., Hermanns, H., Eisenbrand, F., Jansen, D. N.: Flow faster: efficient decision algorithms for probabilistic simulations. In: *TACAS. LNCS*, Vol. 4424. (2007) 155–170