Statistical Verification of Probabilistic Properties with Unbounded Until

Håkan L. S. Younes[1], Edmund M. Clarke[2], and Paolo Zuliani[2]

[1] Google Inc
[2] Computer Science Department, Carnegie Mellon University, USA

Abstract. We consider statistical (sampling-based) solution methods for verifying probabilistic properties with unbounded until. Statistical solution methods for probabilistic verification use sample execution trajectories for a system to verify properties with some level of confidence. The main challenge with properties that are expressed using unbounded until is to ensure termination in the face of potentially infinite sample execution trajectories. We describe two alternative solution methods, each one with its own merits. The first method relies on reachability analysis, and is suitable primarily for large Markov chains where reachability analysis can be performed efficiently using symbolic data structures, but for which numerical probability computations are expensive. The second method employs a termination probability and weighted sampling. This method does not rely on any specific structure of the model, but error control is more challenging. We show how the choice of termination probability—when applied to Markov chains—is tied to the subdominant eigenvalue of the transition probability matrix, which relates it to iterative numerical solution techniques for the same problem.

1 Introduction

Probabilistic model checking deals with verification of stochastic systems, such as a queuing system with random arrivals and departures. Temporal stochastic logics, e.g., PCTL [12] and CSL [1], exist for expressing properties of such stochastic systems. Our focus is on *time-unbounded* properties of stochastic systems. For a queuing system, for instance, an interesting property could be: "the probability is at most 0.1 that the queue eventually becomes full." In PCTL, such properties are expressed using the formula $\mathcal{P}_{\leq 0.1}[\top\ \mathcal{U}\ \text{full}]$.

We present two statistical algorithms for solving such model-checking problems that are based on *unbiased* sampling. Sampling is said to be unbiased if the expectation of the sample distribution is the same as the expectation of the true distribution. The use of unbiased sampling distinguishes our methods from most recent efforts to devise sampling-based algorithms for time-unbounded properties, which are based on *biased* sampling [21, 18, 27, 3] (see Sect. 4).

Statistical algorithms for probabilistic model checking use discrete-event simulation to generate sample trajectories, and verify some temporal formula over each generated trajectory. This is combined with hypothesis testing or statistical

estimation to verify probabilistic properties [26, 18]. The challenge for statistical algorithms with time-unbounded properties is to determine the truth-value of $\Phi \,\mathcal{U}\, \Psi$ without generating infinite sample trajectories.

The first method (see Sect 3.1) combines reachability analysis with statistical sampling. This approach has been used in the past for program analysis [20], and more recently for model checking [27] using biased sampling. The algorithm is ensured termination for any finite-state homogeneous discrete-time Markov chain, although it is potentially applicable for any model for which we can perform reachability. The use of reachability analysis requires that we construct the full model, so it may seem counter to the appeal of statistical methods, which usually avoid model construction. The real cost in probabilistic model checking, however, lies in the numerical computation of probabilities, which we replace with sampling. We show in Sect. 5 that the combination of reachability analysis and statistical sampling scales better with the size of the model than standard numerical solution methods. As a result, we can verify time-unbounded properties for larger models than possible with existing numerical algorithms. By using unbiased sampling, we can also make strong guarantees regarding error bounds. Other sampling-based methods, as well as iterative numerical solution methods, do not give the same strong guarantees as they depend on heuristics for bounding sample trajectory lengths or number of iterations.

The second method (see Sect. 3.2) is based on a Monte Carlo method devised in the 1940s by John von Neumann and Stanislaw Ulam for computing the inverse of a matrix [10]. This method uses a termination probability p_{T} that is applied in each state along a trajectory to ensure finite sample trajectories. To account for the change in sample distribution, we weigh satisfying trajectories more heavily the longer the trajectory is. This way, we obtain an unbiased estimator of the probability that $\Phi \,\mathcal{U}\, \Psi$ holds over the set of trajectories that start in some state s. The second method does not rely on reachability analysis, so it has minimal memory requirements. It generally requires a larger number of sample trajectories to achieve the same precision as the first method, so it can be slower than the first method when reachability analysis is fast. It also suffers from the same problem as iterative numerical solution methods in that accuracy can be hard to guarantee. Still, the second method is potentially applicable for a much larger class of models.

We limit our attention to discrete-time Markov chains. The results extend trivially to continuous-time Markov chains and semi-Markov processes, as verification of time-unbounded properties for such models is done on an embedded discrete-time Markov chain.

2 Probabilistic Model Checking

This section describes discrete-time Markov chains (without nondeterminism), which is the class of models that we consider for probabilistic model checking. We present a temporal stochastic logic (a subset of PCTL) and discuss realistic error control for statistical model-checking algorithms.

2.1 Stochastic Processes and Discrete-Time Markov Chains

The terminology introduced here follows that of Stewart [22]. A *stochastic process* with state space S and time domain T is a family of random variables $\mathcal{X} = \{X_t \mid t \in T\}$. A random variable $X_t \in \mathcal{X}$ represents the outcome of observing the state of the stochastic process at time t.

A *discrete-time Markov process* is a stochastic process where T is the nonnegative integers, $\mathbb{Z}^*$, and

$$\Pr[X_{n+1} = s_{n+1} \mid X_0 = s_0, \ldots, X_n = s_n] = \Pr[X_{n+1} = s_{n+1} \mid X_n = s_n] \quad (1)$$

holds for all $n \in \mathbb{Z}^*$ and $s_i \in S$. If the state space is discrete as well, then we refer to the process as a *discrete-time Markov chain*. We will limit our attention to discrete-time Markov chains. The techniques we present later on can be generalized to other types of stochastic processes, but it is beyond the scope of this paper.

Let $p_{ij}(n) = \Pr[X_{n+1} = j \mid X_n = i]$, which denotes the probability of transitioning from state i at time n to state j at time $n+1$. We call $p_{ij}(n)$, for all i and j in S and all $n \in \mathbb{Z}^*$, the transition probabilities of the discrete-time Markov chain. We have $p_{ij}(n) \in [0,1]$ and, for all $i \in S$, $\sum_{j \in S} p_{ij}(n) = 1$. If, in addition to (1), we have $p_{ij}(n) = p_{ij}(m)$ for all n and m in $\mathbb{Z}^*$, then the discrete-time Markov chain is called *homogeneous*. In a homogeneous Markov chain, transition probabilities are independent of time. The transition probabilities of a finite-state homogeneous discrete-time Markov chain can be represented by a single $|S| \times |S|$ transition probability matrix $\mathbf{P}$. For notational convenience, we will use $\mathbf{P}$ to represent the collection of transition probabilities, $p_{ij}(n)$, for nonhomogeneous Markov chains as well.

The evolution of a discrete-time Markov chain over time is captured by a *trajectory*. The trajectory of such a system is a sequence of states $\sigma = s_0 \rightarrow s_1 \rightarrow \cdots$, with $s_i \in S$. We denote by $\sigma[i]$ the ith state, s_i, along the trajectory σ, and the finite prefix of length n of σ is denoted $\sigma{\uparrow}n$.

Let $Path(s)$ denote the set of trajectories with initial state s. Following Hansson and Jonsson [12], we define a probability measure μ on the set $Path(s)$, for each $s \in S$. The measure μ is defined on the probability space $\langle \Omega, \mathcal{F}_\Omega \rangle$, where $\Omega = Path(s)$, and $\mathcal{F}_\Omega$ is a σ-algebra generated by sets $\{\sigma \in Path(s) \mid \sigma{\uparrow}n = s \rightarrow s_1 \rightarrow \cdots \rightarrow s_n\}$ of trajectories with common finite prefix of length n. The measure μ can then be defined uniquely by induction on the length of the common prefix as follows:

$$\mu(\{\sigma \in Path(s) \mid \sigma{\uparrow}0 = s\}) = 1 \quad (2)$$

$$\mu(\{\sigma \in Path(s) \mid \sigma{\uparrow}n = s \rightarrow s_1 \rightarrow \cdots \rightarrow s_n\}) = \mu(\{\sigma \in Path(s) \mid \sigma{\uparrow}(n-1) = s \rightarrow \cdots \rightarrow s_{n-1}\}) \cdot p_{s_{n-1}s_n}(n-1) \quad (3)$$

2.2 Temporal Stochastic Logic

We use the Probabilistic Computation Tree Logic (PCTL [12]) to specify properties of discrete-time Markov chains. We describe only a subset of PCTL that

includes unbounded until, since that is the focus of this paper. The techniques described later in the paper can of course be combined with the techniques developed by Younes and Simmons [26] to handle a more expressive logic. The logic permits nested probabilistic operators, although we do not discuss such formulae here. Younes and Simmons [26] have already shown how to deal with nested probabilistic formulae without ties to any specific type of path formula. We could also replace PCTL with probabilistic LTL, which would avoid nested probabilistic operators altogether. The solution methods of this paper can be adapted for probabilistic LTL, but we do not consider that here.

Let AP be a fixed, finite set of atomic propositions. We assume a *labeled* discrete-time Markov chain $\mathcal{M} = \langle S, \mathbf{P}, L\rangle$. S, and $\mathbf{P}$ as above, with the addition of a labeling function $L\colon S \to 2^{AP}$. $L(s)$ is the set of atomic propositions $a \in AP$ that are valid in s. PCTL formulae (for the relevant subset that we consider) are of the form

$$\Phi ::= a \;\big|\; \neg\Phi \;\big|\; \Phi \wedge \Phi \;\big|\; \mathcal{P}_{\bowtie\theta}[\Phi\,\mathcal{U}\,\Phi] \;\;,$$

where $\theta \in [0,1]$ and $\bowtie \in \{\leq, \geq\}$. Additional PCTL formulae can be derived in the usual way. For example, $\bot \equiv a \wedge \neg a$ for some $a \in AP$, $\top \equiv \neg\bot$, $\Phi \vee \Psi \equiv \neg(\neg\Phi \wedge \neg\Psi)$, $\Phi \to \Psi \equiv \neg\Phi \vee \Psi$, and $\mathcal{P}_{<\theta}[\varphi] \equiv \neg\mathcal{P}_{\geq\theta}[\varphi]$.

The standard logic operators have their usual meaning. $\mathcal{P}_{\bowtie\theta}[\varphi]$ asserts that the probability measure over the set of trajectories satisfying the path formula φ is related to θ according to $\bowtie$. Path formulae are constructed using the temporal path operator $\mathcal{U}$ ("until"). The path formula $\Phi\,\mathcal{U}\,\Psi$ asserts that Ψ becomes true at some time $t \geq 0$ while Φ holds in all states prior to t. We can define mutually inductive satisfaction relations for PCTL state and path formulae as follows:

$$\begin{aligned}
&\mathcal{M}, s \models a && \text{if } a \in L(s)\\
&\mathcal{M}, s \models \neg\Phi && \text{if } \mathcal{M}, s \not\models \Phi\\
&\mathcal{M}, s \models \Phi \wedge \Psi && \text{if } (\mathcal{M}, s \models \Phi) \wedge (\mathcal{M}, s \models \Psi)\\
&\mathcal{M}, s \models \mathcal{P}_{\bowtie\theta}[\varphi] && \text{if } \mu(\{\sigma \in \mathit{Path}(s) \mid \mathcal{M}, \sigma \models \varphi\}) \bowtie \theta
\end{aligned}$$

$$\mathcal{M}, \sigma \models \Phi\,\mathcal{U}\,\Psi \;\text{ if } \exists i.\big((\mathcal{M}, \sigma[i] \models \Psi) \wedge \forall j < i.\big(\mathcal{M}, \sigma[j] \models \Phi\big)\big)$$

The fact that $\{\sigma \in \mathit{Path}(s) \mid \mathcal{M}, \sigma \models \varphi\}$ is measurable can be verified from the probability-space construction in Sect. 2.1 (cf. [1]), which makes the semantics for PCTL well-defined.

2.3 Error Control

Statistical solution methods cannot achieve the exact precision for probabilistic PCTL formulae that is required by the semantics given above. Following Younes and Simmons [25], we relax the semantics of PCTL by introducing an indifference region of half-width δ centered around any probability thresholds. The purpose is to quantify the error that we are willing to accept by using sampling and simulation in place of exact computations of probability measures.

Consider the model-checking problem $\mathcal{M}, s \models \mathcal{P}_{\bowtie\theta}[\varphi]$, and let p be the probability measure for the set of trajectories that start in s and satisfy φ. If $|p-\theta| < \delta$, then the truth value of $\mathcal{P}_{\bowtie\theta}[\varphi]$ is undetermined ("too close to call") under the relaxed semantics; otherwise, it is the same as for PCTL.

Formally, given $\delta > 0$, we define two relations: $\models^{\delta}_{\top}$ (approximate satisfaction) and $\models^{\delta}_{\bot}$ (approximate "unsatisfaction"). The definitions of $\models^{\delta}_{\top}$ and $\models^{\delta}_{\bot}$ coincide with $\models$ and $\not\models$, respectively, except for probabilistic formulae where we instead have:

$$\begin{aligned}
&\mathcal{M}, s \models^{\delta}_{\top} \mathcal{P}_{\geq\theta}[\varphi] && \text{if } \mu(\{\sigma \in \mathit{Path}(s) \mid \mathcal{M}, \sigma \models^{\delta}_{\top} \varphi\}) \geq \theta + \delta \\
&\mathcal{M}, s \models^{\delta}_{\bot} \mathcal{P}_{\geq\theta}[\varphi] && \text{if } \mu(\{\sigma \in \mathit{Path}(s) \mid \mathcal{M}, \sigma \models^{\delta}_{\bot} \varphi\}) \leq \theta - \delta \\
&\mathcal{M}, s \models^{\delta}_{\top} \mathcal{P}_{\leq\theta}[\varphi] && \text{if } \mu(\{\sigma \in \mathit{Path}(s) \mid \mathcal{M}, \sigma \models^{\delta}_{\top} \varphi\}) \leq \theta - \delta \\
&\mathcal{M}, s \models^{\delta}_{\bot} \mathcal{P}_{\leq\theta}[\varphi] && \text{if } \mu(\{\sigma \in \mathit{Path}(s) \mid \mathcal{M}, \sigma \models^{\delta}_{\bot} \varphi\}) \geq \theta + \delta
\end{aligned}$$

Let $\mathcal{M}, s \underline{\mathit{accept}}_{\mathcal{A}} \Phi$ represent the fact that Φ is accepted as true in state s of $\mathcal{M}$ by a model-checking algorithm $\mathcal{A}$, and $\mathcal{M}, s \underline{\mathit{reject}}_{\mathcal{A}} \Phi$ that Φ is rejected as false in state s of $\mathcal{M}$ by $\mathcal{A}$. The solution methods we present aim to guarantee the following error bounds:

$$\Pr[\mathcal{M}, s \underline{\mathit{reject}}_{\mathcal{A}} \Phi] \leq \alpha \qquad \text{if } \mathcal{M}, s \models^{\delta}_{\top} \Phi \tag{4}$$

$$\Pr[\mathcal{M}, s \underline{\mathit{accept}}_{\mathcal{A}} \Phi] \leq \beta \qquad \text{if } \mathcal{M}, s \models^{\delta}_{\bot} \Phi \tag{5}$$

The parameters α and β allow a user to control the probability of false negatives and false positives, respectively. For example, consider the formula $\mathcal{P}_{\geq 0.5}[\Phi \,\mathcal{U}\, \Psi]$ and let p denote the probability measure of trajectories that start in some state s and satisfy $\Phi \,\mathcal{U}\, \Psi$. Let $\delta = 0.01$. The statistical model-checking algorithms in this paper aim to guarantee that we reject the formula as false with probability at most α if $p \geq 0.5 + \delta = 0.51$, and that we accept the formula as true with probability at most β if $p \leq 0.5 - \delta = 0.49$. The three parameters α, β, and δ determine the precision of the model-checking algorithm. It is up to the user to set these to his or her satisfaction, with the understanding that higher precision will result in longer model-checking times.

3 Sampling-Based Verification of Unbounded Until

This section presents two methods for verifying probabilistic properties with unbounded until, based on statistical sampling. For the model-checking problem $\mathcal{M}, s \models \mathcal{P}_{\bowtie\theta}[\Phi \,\mathcal{U}\, \Psi]$, define the random variable $X\colon \mathit{Path}(s) \to \{0, 1\}$ as follows:

$$X(\sigma) = \begin{cases} 1 & \text{if } \mathcal{M}, \sigma \models \Phi \,\mathcal{U}\, \Psi \\ 0 & \text{if } \mathcal{M}, \sigma \not\models \Phi \,\mathcal{U}\, \Psi \end{cases} . \tag{6}$$

X represents a *Bernoulli trial* (i.e., outcomes are limited to 0 and 1). The expectation of X is

$$\mathrm{E}[X] = \mu(\{\sigma \in \mathit{Path}(s) \mid \mathcal{M}, \sigma \models \Phi \,\mathcal{U}\, \Psi\}) \ . \tag{7}$$

Hence, $\mathcal{M}, s \models \mathcal{P}_{\bowtie\theta}[\Phi\ \mathcal{U}\ \Psi]$ has a positive answer if and only if $\mathrm{E}[X] \bowtie \theta$.

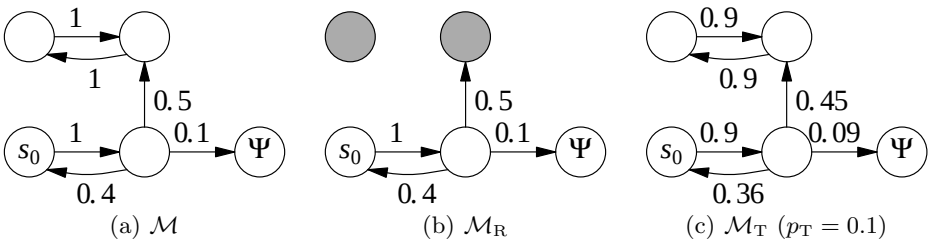
If we could sample observations of X, then we could use statistical hypothesis testing or estimation to verify $\mathcal{P}_{\bowtie\theta}[\Phi\ \mathcal{U}\ \Psi]$. To sample an observation of X, we would sample a trajectory from $Path(s)$ (e.g., using discrete-event simulation) and verify $\Phi\ \mathcal{U}\ \Psi$ over the sample trajectory. A single sample trajectory would be extended incrementally until we reach a state that satisfies either Ψ (positive observation; the path formula holds over the sample trajectory) or $\neg\Phi$ (negative observation; the path formula does not hold over the sample trajectory). The problem with this naive approach is that we are not guaranteed to ever reach a state that satisfies either Ψ or $\neg\Phi$. It works well for probabilistic properties with *time-bounded* until, as demonstrated by Younes and Simmons [26], because we can stop extending a sample trajectory if the time bound is exceeded. This ensures termination (with probability 1) provided that the model is time divergent. For unbounded until, however, there is no finite time bound to stop us from going on indefinitely, so we can no longer guarantee termination.

Consider, for example, the model in Fig. 1(a), with a single state satisfying some state formula Ψ. Assume that we want to verify $\mathcal{M}, s_0 \models \mathcal{P}_{\geq 0.15}[\top\ \mathcal{U}\ \Psi]$ (i.e., the probability of eventually reaching the state satisfying Ψ is at least 0.15 if we start in state s_0 at time 0). Note that the state satisfying Ψ is shown as absorbing—i.e., it has no outgoing transitions. This is to reflect that sample-trajectory generation ends in that state. For all other states, the outgoing transition probabilities sum to 1. Any trajectory that starts in s_0 and does not satisfy $\top\ \mathcal{U}\ \Psi$ is infinite. For this simple model, the probability measure of the set of satisfying trajectories that start in s_0 at time 0 can be computed as:

$$\mu(\{\sigma \in Path(s_0) : \mathcal{M}, \sigma \models \top\ \mathcal{U}\ \Psi\}) = 0.1 \cdot \sum_{i=0}^{\infty} 0.4^i = \frac{1}{6}\ . \tag{8}$$

Thus the stated model-checking problem has a positive answer, but a naive sampling-based approach does not work due to the positive probability ($\frac{5}{6}$) for the set of infinite trajectories.

Next, we describe two sampling-based solution methods that aim to avoid infinite sample trajectories.



(a) $\mathcal{M}$ (b) $\mathcal{M}_{\mathrm{R}}$ (c) $\mathcal{M}_{\mathrm{T}}$ ($p_{\mathrm{T}} = 0.1$)

Fig. 1. Three variations of a simple discrete-time Markov chain

3.1 Sampling-Based Method with Reachability Analysis

The first method uses reachability analysis to avoid infinite sample trajectories. For an unbounded until formula $\Phi\,\mathcal{U}\,\Psi$ to hold over a single sample trajectory σ for model $\mathcal{M}$, it is necessary (although not sufficient) that $\mathcal{M}, \sigma[i] \models \Psi$ for some $i \geq 0$. If after the generation of a finite prefix $\sigma{\uparrow}n$ it becomes evident that $\neg\Psi$ invariably holds along all possible extensions of $\sigma{\uparrow}n$, then we can determine that $\Phi\,\mathcal{U}\,\Psi$ does not hold over σ without generating a complete (possibly infinite) sample trajectory.

This condition for early termination can be expressed formally as the *non-probabilistic* CTL [6] formula $AG\,\neg\Psi$, or equivalently $\neg EF\,\Psi$. Hence, if we first verify $EF\,\Psi$ for a model $\mathcal{M}$, which amounts to reachability analysis, then we can terminate the generation of any sample trajectory entering a state of $\mathcal{M}$ that does not satisfy $EF\,\Psi$. This pre-processing step requires that we construct the full model, so it may seem counter to the appeal of sampling-based methods, which usually avoid model construction. We show, however, in Sect. 5 that this approach that combines symbolic reachability analysis and statistical sampling can work very well in practice. It scales better with the size of the model than numerical solution methods, which enables us to verify time-unbounded properties for larger models.

Let $\mathcal{M}_{\mathrm{R}}$ be the model we get by removing all outgoing transitions from all states in $\mathcal{M}$ that do not satisfy $EF\,\Psi$. We can now define the Bernoulli random variable $X_{\mathrm{R}}\colon Path_{\mathrm{R}}(s) \to \{0,1\}$ as follows:

$$X_{\mathrm{R}}(\sigma) = \begin{cases} 1 & \text{if } \mathcal{M}_{\mathrm{R}}, \sigma \models \Phi\,\mathcal{U}\,\Psi \\ 0 & \text{if } \mathcal{M}_{\mathrm{R}}, \sigma \not\models \Phi\,\mathcal{U}\,\Psi \end{cases} . \tag{9}$$

Theorem 1. *Let X be the random variable defined in (6) and let X_{R} be the random variable defined in (9). The expectation of X_{R} is the same as the expectation of X:* $\mathrm{E}[X_{\mathrm{R}}] = \mathrm{E}[X]$.

This theorem is a standard result in Markov chain theory (see, e.g., [2]), and a consequence of it is that we can use statistical hypothesis testing with observations of X_{R} instead of X to verify $\mathcal{P}_{\bowtie\theta}[\Phi\,\mathcal{U}\,\Psi]$ in $\mathcal{M}$. The benefit of using observations of X_{R} instead of X is that certain trajectories that are infinite in $\mathcal{M}$ have been made finite in $\mathcal{M}_{\mathrm{R}}$. Since X_{R} still represents a Bernoulli trial, the exact same techniques as for formulae with time-bounded until (described in detail by Younes and Simmons [26]) can be used to verify formulae with unbounded until for model $\mathcal{M}_{\mathrm{R}}$, satisfying conditions (4) and (5).

We illustrate this solution method on the discrete-time Markov chain in Fig. 1. The original model is shown in Fig. 1(a). After performing reachability analysis, we obtain the model in Fig. 1(b). The gray states have been made absorbing because they do not satisfy $EF\,\Psi$. Trajectories generated from the modified model will almost surely (with probably 1) eventually terminate.

We assume here that reachability analysis can be performed efficiently on the model $\mathcal{M}$. For Markov chains, we can ignore the actual values of transition

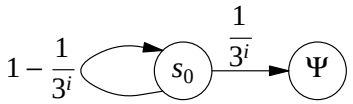


Fig. 2. A nonhomogeneous discrete-time Markov chain

probabilities and use BDD-based symbolic model checking [4]. Other models may require more advanced techniques (see, e.g., [13]). A discussion of concrete techniques for reachability analysis is beyond the scope of this paper. Clarke et al. [6] cover this topic in great depth.

For some *nonhomogeneous* Markov chains, termination may not be guaranteed (with probability 1) even after states have been made absorbing based on reachability analysis. Consider the nonhomogeneous Markov chain in Fig. 2, where i represents time. While $EF\,\Psi$ holds in s_0, the probability measure of trajectories that start in s_0 and never terminate is approximately 0.56. Hence, if we applied the reachability-based approach to this model, more than half of the sample trajectories would never terminate.

This example shows that the reachability approach is not applicable to all Markov chains. The following theorem, however, identifies a large class of models for which this approach is applicable. This theorem, too, is standard in Markov chain theory (see, e.g., [2]).

Theorem 2. *if $\mathcal{M}$ is a finite-state homogeneous discrete-time Markov chain, then the probability measure is zero for the set of infinite trajectories of $\mathcal{M}_{\mathrm{R}}$.*

3.2 Sampling-Based Method with Termination Probability

The first solution method cannot be used if reachability analysis is ineffective as exemplified by the model in Fig. 2. In the case of infinite-state systems, reachability analysis may not even be feasible.

To ensure finite trajectories without relying on reachability analysis, we can introduce a *termination probability* $p_{\mathrm{T}} < 1$, which is used as follows. Start with the stochastic discrete-event system $\mathcal{M}$. Let $\mathcal{M}_{\mathrm{T}}$ be the system we get if before each transition out of a non-absorbing state in $\mathcal{M}$ we terminate execution prematurely with probability p_{T}. Concretely, for a discrete-time Markov chain with transition probabilities $p_{ij}(n)$, we construct a new discrete-time Markov chain with transition probabilities $(1-p_{\mathrm{T}})\cdot p_{ij}(n)$. Figure 1(c) shows the result of this transformation on the model in Fig. 1(a) using termination probability $p_{\mathrm{T}} = 0.1$ (we note later on that there are limitations on the choice of p_{T}—0.1 is not an admissible choice for all models). Each transition probability in $\mathcal{M}$ is multiplied by $1-p_{\mathrm{T}}$ to obtain the corresponding transition probability in $\mathcal{M}_{\mathrm{T}}$. For example, 0.5 becomes $(1-0.1)\cdot 0.5 = 0.45$. The outgoing transition probabilities now sum to $1-p_{\mathrm{T}}$ for all non-absorbing states. In reality, of course, we never construct the new Markov chain. Instead we just take the termination probability into account when we generate sample trajectories. At each state, we terminate the generation of the trajectory prematurely with probability p_{T}.

Let $|\sigma|$ denote the number of state transitions along the trajectory σ. Define the random variable $X_{\mathrm{T}} \colon Path_{\mathrm{T}}(s) \to [0, \infty)$ as follows:

$$X_{\mathrm{T}}(\sigma) = \begin{cases} (1 - p_{\mathrm{T}})^{-|\sigma|} & \text{if } \mathcal{M}_{\mathrm{T}}, \sigma \models \Phi \mathcal{U} \Psi \\ 0 & \text{if } \mathcal{M}_{\mathrm{T}}, \sigma \not\models \Phi \mathcal{U} \Psi \end{cases} . \tag{10}$$

Trajectories that satisfy $\Phi \mathcal{U} \Psi$ are finite as they must terminate in a Ψ-satisfying state, so (10) is well-defined. Note the *negative* exponent, which means that the weight of a satisfying trajectory grows exponentially in the length of the trajectory. This construction is due to von Neumann and Ulam (see [10, 11]) as a way to compute the inverse of a matrix by the Monte Carlo method.

Theorem 3. *Let X be the random variable defined in (6) and let X_{T} be the random variable defined in (10). The expectation of X_{T} is the same as the expectation of X:* $\mathrm{E}[X_{\mathrm{T}}] = \mathrm{E}[X]$.

We can thus use observations of X_{T} instead of X to solve the model-checking problem $\mathcal{M}, s \models \mathcal{P}_{\bowtie \theta}[\Phi \ \mathcal{U} \ \Psi]$. Unlike X_{R}, which represents a Bernoulli trial, the distribution of X_{T} is unknown. Therefore we cannot use the same efficient hypothesis-testing techniques as before. However, because $\mathrm{E}[X_{\mathrm{T}}] = \mathrm{E}[X]$ we can use an estimation-based approach. If we can obtain an estimate $\tilde{p}$ of $\mathrm{E}[X_{\mathrm{T}}]$, then we can decide $\mathcal{P}_{\bowtie \theta}[\Phi \ \mathcal{U} \ \Psi]$ by comparing $\tilde{p}$ to the threshold θ. While model checking using $\mathcal{M}_{\mathrm{T}}$ requires more expensive sampling techniques than $\mathcal{M}_{\mathrm{R}}$, we can show that it is more generally applicable.

Theorem 4. *The probability measure is zero for the set of infinite trajectories of $\mathcal{M}_{\mathrm{T}}$.*

Note that Theorem 4 does not depend on any property of $\mathcal{M}$, so we are guaranteed (with probability one) finite sample trajectories for any model. For example, Theorem 4 applies to the nonhomogeneous Markov chain in Fig. 2, as well as to any infinite-state Markov process and even general discrete-event systems.

It remains to find a way to estimate $\mathrm{E}[X_{\mathrm{T}}]$. Chow and Robbins [5] provide such a method. Their method sequential procedure for computing a fixed-width confidence interval for a random variable with unknown but *finite* variance. We can use their procedure to obtain a confidence interval for $\mathrm{E}[X_{\mathrm{T}}]$ of width 2δ centered around a point estimate $\tilde{p}$ with coverage probability at least $1 - \alpha$. Let x_i be the ith observation of X_{T} and let $\bar{x}_n$ be the arithmetic mean of the first n observations. Furthermore, let $a_1, a_2, \ldots$ be a sequence of positive constants such that $\lim_{n \to \infty} a_n = \mathbf{\Phi}^{-1}(1 - \frac{\alpha}{2})$, where $\mathbf{\Phi}^{-1}$ is the inverse standard normal cumulative distribution function (in practice, we can choose a_n to be the $1 - \frac{\alpha}{2}$ quantile of the t-distribution with $n - 1$ degrees of freedom). The stopping rule for the sequential procedure is then given by [5, Eqn. 3]:

$$N = \inf \left\{ n \geq 1 : \frac{1}{n} + \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x}_n)^2 \leq \frac{\delta^2 n}{a_n^2} \right\} \tag{11}$$

We can now use $\tilde{p} = \bar{x}_N$ as a point estimate for $\mathrm{E}[X_\mathrm{T}]$, and accept $\mathcal{P}_{\bowtie\theta}[\Phi\ \mathcal{U}\ \Psi]$ as true if and only if $\tilde{p} \bowtie \theta$. As shown by Younes [25], this gives us a model-checking procedure that satisfies conditions (4) and (5) with $\beta = \alpha$.

It should be noted that the procedure of Chow and Robbins provides *asymptotic* guarantees only, meaning that the coverage probability of the confidence interval is guaranteed to be $1 - \alpha$ in the limit as δ approaches 0. In practice, the coverage probability can be somewhat less than $1 - \alpha$ for any selected δ, no matter how small, as has been shown for the normal distribution [9]. On the other hand, empirical coverage tends to be greater than $1 - \alpha$ for Bernoulli random variables. Further empirical studies are needed to determine the empirical coverage for the type of random variables we have here, which are neither normal nor Bernoulli, but this is beyond the scope of our paper.

A prerequisite for using the procedure of Chow and Robbins is that X_T has finite variance. This restriction effectively limits our choice of the termination probability p_T. For finite-state homogeneous Markov chains, we have the following theoretical result:

Theorem 5. *Let* $\mathbf{P}$ *be the transition probability matrix for* $\mathcal{M}$ *(the original model).* X_T *has finite variance iff* $p_\mathrm{T} < 1 - \rho(\mathbf{P})$*, where* $\rho(\mathbf{P})$ *is the subdominant (second-largest) eigenvalue of* $\mathbf{P}$*.*

In practice, computing the subdominant eigenvalue of a stochastic matrix is no easier than solving the model-checking problem at hand, so choosing the right value for p_T is not trivial. In Sect. 5, we apply the algorithm to *parametric models* [6] and compute $\rho(\mathbf{P})$ for small models to find a p_T that is likely to give finite variance for larger variations of the same basic model. It is important to note, however, that numerical iterative solution methods suffer from the exact same problem as discussed in the next section.

4 Related Work

To verify the formula $\mathcal{P}_{\bowtie\theta}[\Phi\ \mathcal{U}\ \Psi]$ in some state s, we can first compute the probability measure p of the set of trajectories that start in s and satisfy $\Phi\ \mathcal{U}\ \Psi$, and then compare p to θ. A numerical computation of p for any state of a Markov-chain model amounts to the solution of a set of linear equations specified as follows (cf. [1]). Let $\mathbf{P}$ be the transition probability matrix of the Markov chain and let $\mathbf{P}'$ equal $\mathbf{P}$, with the exception that states satisfying $\neg\Phi \vee \Psi$ have been made absorbing. Furthermore, let $\boldsymbol{v}$ be a binary column vector with a 1 in each row corresponding to a state that satisfies Ψ. Then $\boldsymbol{p}$ is the solution to

$$\boldsymbol{p} = \mathbf{P}' \cdot \boldsymbol{p} + \boldsymbol{v}\ . \tag{12}$$

The equation system in (12) can be written as $(\mathbf{I} - \mathbf{P}') \cdot \boldsymbol{p} = \boldsymbol{v}$ and solved using Gaussian elimination, which is guaranteed to be polynomial in the size of the state space. This approach is memory intensive, however, and also suffers from numerical instability. For these reasons, Iterative solution methods, such

as Jacobi and Gauss-Seidel [22], are typically preferred. The leading tool for probabilistic model checking, PRISM [17], relies on iterative methods to verify properties with unbounded until. Each iteration involves a matrix–vector multiplication, which in the worst case is $O(n^2)$, but often $O(n)$ (for sparse models), where n is the size of the state space. This dependence on the size of the state space make numerical solution methods impractical for very large models, in which case sampling-based solution become an attractive alternative.

The number of iterations (k) required to achieve some numerical precision ϵ is related to the subdominant eigenvalue (ρ) of $\mathbf{P}'$ as follows [22, p. 156]: $k = \frac{\log \epsilon}{\log \rho}$. Since computing eigenvalues is no easier than solving the model-checking problem, heuristics must be employed to bound the number of iterations, but this means that no formal correctness guarantees can be made. This is similar to the situation for the second sampling-based method we described. The reachability-based sampling approach, in contrast, does not suffer from this weakness as the precision of the result is independent of any property of the model.

John von Neumann and Stanislaw Ulam, as early as the 1940s, devised a Monte Carlo method for solving systems of linear equations of the type in (12). It is this algorithm, first published by Forsythe and Leibler [10], that we use in the second of our solution methods. It should come as no surprise that both the iterative numerical solution methods, and the Monte Carlo approach that uses a termination probability, have a dependency on the subdominant eigenvalue to provide some prescribed precision. The method of von Neumann and Ulam is essentially a Monte Carlo version of a numerical iterative algorithm.

Sampling-based solution methods for time-unbounded formulae have received some attention recently [21, 18, 3, 7], but these authors appear unaware of the method devised by von Neumann and Ulam.

Sen et al. [21] propose a solution method that on the surface looks similar to our second approach (the one based on the method by von Neumann and Ulam). They use a termination probability p_{T} in the same way as we to ensure terminating sample trajectories. Instead of using weighted sampling with the random variable X_{T}, however, they use the following Bernoulli random variable:

$$Y_{\mathrm{T}}(\sigma) = \begin{cases} 1 & \text{if } \mathcal{M}_{\mathrm{T}}, \sigma \models \Phi \,\mathcal{U}\, \Psi \\ 0 & \text{if } \mathcal{M}_{\mathrm{T}}, \sigma \not\models \Phi \,\mathcal{U}\, \Psi \end{cases} . \tag{13}$$

The problem with Y_{T} is that its expectation does not match that of the random variable X. Sen et al. recognize this problem and provide a bound on the expectation $\mathrm{E}[X]$ expressed in terms of $\mathrm{E}[Y_{\mathrm{T}}]$. The bound depends on the size of the model, however, and is too loose to be useful in practice.

In other work [18, 27, 3], it is proposed to use the results from verifying time-bounded properties to obtain a probability estimate for unbounded properties. These methods essentially boil down to estimating the expected value of the following random variable:

$$Z_k(\sigma) = \begin{cases} 1 & \text{if } \mathcal{M}, \sigma{\uparrow}k \models \Phi \,\mathcal{U}\, \Psi \\ 0 & \text{if } \mathcal{M}, \sigma{\uparrow}k \not\models \Phi \,\mathcal{U}\, \Psi \end{cases} . \tag{14}$$

As with Y_T, Z_k does not have the same expectation of X, although we do have $\lim_{k\to\infty} Z_k = X$. The expected value for Z_k is estimated for a series of increasing values for k until some convergence criterion is met. Lassaigne and Peyronnet [18] relate the choice of k to the subdominant eigenvalue, but as with the other theoretical results mentioned earlier that involve the subdominant eigenvalue, this result does not help to choose k in practice. Ensuring a certain accuracy becomes hard because each successive iteration with a different value for k is subject to error, and different ad-hoc termination criteria are proposed by the various authors. In the solution methods presented in this paper, we avoid this iterative estimation approach by always setting up experiments that preserve the expected value of the quantity of interest.

Rabih et al. [7] present a very different simulation-based approach to verifying unbounded until properties. They develop an algorithm based on perfect simulation. The approach is interesting, but impractical unless the model is monotone. The authors do not discuss how to determine monotonicity for a model, or whether the widely-used PRISM benchmarks satisfy this property, so it is hard to assess the general applicability of their method.

The termination-probability approach has been used for rare-event simulation (see, e.g., [19]). Improvements to the basic algorithm from the simulation community would be valuable for model checking as well.

5 Empirical Evaluation

We use two different continuous-time Markov-chain models as the basis for our empirical evaluation, with rather different characteristics. Note that model checking of unbounded until for continuous-time Markov-chains reduces to model checking for the embedded discrete-time Markov chain [1]. Thus, a continuous-time Markov chain presents no additional challenge for the algorithms described in Sect. 3.

5.1 Modified Polling System

The first model is a variation of an n-station symmetric polling system, described by Ibe and Trivedi [16]. In the original polling-system model, each station has a single-message buffer and the stations are attended to by a single server in cyclic order. When attending to a station, the server checks for a message in the buffer of the station by polling the station, and goes on to serve the station if there is a message. The polling and service times are exponentially distributed with rates $\gamma = 200$ and $\mu = 1$, respectively. Furthermore, each station has an inter-arrival time for messages that is exponentially distributed with rate $\lambda = 1/n$.

Here, we consider a modified version of the polling-system model, where polling stations can fail and stop accepting messages. The failure rate of a station is $\kappa = \lambda/3$. After a station has failed, it can never be served again. This way, infinite sample trajectories become a possibility for the property we consider.

We verify the following property: the probability is at least 0.4 that station 1 is served before station 2. Let $s \in \{1, \ldots, n\}$ be the station currently attended to by the server, and let $a \in \{0, 1\}$ represent the activity of the server (0 for polling and 1 for serving). The property can then be expressed as the formula $\mathcal{P}_{\geq 0.4}[\neg(s{=}2 \wedge a{=}1)\, \mathcal{U}\ s{=}1 \wedge a{=}1]$. We verify this formula in the state where the server is about to attend to station 1 (s=1 and a=0) and all buffers are empty. If both station 1 and station 2 fail before they are served, $\neg(s{=}2 \wedge a{=}1)$ will remain true and $s{=}1 \wedge a{=}1$ false indefinitely, and a sample trajectory for the given model-checking problem may never terminate.

We have implemented the two sampling-based algorithms described in this paper in YMER [24]. We compare the sampling-based approaches with numerical iterative algorithms implemented in PRISM [17]. For the experiments, we used $\alpha = \beta = 0.01$ and $\delta = 0.005$ for the sampling-based algorithms, and $\epsilon = 0.005$ (absolute error) for the numerical algorithms. These choices are of course somewhat arbitrary. Younes and Simmons [26] provide a thorough investigation of how these parameter choices affect performance for sampling-based algorithms. For example, using $\alpha = \beta = 10^{-8}$ increases verification time by about a factor 10. For the reachability approach, we tried both Wald's SPRT [23] (sequential hypothesis testing) and estimation based on the Hoeffding bound [15] that gives a fixed sample size of $N = \lceil \frac{1}{2\delta^2} \log \frac{2}{\alpha} \rceil$ (105,967 for our choice of parameters). For the approach based on termination probability we used Chow and Robbins' sequential estimation procedure mentioned earlier and $p_{\mathrm{T}} = 10^{-4}$ (this choice was made after computing the subdominant eigenvalue for the transition matrices of models with up to 12 stations). Finally, for the numerical approaches, we used the hybrid engine in PRISM, trying both Jacobi and (backwards) Gauss-Seidel.

Figure 3 plots the verification time as a function of state-space size for the modified polling system, when verifying the stated property using different algorithms. For the sampling-based approaches, the graph shows the average time over 20 trials. The state-space size for a model with n stations is $4n \cdot 3^{n-1}$. A model with 24 stations, for example, has close to 10^{13} states.

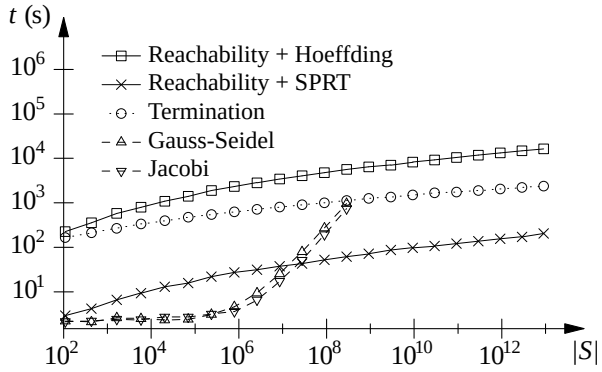


Fig. 3. Verification time as a function of state-space size for modified polling system

The sampling-based approaches scale well as the state-space size grows. Reachability combined with the SPRT does best, partly because the underlying probability is close to 0.5, while the bound in the formula is 0.4, so a decision can be reached with an average sample size of about 1,200. The estimation-based approaches require much larger sample sizes. The termination-based approach beats the reachability-based approach when the latter is used with a sample size derived from the Hoeffding bound.

The numerical algorithms are much faster for small state spaces, but performance deteriorates quickly for larger state spaces. They also use much more memory than the sampling-based approaches. For a model with 16 stations (almost 1 billion states), PRISM caused serious thrashing on our test computer equipped with 8 GB of RAM. The reachability-based approach has a much more modest memory growth, and nothing suggests that it could not handle models much larger than what we have tested here. The termination-based approach uses the least amount of memory, as it does not require reachability analysis, and does not show any noticeable growth as models get bigger.

5.2 Tandem Queuing Network

The second model is a tandem queuing network due to Hermanns et al. [14]. The network consists of two serially connected queues, each with capacity n. Messages arrive at the first queue, get routed to the second queue, and eventually leave the system from the second queue. The inter-arrival time for messages at the first queue is exponentially distributed with rate $\lambda = 4n$. The processing time at the second queue is exponentially distributed with rate $\kappa = 4$. The size of the state space for this model is $O(n^2)$.

We verify that the probability is at most 0.03 that the second queue becomes full before the first queue: $\mathcal{P}_{\leq 0.03}[\neg \mathit{full}_1 \,\mathcal{U}\, \mathit{full}_2]$. We use the same experimental setup as for the first model, except for the choice of p_{T}. Instead of a fixed value, we use $p_{\mathrm{T}} = \frac{1}{n+2}$ for a model with queues of size n. We do so because the subdominant eigenvalue for this model more quickly approaches 1 as n grows. Figure 4 plots the verification time as a function of state-space size for the tandem queuing network, using the same algorithms as before. Again, the results for sampling-based approaches are averages over 20 trials.

For this model, the sampling-based algorithm that uses termination probability comes out on top. The reason is that reachability analysis is much more expensive for this model. The difference between Hoeffding and SPRT gets smaller for larger state spaces, as the time needed to perform reachability analysis starts to dominate the difference in sample size (446 for SPRT; 105,967 for Hoeffding).

6 Discussion

We have presented two sampling-based algorithms for probabilistic model checking of time-unbounded properties. Both solution methods are based on unbiased

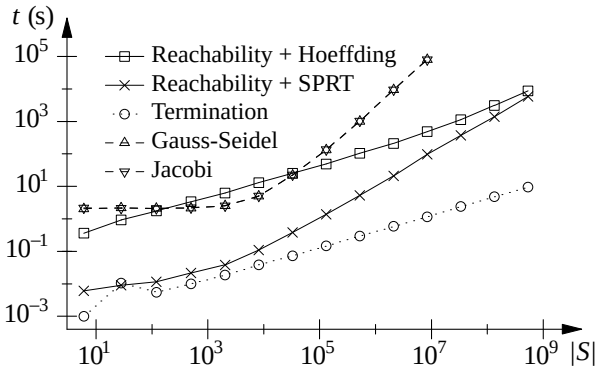


Fig. 4. Verification time as a function of state-space size for tandem queuing network

estimators. This avoids the convergence issues that haunt existing sampling-based algorithms, which all use biased estimators. The first method, especially, is valuable as it provides correctness guarantees that are independent of any model parameters. The second method has the same weakness as popular iterative numerical solution methods, in that accuracy cannot be guaranteed fully without knowledge of the subdominant eigenvalue. Still, it allows us to analyze models far beyond the reach of methods that require model construction (as is the case for the first method, as well as numerical methods). Future work could focus on extending the theoretical results of this paper. In particular, conditions under which X_{T} has finite variance should be established for a more general class of systems. Techniques from the simulation community should be incorporated to reduce the variance of X_{T}, and the empirical coverage probability for sequential estimation should be established.

Acknowledgments. Edmund M. Clarke and Paolo Zuliani were supported by the National Science Foundation under Grant No. 0926181.

References

1. Baier, C., Haverkort, B. R., Hermanns, H., and Katoen, J.-P. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.
2. Baier, C. and Katoen, J.-P. *Principles of Model Checking*. The MIT Press, 2008.
3. Basu, S., Ghosh, A. P., and He, R. Approximate model checking of PCTL involving unbounded path properties. In *Proc. 11th ICFEM*, LNCS. Springer, 2009.
4. Burch, J. R., Clarke, E. M., McMillan, K. L., Dill, D. L., and Hwang, L. J. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98(2):142–170, 1992.
5. Chow, Y. S. and Robbins, H. On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *Annals of Mathematical Statistics*, 36(2):457–462, 1965.

6. Clarke, E. M., Grumberg, O., and Peled, D. A. *Model Checking*. The MIT Press, 1999.
7. El Rabih, D. and Pekergin, N. Statistical model checking using perfect simulation. In *Proc. 7th ATVA*, volume 5799 of *LNCS*, pages 120–134. Springer, 2009.
8. Etessami, K. and Rajamani, S. K., editors. *Proc. 17th CAV*, volume 3576 of *LNCS*. Springer, 2005.
9. Fishman, G. S. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer, 1996.
10. Forsythe, G. E. and Leibler, R. A. Matrix inversion by a Monte Carlo method. *Mathematical Tables and Other Aids to Computation*, 4(31):127–129, 1950.
11. Hammersley, J. M. and Handscomb, D. C. Solution of linear operator equations. In *Monte Carlo Methods*, chapter 7, pages 85–96. Methuen & Co, 1964.
12. Hansson, H. and Jonsson, B. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
13. Henzinger, T. A., Nicollin, X., Sifakis, J., and Yovine, S. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.
14. Hermanns, H., Meyer-Kayser, J., and Siegle, M. Multi terminal binary decision diagrams to represent and analyse continuous time Markov chains. In *Proc. 3rd International Workshop on the Numerical Solution of Markov Chains*, pages 188–207. Prensas Universitarias de Zaragoza, 1999.
15. Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
16. Ibe, O. C. and Trivedi, K. S. Stochastic Petri net models of polling systems. *IEEE Journal on Selected Areas in Communications*, 8(9):1649–1657, 1990.
17. Kwiatkowska, M., Norman, G., and Parker, D. Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer*, 6(2):128–142, 2004.
18. Lassaigne, R. and Peyronnet, S. Probabilistic verification and approximation. *Annals of Pure and Applied Logic*, 152(1–3):122–131, 2008.
19. L'Ecuyer, P., Demers, V., and Tuffin, B. Splitting for rare-event simulation. In *Proc. 2006 Winter Simulation Conference*. IEEE, 137-148.
20. Monniaux, D. An abstract monte-carlo method for the analysis of probabilistic programs. In *Proc. 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 93–101. Association for Computing Machinery, 2001.
21. Sen, K., Viswanathan, M., and Agha, G. On statistical model checking of stochastic systems. In *Proc. 17th CAV*, volume 3576 of *LNCS*, pages 266–280. Springer, 2005.
22. Stewart, W. J. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.
23. Wald, A. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.
24. Younes, H. L. S. Ymer: A statistical model checker. In *Proc. 17th CAV*, volume 3576 of *LNCS*, pages 429–433. Springer, 2005.
25. Younes, H. L. S. Error control for probabilistic model checking. In *Proc. 7th International Conference on Verification, Model Checking, and Abstract Interpretation*, volume 3855 of *LNCS*, pages 142–156. Springer, 2006.
26. Younes, H. L. S. and Simmons, R. G. Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9):1368–1409, 2006.
27. Zapreev, I. S. *Model checking Markov chains: Techniques and tools*. PhD thesis, University of Twente, 2008.