Probabilistic Verification for "Black-Box" Systems[*]

Håkan L.S. Younes

Computer Science Department, Carnegie Mellon University,
Pittsburgh, PA 15213, USA

Abstract. We explore the concept of a "black-box" stochastic system, and propose an algorithm for verifying probabilistic properties of such systems based on very weak assumptions regarding system dynamics. Properties are expressed as formulae in a probabilistic temporal logic. Our presentation is a generalization of and an improvement over recent work by Sen et al. on probabilistic verification for "black-box" systems.

1 Introduction

Stochastic processes are used to model phenomena in nature that involve an element of chance (the throwing of a die) or are too complex to fully capture in a deterministic fashion (the duration of a call in a telephone system). Certain classes of stochastic processes have been studied extensively in the performance evaluation and model checking communities. Numerous temporal logics, such as TCTL [1], PCTL [8], and CSL [2, 3], exist for expressing interesting properties of various types of stochastic processes. Model checking algorithms have been developed for verifying properties of discrete-time Markov chains [8], continuous-time Markov chains [3, 11], semi-Markov processes [10], generalized semi-Markov processes [1], and stochastic discrete event systems in general [15].

Given a stochastic process, we want to know if certain probabilistic properties hold. For instance, we may ask whether the probability of exhausting bandwidth over a communication link is below 0.01. We can also introduce deadlines, for example that a message arrives at its destination within 15 seconds with probability at least 0.8. Properties of this type can be verified using either numerical methods or statistical sampling techniques, as discussed by Younes et al. [14]. Numerical methods provide highly accurate results, but rely on strong assumptions regarding the dynamics of the systems they are used to analyze. Statistical techniques require only that the dynamics of a system can be simulated. They can thus be used for a larger class of stochastic processes, but results are only probabilistic and attaining high accuracy can to be costly.

For some systems, it may not even be feasible to assume that we can simulate their behavior. Sen et al. [12] consider the verification problem for such "black-box" systems. Here, "black-box" means that the system cannot be controlled to

[*] Supported in part by the US Army Research Office (ARO), under contract no. DAAD190110485, and the Royal Swedish Academy of Engineering Sciences (IVA).

K. Etessami and S.K. Rajamani (Eds.): CAV 2005, LNCS 3576, pp. 253–265, 2005.

generate execution traces, or trajectories, on demand starting from arbitrary states. This is a reasonable assumption, for instance, for a system that has already been deployed and for which we are given only a set of trajectories generated during actual execution of the system. We are then asked to verify a probabilistic property of the system based on the information provided to us as a fixed set of trajectories. Statistical solution techniques are certainly required to solve this problem. The statistical method described by Younes and Simmons [15] (see also [13–Chap. 5]) cannot be used to verify "black-box" systems, however, because it depends on the ability to generate trajectories on demand.

Sen et al. [12] present an alternative solution method for verification of "black-box" systems based on statistical hypothesis testing with fixed sample sizes. In this paper, we improve upon their algorithm by making sure to always accept the most likely hypothesis, and we correct their procedure for verifying nested probabilistic properties. Differences between the two approaches are discussed in detail in Sect. 5.

We focus our attention on systems with piecewise constant trajectories. The class of stochastic discrete event systems, defined in Sect. 2, satisfies this constraint. Sect. 3 introduces the *unified temporal stochastic logic* (UTSL), which can be used to express probabilistic and temporal properties of stochastic discrete event systems. UTSL represents a unification of Hansson and Jonsson's [8] PCTL, which has a semantics defined for discrete-time Markov chains, and Baier et al.'s [3] version of CSL (excluding the steady-state operator), which has a semantics defined for continuous-time Markov chains.

Sect. 4 presents an algorithm for the verification of "black-box" systems. Our algorithm, like that of Sen et al. [12], provides no *a priori* guarantees regarding accuracy. Instead, the algorithm computes a p-value for the result, which is a measure of confidence. The algorithm is essentially finding the most likely answer to a model checking problem given a fixed set of trajectories. This is the best we can do, provided that we cannot generate trajectories for the system as we see fit and are restricted to using a predetermined set of trajectories.

The algorithm presented in this paper is complementary to the statistical model checking algorithm presented by Younes and Simmons [15], and is useful under different assumptions. If we cannot generate trajectories for a system on demand, then the algorithm presented here still allows us to reach conclusions regarding the behavior of the system. If, however, we can simulate the dynamics of the system, then we are better off with the approach of Younes and Simmons as it gives us full control over the probability of obtaining an incorrect result.

2 Stochastic Discrete Event Systems

A *stochastic process* is any process that evolves over time, and whose evolution one can follow and predict in terms of probability [4]. At any point in time, a stochastic process is said to occupy some state. If we attempt to observe the state of a stochastic process at a specific time, the outcome of such an observation is governed by some probability law. Mathematically, a stochastic process is defined as a family of random variables.

Definition 1 (Stochastic Process). *Let S and T be two sets. A* stochastic process *is a family of random variables $\mathcal{X} = \{X_t \mid t \in T\}$, with each random variable X_t having range S.*

The index set T in Definition 1 represents time and is typically the set of non-negative integers, $\mathbb{Z}^*$, for discrete-time stochastic processes and the set of non-negative real numbers, $[0, \infty)$, for continuous-time stochastic processes. The set S represents the states that the stochastic process can occupy, and this can be an infinite, or even uncountable, set.

The definition of a stochastic process as a family of random variables is quite general and includes systems with both continuous and discrete dynamics. We will focus our attention on a limited, but important, class of stochastic processes: *stochastic discrete event systems*. This class includes any stochastic process that can be thought of as occupying a single state for a duration of time before an *event* causes an instantaneous state transition to occur. The canonical example of such a process is a queuing system, with the state being the number of items currently in the queue. The state changes at the occurrence of an event representing the arrival or departure of an item.

2.1 Trajectories

A random variable $X_t \in \mathcal{X}$ represents the chance experiment of observing the stochastic process $\mathcal{X}$ at time t. If we record our observations at consecutive time points for all $t \in T$, then we have a *trajectory*, or *sample path*, for $\mathcal{X}$. Our work in probabilistic verification is centered around the verification of temporal logic formulae over trajectories for stochastic discrete event systems. The terminology and notation introduced here is used extensively in later sections.

Definition 2 (Trajectory). *A* trajectory *for a stochastic process $\mathcal{X}$ is any sequence of observations $\{x_t \in S \mid t \in T\}$ of the random variables $X_t \in \mathcal{X}$.*

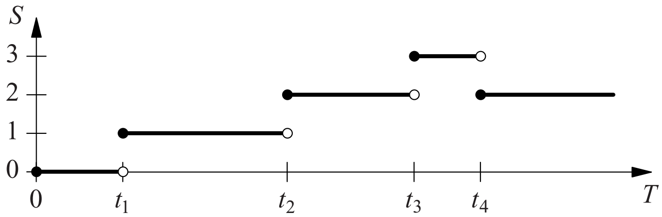

Fig. 1. A trajectory for a simple queuing system with arrival events occurring at t_1, t_2 and t_3 and a departure event occurring at t_4.

The trajectory of a stochastic discrete event system is *piecewise constant* and can therefore be represented as a sequence $\sigma = \{\langle s_0, t_0\rangle, \langle s_1, t_1\rangle, \ldots\}$, with $s_i \in S$ and $t_i \in T \setminus \{0\}$. Zero is excluded to ensure that only a single state can be occupied at any point in time. Fig. 1 plots part of a trajectory for a simple queuing system. Let

$$T_i = \begin{cases} 0 & \text{if } i = 0 \\ \sum_{j=0}^{i-1} t_j & \text{if } i > 0 \end{cases} , \tag{1}$$

i.e. T_i is the time at which state s_i is entered and t_i is the duration of time for which the process remains in s_i before an event triggers a transition to state s_{i+1}. A trajectory σ is then a sequence of observations of $\mathcal{X}$ with $x_t = s_i$ for $T_i \leq t < T_i + t_i$. According to this definition, trajectories of stochastic discrete event systems are *right-continuous.* A finite trajectory is a sequence $\sigma = \{\langle s_0, t_0\rangle, \ldots, \langle s_n, \infty\rangle\}$ where s_n is an *absorbing* state, meaning that no events can occur in s_n and that $x_t = s_n$ for all $t \geq T_n$.

2.2 Measurable Stochastic Discrete Event Systems

Of utmost importance to probabilistic verification is the definition of a *probability measure* over sets of trajectories for a system. The set of trajectories must be *measurable.* Formally, a *measurable space* is a set Ω with a σ-algebra $\mathcal{F}_\Omega$ of subsets of Ω [7]. A *probability space* is a measurable space $\langle \Omega, \mathcal{F}_\Omega\rangle$ and a probability measure μ.

For stochastic discrete event systems, the elements of the σ-algebra are sets of trajectories with common *prefix.* A prefix of $\sigma = \{\langle s_0, t_0\rangle, \langle s_1, t_1\rangle, \ldots\}$ is a sequence $\sigma_{\leq \tau} = \{\langle s'_0, t'_0\rangle, \ldots, \langle s'_k, t'_k\rangle\}$, with $s'_i = s_i$ for all $i \leq k$, $\sum_{i=0}^{k} t'_i = \tau$, $t'_i = t_i$ for all $i < k$, and $t'_k < t_k$. Let $Path(\sigma_{\leq \tau})$ denote the set of trajectories with common prefix $\sigma_{\leq \tau}$. This set must be measurable, and we assume that a probability measure μ over sets of trajectories with common prefix exists. This requirement is not a problem in practice. In general, a stochastic discrete event system is measurable if the sets S and T are measurable.

The precise definition of μ depends on the specific probability structure of the stochastic process being studied. A stochastic process is a Markov chain if $\mu(Path(\{\langle s_0, t_0\rangle, \ldots, \langle s_k, t_k\rangle\})) = \mu(Path(\{\langle s_k, 0\rangle\}))$ for all trajectory prefixes $\{\langle s_0, t_0\rangle, \ldots, \langle s_k, t_k\rangle\}$. We define a “black-box” probabilistic system in terms of what we know (or rather, do not know) regarding the probability measure μ.

Definition 3 (“Black-Box” Probabilistic System). *A “black-box” probabilistic system is a stochastic discrete event system for which the probability measure μ over sets of trajectories with common prefix is not fully specified.*

3 UTSL: The Unified Temporal Stochastic Logic

A stochastic discrete event system is a triple $\langle S, T, \mu\rangle$. We assume a factored representation of S, with a set of state variables SV and a value assignment function $V(s, x)$ providing the value of $x \in SV$ in state s. The domain of x is the set $D_x = \bigcup_{s \in S} V(s, x)$ of possible values that x can take on. We define the syntax of UTSL for a factored stochastic discrete event system $\mathcal{M} = \langle S, T, \mu, SV, V\rangle$ as

$$\Phi ::= x \sim v \bigm| \neg\Phi \bigm| \Phi \wedge \Phi \bigm| \mathcal{P}_{\bowtie \theta}[X^I\ \Phi] \bigm| \mathcal{P}_{\bowtie \theta}[\Phi\ \mathcal{U}^I\ \Phi] ,$$

where $x \in SV$, $v \in D_x$, $\sim \in \{\leq, =, \geq\}$, $\theta \in [0,1]$, $\bowtie \in \{\leq, \geq\}$, and $I \subset T$. Additional UTSL formulae can be derived in the usual way. For example, $\bot \equiv (x = v) \wedge \neg(x = v)$ for some $x \in SV$ and $v \in D_x$, $\top \equiv \neg\bot$, $\Phi \vee \Psi \equiv \neg(\neg\Phi \wedge \neg\Psi)$, $\Phi \rightarrow \Psi \equiv \neg\Phi \vee \Psi$, $\mathcal{P}_{\bowtie\theta}[\Phi\, \mathcal{U}\, \Psi] \equiv \mathcal{P}_{\bowtie\theta}[\Phi\, \mathcal{U}^T\, \Psi]$, 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 operators X^I ("next") and $\mathcal{U}^I$ ("until"). The path formula $X^I\ \Phi$ asserts that the next state transition occurs $t \in I$ time units into the future and that Φ holds in the next state, while $\Phi\, \mathcal{U}^I\ \Psi$ asserts that Ψ becomes true $t \in I$ time units into the future while Φ holds continuously prior to t.

The validity of a UTSL formula, relative to a factored stochastic discrete event system $\mathcal{M}$, is defined in terms of a satisfaction relation $\models_{\mathcal{M}}$:

$$\begin{aligned}
\{\langle s_0, t_0\rangle, \ldots, \langle s_k, t_k\rangle\} \models_{\mathcal{M}} x \sim v \quad & \text{iff } V(s_k, x) \sim v \\
\sigma_{\leq\tau} \models_{\mathcal{M}} \neg\Phi \quad & \text{iff } \sigma_{\leq\tau} \not\models_{\mathcal{M}} \Phi \\
\sigma_{\leq\tau} \models_{\mathcal{M}} \Phi \wedge \Psi \quad & \text{iff } (\sigma_{\leq\tau} \models_{\mathcal{M}} \Phi) \wedge (\sigma_{\leq\tau} \models_{\mathcal{M}} \Psi) \\
\sigma_{\leq\tau} \models_{\mathcal{M}} \mathcal{P}_{\bowtie\theta}[\varphi] \quad & \text{iff } \mu(\{\sigma \in \mathit{Path}(\sigma_{\leq\tau}) \mid \sigma, \tau \models_{\mathcal{M}} \varphi\}) \bowtie \theta
\end{aligned}$$

$$\begin{aligned}
\sigma, \tau \models_{\mathcal{M}} X^I\ \Phi \quad & \text{iff } \exists k \in \mathbb{N}.\big((T_{k-1} \leq \tau) \wedge (\tau < T_k) \wedge (T_k - \tau \in I) \wedge (\sigma_{\leq T_k} \models_{\mathcal{M}} \Phi)\big) \\
\sigma, \tau \models_{\mathcal{M}} \Phi\, \mathcal{U}^I\ \Psi \quad & \text{iff } \exists t \in I.\big((\sigma_{\leq\tau+t} \models_{\mathcal{M}} \Psi) \wedge \forall t' \in T.\big((t' < t) \rightarrow (\sigma_{\leq\tau+t'} \models_{\mathcal{M}} \Phi)\big)\big)
\end{aligned}$$

The semantics of $\Phi\, \mathcal{U}^I\ \Psi$ requires that Φ holds continuously, i.e. at all time points, along a trajectory until Ψ is satisfied. This is consistent with the semantics of time-bounded until for TCTL [1]. Depending on the probability measure μ, Φ may hold immediately at the entry of a state s and also immediately after a transition from s to s', but still not hold continuously while the system remains in s. Conversely, Ψ may hold at some point in time while the system remains in s, and not hold immediately upon entry to s nor immediately after a transition from s to s'. It is therefore not sufficient, in general, to verify Φ and Ψ at discrete points along a trajectory. It is sufficient to do so, however, for Markov chains. Our semantics for UTSL interpreted over general stochastic discrete event systems therefore coincides with the semantics for PCTL interpreted over discrete-time Markov chains [8] and CSL interpreted over continuous-time Markov chains [3], provided we choose the time domain T appropriately.

A UTSL model checking problem is a triple $\langle\mathcal{M}, s, \Phi\rangle$, with the problem being to verify whether Φ holds for $\mathcal{M}$ if execution starts in state s, i.e. $\{\langle s, 0\rangle\} \models_{\mathcal{M}} \Phi$. We use $s \models \Phi$ as a short form for the latter, leaving out $\mathcal{M}$ when it is clear from the context which system is involved in the model checking problem.

4 Statistical Verification Algorithm

A stochastic discrete event system $\mathcal{M}$ is a "black-box" system if we lack an exact definition of the probability measure μ over sets of trajectories of $\mathcal{M}$ (Definition 3) and we cannot sample trajectories according to μ. Thus, to solve a verification problem $s \models \Phi$ for $\mathcal{M}$, we must rely on an external source to provide a sample set of n trajectories for $\mathcal{M}$ that is representative of the probability measure μ. We further assume that we are provided only with *truncated* trajectories, because infinite trajectories would require infinite memory to store.

We use statistical hypothesis testing to verify properties of a "black-box" system given a sample of n truncated trajectories. Since we rely on statistical techniques, we will typically not know with certainty if the result we produce is correct. The method we present for verification of "black-box" systems computes a p-value for a verification result, which is a value in the interval $[0, 1]$ with values closer to 0 representing higher confidence in the result [9–pp. 255–256].

4.1 Verification Without Nested Probabilistic Operators

Given a state s, verification of a UTSL formula $x \sim v$ is trivial. We can simply read the value assigned to x in s and compare it to v. We consider the remaining three cases in more detail, starting with the probabilistic operator $\mathcal{P}_{\bowtie\theta}[\cdot]$. The objective is to produce a Boolean result annotated with a p-value.

Probabilistic Operator. Consider the problem of verifying the UTSL formula $\mathcal{P}_{\bowtie\theta}[\varphi]$ in state s of a stochastic discrete event system $\mathcal{M}$. Let X_i be a random variable representing the verification of the path formula φ over a trajectory for $\mathcal{M}$ drawn according to the probability measure $\mu(\mathit{Path}(\{\langle s, 0\rangle\}))$. If we choose $X_i = 1$ to represent the fact that φ holds over a random trajectory, and $X_i = 0$ to represent the opposite fact, then X_i is a *Bernoulli variate* with parameter $p = \mu(\{\sigma \in \mathit{Path}(\{\langle s, 0\rangle\}) \mid \sigma, 0 \models \varphi\})$, i.e. $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$. To verify $\mathcal{P}_{\bowtie\theta}[\varphi]$, we can make observations of X_i and use statistical hypothesis testing to determine if $p \bowtie \theta$ is likely to hold. An observation of X_i, denoted x_i, is the verification of φ over a specific trajectory σ_i. If σ_i satisfies the path formula φ, then $x_i = 1$, otherwise $x_i = 0$.

In our case, we are given n truncated trajectories for a "black-box" system that we can use to generate observations of X_i. Each observation is obtained by verifying the path formula φ over one of the truncated trajectories. This is straightforward given a truncated trajectory $\{\langle s_0, t_0\rangle, \ldots, \langle s_{k-1}, t_{k-1}\rangle, s_k\}$, provided that φ does not contain any probabilistic operators. For $\varphi = X^I\ \Phi$, we just check if $t_0 \in I$ and $s_1 \models \Phi$. For $\varphi = \Phi\ \mathcal{U}^I\ \Psi$, we traverse the trajectory until we find a state s_i such that one of the following conditions holds, with T_i defined as in (1) to be the time at which state s_i is entered:

1. $(s_i \models \neg\Phi) \wedge ((T_i \notin I) \vee (s_i \models \neg\Psi))$
2. $(T_i \in I) \wedge (s_i \models \Psi)$
3. $((T_i, T_{i+1}) \cap I \neq \emptyset) \wedge (s_i \models \Phi) \wedge (s_i \models \Psi)$

In the first case, $\Phi\ \mathcal{U}^I\ \Psi$ does not hold over the trajectory, while in the last two cases the time-bounded until formula does hold. Note that we may not always be able to determine the value of φ over all trajectories because the trajectories that are provided to us are assumed to be truncated.

We consider the case $\mathcal{P}_{\geq\theta}[\varphi]$ in detail, noting that $\mathcal{P}_{\leq\theta}[\varphi]$ can be handled in the same way simply by reversing the value of each observation. We want to test the hypothesis $H_0 : p \geq \theta$ against the alternative hypothesis $H_1 : p < \theta$ by using the n observations $x_1, \ldots, x_n$ of $X_1, \ldots, X_n$. To do so, we specify a constant c. If $\sum_{i=1}^{n} x_i$ is greater than c, then hypothesis H_0 is accepted, i.e. $\mathcal{P}_{\geq\theta}[\varphi]$ is determined to hold. Otherwise, if the given sum is at most c, then hypothesis H_1 is accepted, meaning that $\mathcal{P}_{\geq\theta}[\varphi]$ is determined not to hold. The constant c should be chosen so that it becomes roughly equally likely to accept H_0 as H_1 if p equals θ. The pair $\langle n, c\rangle$ is referred to as a *single sampling plan* [6, 5].

The probability distribution of a sum of n Bernoulli variates with parameter p is a binomial distribution with cumulative distribution function $F(c; n, p) = \sum_{i=0}^{c} \binom{n}{i} p^i (1-p)^{n-i}$. Using a single sampling plan $\langle n, c\rangle$, we accept hypothesis H_1 with probability $F(c; n, p)$ and hypothesis H_0 with probability $1 - F(c; n, p)$. Ideally, we should choose c such that $F(c; n, \theta) = 0.5$, but it is not always possible to attain equality because the binomial distribution is a discrete distribution. The best we can do is to choose c such that $|F(c; n, \theta) - 0.5|$ is minimized.

We now have a way to decide whether to accept or reject the hypothesis that $\mathcal{P}_{\geq\theta}[\varphi]$ holds, but we also want to report a p-value reflecting the confidence in our decision. The p-value is defined as the probability of the sum of observations being at least as extreme as the one obtained provided that the hypothesis that was not accepted holds. The p-value for accepting H_0 when $\sum_{i=1}^{n} x_i = d$ is $\Pr[\sum_{i=1}^{n} X_i \geq d \mid p < \theta]$, which is less than $F(n-d; n, 1-\theta) = 1 - F(d-1; n, \theta)$. The p-value for accepting H_1 is $\Pr[\sum_{i=1}^{n} X_i \leq d \mid p \geq \theta]$, which is at most $F(d; n, \theta)$. The following theorem justifies our choice of the constant c [13–Theorem 7.1]:

Theorem 1 (Minimization of p-value). *By choosing c to minimize the value of $|F(c; n, \theta) - 0.5|$ when testing $H_0 : p \geq \theta$ against $H_1 : p < \theta$ using a single sampling plan $\langle n, c\rangle$, the hypothesis with the lowest p-value is always accepted.*

In practice, it is unnecessary to compute c. It is easier simply to compute the p-value of each hypothesis and accept the hypothesis with the lowest p-value.

Example 1. Consider the problem of verifying $\Phi = \mathcal{P}_{\geq 0.9}\left[\top\ \mathcal{U}^{[0,100]}\ x{=}1\right]$ in a state satisfying $x{=}0$ for a "black-box" system that in reality is the continuous-time Markov chain shown in Fig. 2. The probability measure of trajectories starting in state $x{=}0$ and satisfying $\top\ \mathcal{U}^{[0,100]}\ x{=}1$ is $1 - e^{-1} \approx 0.63$, so the UTSL formula does not hold, but we would of course not know this unless we had access to the model. Assume that we are given a set of 100 truncated trajectories, of which 63 satisfy and 37 do not satisfy the path formula $\top\ \mathcal{U}^{[0,100]}\ x{=}1$. Thus, $n = 100$ and $d = 63$. The p-value for H_0 is $1 - F(62; 100, 0.9) \approx 1 - 10^{-13}$, while the p-value for H_1 is $F(63, 100, 0.9) \approx 5.48 \cdot 10^{-13}$. The hypothesis with the lowest p-value is H_1, so we conclude that Φ does not hold.

In the analysis so far we have assumed that the value of φ can be determined over all n truncated trajectories. Now, assume that we are unable to verify the path formula φ over some of the n truncated trajectories. This would happen

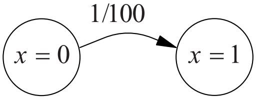

Fig. 2. A simple two-state continuous-time Markov chain

if we verify $\Phi \ \mathcal{U}^I \ \Psi$ over a trajectory that has been truncated before either $\neg\Phi \vee \Psi$ is satisfied or time exceeds all values in I. We cannot simply ignore such trajectories: it is assumed that the *entire* set of n trajectories is representative of the measure μ, but the subset of truncated trajectories for which we can determine the value of φ is not guaranteed to be a representative sample.

Example 2. Consider the same problem as in Example 1. Assume that we are given a set of 100 trajectories for the system that all have been truncated before time 50. Some of the trajectories, on average 39 in every 100, will satisfy $\top \ \mathcal{U}^{[0,100]} \ x{=}1$, while the remaining truncated trajectories will not contain sufficient information to determine the validity of $\top \ \mathcal{U}^{[0,100]} \ x{=}1$ over these trajectories. An analysis based solely on the trajectories over which the path formula can be decisively verified would be severely biased. If the number of positive observations is exactly 39, with 61 undetermined observations, we would wrongly conclude that Φ holds with p-value $1 - F(38; 39, 0.9) \approx 0.0164$, which implies a fairly high confidence in the result.

Let n' be the number of observations whose value we can determine and let d' be the sum of these observations. We then know that the sum of all observations, d, is at least d' and at most $d' + n - n'$. If $d' > c$, then hypothesis H_0 can safely be accepted. Instead of a single p-value, we associate an interval of possible p-values with the result: $[F(n' - d'; n, 1 - \theta), F(n - d'; n, 1 - \theta)]$. Conversely, if $d' + n - n' \leq c$, then hypothesis H_1 can be accepted with p-value in the interval $[F(d'; n, \theta), F(d' + n - n'; n, \theta)]$. In all other cases it is not clear which hypothesis should be accepted. We could then say that we do not have enough information to make an informed choice. Alternatively, we could accept one of the hypotheses with its associated p-value interval. We prefer to always make some choice, and we recommend choosing H_0 if $F(n - d'; n, 1 - \theta) \leq F(d' + n - n'; n, \theta)$ and H_1 otherwise. This strategy minimizes the maximum possible p-value. Alternatively, we could minimize the minimum possible p-value by instead choosing H_0 if $F(n' - d'; n, 1 - \theta) \leq F(d; n, \theta)$ and H_1 otherwise.

Example 3. Consider the same situation as in Example 2, with 39 positive and 61 undetermined observations. The p-value for accepting $\Phi = \mathcal{P}_{\geq 0.9}\big[\top \ \mathcal{U}^{[0,100]} \ x{=}1\big]$ as true lies in the interval $[F(0; 100, 0.1), F(61, 100, 0.1)] \approx [2.65 \cdot 10^{-5}, 1 - 3.77 \cdot 10^{-15}]$. For the opposite decision, we get $[F(39; 100, 0.9), F(100; 100, 0.9)] \approx [1.59 \cdot 10^{-35}, 1]$. Both intervals are almost equally uninformative, so no matter what decision we make, we will have a low confidence in the result. This is in sharp contrast to the faulty analysis suggested in Example 2, which lead to an acceptance of Φ as true with a low p-value.

Composite State Formulae. To verify $\neg\Phi$, we first verify Φ. If we conclude that Φ has a certain truth value with p-value pv, then we conclude that $\neg\Phi$ has the opposite truth value with the same p-value. To motivate this, consider the case $\neg\mathcal{P}_{\geq\theta}[\varphi]$. To verify $\mathcal{P}_{\geq\theta}[\varphi]$, we test the hypothesis $H_0 : p \geq \theta$ against $H_1 : p < \theta$ as stated above. Note, however, that $\neg\mathcal{P}_{\geq\theta}[\varphi] \equiv \mathcal{P}_{<\theta}[\varphi]$, which could be posed as the problem of testing the hypothesis $H_0' : p < \theta$ against $H_1' : p \geq \theta$. Since $H_0' = H_1$ and $H_1' = H_0$, we can simply negate the result of verifying $\mathcal{P}_{\geq\theta}[\varphi]$ while maintaining the same p-value (cf. [12]).

For a conjunction $\Phi \wedge \Psi$, we have to consider four cases. First, if we verify Φ to hold with p-value pv_Φ and Ψ to hold with p-value pv_Ψ, then we conclude that $\Phi \wedge \Psi$ holds with p-value $\max(pv_\Phi, pv_\Psi)$. Thus, we are no more confident in the result for $\Phi \wedge \Psi$ than we are in the results for the individual conjuncts. Second, if we verify Φ not to hold with p-value pv_Φ, while verifying that Ψ holds, then we base the decision for the conjunction on the result for Φ alone and conclude that $\Phi \wedge \Psi$ does not hold with p-value pv_Φ. The third case is analogous to the second with Φ and Ψ interchanged. Finally, if we verify Φ not to hold with p-value pv_Φ and Ψ not to hold with p-value pv_Ψ, then we conclude that $\Phi \wedge \Psi$ does not hold with p-value $\min(pv_\Phi, pv_\Psi)$. In this case, we have two sources (not necessarily independent) telling us that the conjunction is false. We have no reason to be less confident in the result for the conjunction than in the result for each of the conjuncts, hence the minimum.

For a mathematical derivation of the given expressions, we consider the formula $\mathcal{P}_{\geq\theta_1}[\varphi_1] \wedge \mathcal{P}_{\geq\theta_2}[\varphi_2]$. Let d_i denote the number of trajectories that satisfy φ_i. Provided we accept the conjunction as true, which means we accept each conjunct as true, the p-value for the result is

$$\Pr[\sum_{i=1}^{n} X_i^{(1)} \geq d_1 \wedge \sum_{i=1}^{n} X_i^{(2)} \geq d_2 \mid p_1 < \theta_1 \vee p_2 < \theta_2] \ . \tag{2}$$

To compute this p-value, consider the three ways in which $p_1 < \theta_1 \vee p_2 < \theta_2$ can be satisfied (cf. [12]). We know from elementary probability theory that $\Pr[A \wedge B] \leq \min(\Pr[A], \Pr[B])$ for arbitrary events A and B. From this fact, and assuming that pv_i is the p-value associated with the verification result for $\mathcal{P}_{\geq\theta_i}[\varphi_i]$, we derive the following:

1. $\Pr[\sum_{i=1}^{n} X_i^{(1)} \geq d_1 \wedge \sum_{i=1}^{n} X_i^{(2)} \geq d_2 \mid p_1 < \theta_1 \wedge p_2 < \theta_2] \leq \min(pv_1, pv_2)$
2. $\Pr[\sum_{i=1}^{n} X_i^{(1)} \geq d_1 \wedge \sum_{i=1}^{n} X_i^{(2)} \geq d_2 \mid p_1 < \theta_1 \wedge p_2 \geq \theta_2] \leq \min(pv_1, 1) = pv_1$
3. $\Pr[\sum_{i=1}^{n} X_i^{(1)} \geq d_1 \wedge \sum_{i=1}^{n} X_i^{(2)} \geq d_2 \mid p_1 \geq \theta_1 \wedge p_2 < \theta_2] \leq \min(1, pv_2) = pv_2$

We take the maximum over these three cases to obtain a bound for (2), which gives us $\max(pv_1, pv_2)$. For the same formula, but now assuming we have verified both conjuncts to be false, we compute the p-value as

$$\Pr[\sum_{i=1}^{n} X_i^{(1)} \leq d_1 \wedge \sum_{i=1}^{n} X_i^{(2)} \leq d_2 \mid p_1 \geq \theta_1 \wedge p_2 \geq \theta_2] \leq \min(pv_1, pv_2) \ . \tag{3}$$

If one conjunct has been verified to be false with p-value pv and the other conjunct has been verified to be true with p-value pv', then the conjunction is determined to be false with p-value pv. This is because the result for the entire conjunction depends only on the conjunct that has been verified to be false.

4.2 Verification with Nested Probabilistic Operators

If we allow nested probabilistic operators, verification of UTSL formulae for "black-box" stochastic discrete event systems becomes much harder. Consider the formula $\mathcal{P}_{\geq\theta}\bigl[\top\ \mathcal{U}^{[0,100]}\ \mathcal{P}_{\geq\theta'}[\varphi]\bigr]$. In order to verify this formula, we must test if $\mathcal{P}_{\geq\theta'}[\varphi]$ holds at some time $t \in [0, 100]$ along the set of trajectories that we are given. Unless the time domain T is such that there is a finite number of time points in a finite interval, then we potentially have to verify $\mathcal{P}_{\geq\theta'}[\varphi]$ at an infinite or even uncountable number of points along a trajectory, which clearly is infeasible. Even if $T = \mathbb{Z}^*$, so that we only have to verify nested probabilistic formulae at a finite number of points, we still have to take the entire prefix of the trajectory into account at each time point. We are given a fixed set of trajectories, and we can use only the subset of trajectories with a matching prefix to verify a nested probabilistic formula. It is thus likely that we will have few trajectories available to use for verifying nested probabilistic formulae. In the worst case, there will be only a single matching prefix, in which case the uncertainty in the result will be overwhelming.

Only if we assume that the "black-box" system is a Markov chain, which is a rather strong assumption, can we hope to have a significant number of trajectories available for the verification of nested probabilistic formulae. This is because, under the Markov assumption, we only have to take the last state along a trajectory prefix into consideration. Consequently, *any* suffix of a truncated trajectory starting at a specific state s can be regarded as representative of the probability measure $\mu(\{\langle s, 0\rangle\})$ for a Markov chain.

Another complicating factor for verifying $\mathcal{P}_{\geq\theta}[\varphi]$, where φ contains nested probabilistic operators, is that we cannot verify φ over trajectories without some uncertainty in the result. This means that we no longer obtain observations of the random variables X_i, as defined above, but instead we observe some other random variables Y_i, related to X_i through bounds on the observation error.

To compute a p-value for nested verification, we assume that $\Pr[Y_i = 0 \mid X_i = 1] \leq \alpha$ and $\Pr[Y_i = 1 \mid X_i = 0] \leq \beta$. We can make this assumption if we introduce indifference regions in the verification of nested probabilistic formulae and use the procedure described by Younes [13–Chap. 5] to verify path formulae over truncated trajectories. We have the following bounds [13–Lemma 5.7]: $p(1-\alpha) \leq \Pr[Y_i = 1] \leq 1 - (1-p)(1-\beta)$. The p-value for accepting $\mathcal{P}_{\geq\theta}[\varphi]$ as true when the sum of the observations is d is $\Pr[\sum_{i=1}^{n} Y_i \geq d \mid p < \theta]$, which is less than $F(n-d; n, (1-\theta)(1-\beta))$. The p-value for the opposite decision is $\Pr[\sum_{i=1}^{n} Y_i \leq d \mid p \geq \theta]$, which is at most $F(d; n, \theta(1-\alpha))$. Since $F(d; n, p)$ increases as p decreases, we see that the p-value increases as the error bounds α and β increase, which makes perfect sense. As was suggested earlier, we can minimize the p-value of the verification result by computing the p-values of both hypotheses and accept the one with the lowest p-value.

We can let the user specify a parameter δ_0 that controls the relative width of the indifference regions. A nested probabilistic formula $\mathcal{P}_{\geq \theta}[\varphi]$ is verified with an indifference region of half-width $\delta = \delta_0\theta$ if $\theta \leq 0.5$ and $\delta = \delta_0(1-\theta)$ otherwise. The verification is carried out using acceptance sampling as before, but with hypotheses $H_0 : p \geq \theta + \delta$ and $H_1 : p \leq \theta - \delta$. Instead of reporting a p-value, as is done for top-level probabilistic operators, we report bounds for the type I error probability of the sampling plan in use if H_1 is accepted and the type II error probability if H_0 is accepted. In our case, assuming a sampling plan $\langle n, c\rangle$ is used, the type I error bound is $1 - F(c; n, \theta + \delta)$ and the type II error bound is $F(c; n, \theta - \delta)$. As error bounds for the computation of the p-value for a top-level probabilistic operator, we simply take the maximum error bounds for the verification of the path formula over all trajectories.

5 Comparison with Related Work

The idea of using statistical hypothesis testing for verification of "black-box" systems was first proposed by Sen et al. [12]. This section highlights the differences between their approach and the approach presented in this paper.

First, consider the verification of a probabilistic formula $\mathcal{P}_{\geq \theta}[\varphi]$. Our approach is essentially the same as theirs: given a constant c, accept if $\sum_{i=1}^{n} X_i > c$ and reject otherwise. Their choice of c is different, however, and is based on the normal approximation for the binomial distribution. Their acceptance condition is $\sum_{i=1}^{n} X_i \geq n\theta$, which corresponds to choosing c to be $\lceil n\theta \rceil - 1$. Their algorithm, as a consequence, will under some circumstances accept a hypothesis with a larger p-value than the alternative hypothesis. By choosing c as we do, without relying on the normal approximation, we guarantee that the hypothesis with the smallest p-value is always accepted (Theorem 1). Consider $\mathcal{P}_{\geq 0.01}[\varphi]$, for example, with $n = 501$ and $d = 5$. Our procedure would accept the formula as true with p-value 0.562, while the algorithm of Sen et al. would reject it as false with p-value 0.614. It is important to note that their choice of c does not impact the soundness of their algorithm, but it may lead to counterintuitive results.

The second improvement over the method presented by Sen et al. is in the calculation of the p-value for the verification of a conjunction $\Phi \wedge \Psi$ when both conjuncts have been verified to be false. They state that the p-value is bounded by $pv_\Phi + pv_\Psi$, which is correct but unnecessarily conservative. There is no reason to believe that the confidence in the result for $\Phi \wedge \Psi$ would be *lower* (i.e. the p-value *higher*) if we are convinced that both conjuncts are false. We have shown that the p-value in this case is bounded by $\min(pv_\Phi, pv_\Psi)$.

Sen et al., in their handling of nested probabilistic operators, confuse the p-value with the probability of accepting a false hypothesis (generally referred to as the type I or type II error of a sampling plan). The p-value is *not* a bound on the probability of a certain test procedure accepting a false hypothesis. In fact, the test that both they and we use does not provide any useful bound on the probability of accepting a false hypothesis. Their analysis relies heavily on the ability to bound the probability of accepting a false hypothesis, and we have presented a way to provide such bounds by introducing indifference regions (rather than computing p-values) for nested probabilistic operators.

In addition, Sen et al. are vague regarding the assumptions needed for their approach to produce reliable answers. The fact that they treat any portion of a trajectory starting in s, regardless of the portion preceding s, as a sample from the same distribution, hides a rather strong assumption regarding the dynamics of their "black-box" systems. As we have pointed out, this is not a valid assumption unless we know that the system is a Markov chain. They also assume that truncated trajectories are sufficiently long so that a path formula can be verified fully over each truncated trajectory. We have removed this assumption and we have presented a procedure for handling situations when the value of a path formula cannot be determined over all truncated trajectories.

Finally, the empirical analysis offered by Sen et al. easily gives the reader the impression that a low p-value can be guaranteed for a verification result simply by increasing the sample size, even though the authors correctly state that a certain p-value *never* can be guaranteed. If we are unlucky, we may make observations that give us a large p-value even in cases when this is unlikely, and a large p-value may even be the most likely outcome in some cases. The empirical results of Sen et al. cannot be replicated reliably because there is no fixed procedure by which one can determine the sample size required to achieve a certain p-value. Their results give the false impression that their procedure is sequential, i.e. that the sample size automatically adjusts to the difficulty of attaining a certain p-value, when in reality they selected the reported sample sizes *manually* based on prior empirical testing (K. Sen, personal communication, May 20, 2004). It is therefore misleading to say that an algorithm for "black-box" verification is "faster" than a statistical model checking algorithm that is designed to realize certain *a priori* performance characteristics (such as the algorithm described by Younes and Simmons [15]).

6 Discussion

Sen et al. [12] were first to consider the problem of probabilistic verification for "black-box" systems. We have generalized their idea to a wider class of probabilistic systems that can be characterized as stochastic discrete event systems. Our most important contribution is to have given a clear definition of what constitutes a "black-box" system, and to have made explicit any assumptions making feasible the application of statistical hypothesis testing as a solution technique for verification of such systems.

The algorithm presented in this paper should not be thought of as an alternative to the statistical model checking algorithm proposed by Younes and Simmons [15] and empirically evaluated by Younes et al. [14]. The two algorithms are complementary rather than competing, and are useful under disparate sets of assumptions. If we cannot generate trajectories for a system on demand, then the algorithm presented here allows us to still reach conclusions regarding the behavior of the system. If, however, we know the dynamics of a system well enough to enable simulation, then we are better off with the alternative approach as it gives full control over the probability of obtaining an incorrect result.

References

1. Alur, R., Courcoubetis, C., and Dill, D. L. Model-checking for probabilistic real-time systems. In *Proc. 18th International Colloquium on Automata, Languages and Programming*, volume 510 of *LNCS*, pages 115–126. Springer, 1991.
2. Aziz, A., Sanwal, K., Singhal, V., and Brayton, R. K. Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.
3. 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.
4. Doob, J. L. *Stochastic Processes.* John Wiley & Sons, 1953.
5. Duncan, A. J. *Quality Control and Industrial Statistics.* Richard D. Irwin, fourth edition, 1974.
6. Grubbs, F. E. On designing single sampling inspection plans. *Annals of Mathematical Statistics*, 20(2):242–256, 1949.
7. Halmos, P. R. *Measure Theory.* Van Nostrand Reinhold Company, 1950.
8. Hansson, H. and Jonsson, B. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
9. Hogg, R. V. and Craig, A. T. *Introduction to Mathematical Statistics.* Macmillan Publishing Co., fourth edition, 1978.
10. Infante López, G. G., Hermanns, H., and Katoen, J.-P. Beyond memoryless distributions: Model checking semi-Markov chains. In *Proc. 1st Joint International PAPM-PROBMIV Workshop*, volume 2165 of *LNCS*, pages 57–70. Springer, 2001.
11. 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.
12. Sen, K., Viswanathan, M., and Agha, G. Statistical model checking of black-box probabilistic systems. In *Proc. 16th International Conference on Computer Aided Verification*, volume 3114 of *LNCS*, pages 202–215. Springer, 2004.
13. Younes, H. L. S. *Verification and Planning for Stochastic Processes with Asynchronous Events.* PhD thesis, Computer Science Department, Carnegie Mellon University, 2005. CMU-CS-05-105.
14. Younes, H. L. S., Kwiatkowska, M., Norman, G., and Parker, D. Numerical vs. statistical probabilistic model checking. *International Journal on Software Tools for Technology Transfer*, 2005. Forthcoming.
15. Younes, H. L. S. and Simmons, R. G. Probabilistic verification of discrete event systems using acceptance sampling. In *Proc. 14th International Conference on Computer Aided Verification*, volume 2404 of *LNCS*, pages 223–235. Springer, 2002.