“Black-Box” Probabilistic Verification

Håkan L. S. Younes

September 2004

CMU-CS-04-162$_{3}$

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This report is based upon work supported by DARPA and ARO under contract no. DAAD19–01–1–0485, and a grant from the Royal Swedish Academy of Engineering Sciences (IVA). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the sponsors.

Keywords: Probabilistic verification, model checking, temporal logic, statistical hypothesis testing, stochastic processes, discrete event systems

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. The properties are expressed using a variation of PCTL, the Probabilistic Computation Tree Logic. We present a general model of stochastic discrete event systems, which encompasses both discrete-time and continuous-time processes, and we provide a semantics for PCTL interpreted over this model. *Ow* presentation is both a generalization of and an improvement over some recent work by Sen et al. on probat stic verification of "black-box" systems.

1 Introduction

Stochastic processes are used to model phenomena in nature that involve an element of chance, such as the throwing of a die, or are too complex to fully capture in a deterministic fashion, such as 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 (Alur et al. 1991), PCTL (Hansson and Jonsson 1994), and CSL (Aziz et al. 2000; Baier et al. 2003), 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 (Hansson and Jonsson 1994), continuous-time Markov chains (Baier et al. 2003; Kwiatkowska et al. 2002), semi-Markov processes (Infante Lόpez et al. 2001), generalized semi-Markov processes (Alur et al. 1991), and stochastic discrete event systems in general (Younes and Simmons 2002).

Given a stochastic process, we are often interested in knowing if certain probabilistic properties hold. For a computer network, we may want to know that the probability of exhausting bandwidth over a communication link is below some threshold. We can also associate a deadline with a probabilistic property, for example that a message arrives at its destination within 15 seconds after it is sent out with probability at least 0.9. Properties of this type can be verified using either numerical or statistical solution techniques, as discussed by Younes et al. (2004). Numerical techniques provide highly accurate results, but rely on strong assumptions regarding the dynamics of the systems they are used to analyze. Statistical techniques only require that the dynamics of a system can be simulated, and can therefore be used for a larger class of stochastic processes. The result produced by a statistical method is only probabilistic, however, and attaining high accuracy tends to be costly.

For some systems, it may not even be feasible to assume that we can simulate their behavior. Sen et al. (2004) consider the model checking problem for such "black-box" systems. It is assumed of a "black-box" system that it cannot be controlled to generate execution traces, or *trajectories,* on demand starting from arbitrary states. This is a reasonable assumption for a system that has already been deployed, and for which we are only given 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 for probabilistic model checking proposed by Younes and Simmons (2002) cannot be used for verification of "black-box" systems, however, because it depends on the ability to generate trajectories on demand.

Sen et al. (2004) present an alternative solution method for verification of "black-box" systems based on statistical hypothesis testing with fixed sample sizes. We improve upon their algorithm in several ways, for example by making sure to always accept the most likely hypothesis, and we present a procedure for verifying nested probabilistic properties, which unlike that of Sen et al. actually works. The differences between the two competing approaches are discussed in detail towards the end of this paper, where we also make an effort to explain why Sen et al.'s comparison of their algorithm with the statistical model checking procedure used by Younes et al. (2004) is misguided. These two solution methods, while both based on statistical hypothesis testing, are simply not comparable in a meaningful way because the "black-box" approach does not give any a priori correctness guarantees.

We start by presenting a general model of stochastic discrete event systems that encompasses both discrete-time and continuous-time processes. We give a clear definition of a "black-box" system in terms of this model, and we define the syntax and semantics of a logic for expressing properties of general discrete event systems. Our logic has essentially the same syntax as Hansson and Jonsson's (1994) PCTL, and we call it PCTL as well because it includes the original version of the logic as a special case, but it also includes CSL (without the steady-state operator) as defined by Baier et al. (2003). The algorithm we present for verification of "black-box" systems can handle the full logic, including properties without finite time bounds, although the accuracy of the result for such properties may very well be poor. Our algorithm, like that of Sen et al. (2004), does in fact make no guarantees regarding accuracy. Instead of respecting some a priori bounds on the probability of error, the algorithm computes a *p*-value for the result, which is a measure of confidence. This is really the best we can do, provided that we cannot generate trajectories for the system as we see fit and instead are restricted to use a predetermined set of trajectories.

2 Stochastic Discrete Event Systems

A *stochastic process* is in principle any process that evolves over time, and whose evolution we can follow and predict in terms of probability (Doob 1942, 1953). 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.

A *stochastic discrete event system* is a specific type of stochastic process that can be thought of as occupying a single state for some duration of time until 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. We call this a *discrete event* system because the state change is discrete rather than continuous and is caused by the triggering of an event.

2.1 Trajectories

Mathematically, we define a stochastic process as a family of random variables *X* — {X_t \ *t* G T}. The index set *T* represents time and is typically the set of non-negative integers, Z*, for discrete-time stochastic processes and the set of non-negative real numbers, [0, oo), for continuous-time stochastic processes. For each *t* G *T* we have a random variable X_t representing the chance experiment of observing the stochastic process at time *t*. The range of X_t is a set *S* of states that the stochastic process can occupy, which can be infinite or even uncountable. A *trajectory* or *sample path* of a stochastic process is any realization {x_t G 5 | *t* G T} of the family of random variables *X*.

The trajectory of a stochastic discrete event system is *piecewise constant* and can therefore be represented as a sequence *a* = {(s_o,£o), (^{s}i?*i)> • • •}> $^{with\ s_*\ e}$ *S* ^{and} *U* € *T*\ {0}. Figure 1 plots part of a trajectory for a simple queuing system. Let

(1) Ti = { 0 if i = 0 ; ^ ∠$_{\bar{j}}$U ifi>o ,

i.e. *T{* is the time at which state *S{* is entered and *U* is the duration of time for which the process remains in *S{* before an event triggers a transition to state *Si+i*. *A* trajectory *a* is then a realization of *X* with x_t = *Si* for *Ti* < *t* < *Ti* + *U*. According to this definition, trajectories of stochastic discrete event systems are *right-continuous*. *A* finite trajectory is a sequence *a* = *{(so, to),...,* (s_n, oo)} 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* >_^*27=0* ^-

Note that if *Ylt^o* ** $^<$ oo ^$^{or\ an}$ infinite trajectory a, which is possible if *T* is the non-negative rational or real numbers, then x_t is not well-defined for all *t* G *T*. For this to happen, however, an infinite sequence of events must occur in a finite amount of time, which is unrealistic for any physical system. Hoel et al. (1972)

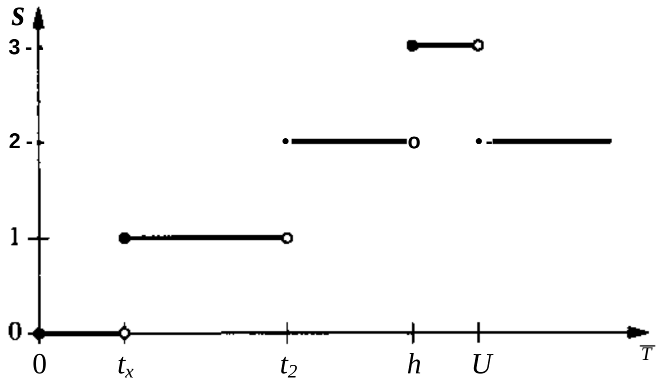

Figure 1: A trajectory for a simple queuing system with arrival events occurring at *ti,* ^2 and £3 and a departure event occurring at *t±.* The state of the system represents the number of items in the queue.

use the term *explosive* to describe processes for which such sequences can occur with non-zero probability. It is common to assume time divergence for infinite trajectories of real-time systems (cf. Alur and Dill 1994), i.e. that the systems are non-explosive, and most finite-state systems satisfy this property by default.

2.2 Probability Space and "Black-Box" Probabilistic Systems

A prefix of a trajectory *a* — {(sn?*o)> (si,*i), • • •} is a sequence a$<_T$ = *{(sQ,t$'_0$),..., (s$_k$,t$'_k$)}$_9$* with *s^* = *Si* for all *i* ≤ *k$_9$ Yli=o* *i = T' *« = **· ^or a^ * ·< ^» anc* ^k < *&• ^et *Poih{a$<_T$)* denote the set of trajectories with common prefix <r$<_T$. This set must be *measurable* for probabilistic model checking to make sense, and we assume that a probability measure /1 over the set of trajectories with common prefix exists. This is hardly a severe restriction as such a measure can be defined for systems of practical interest, although the precise definition thereof is not required for the approach to probabilistic model checking considered in this paper. In fact, the lack of knowledge of the probability measure over sets of trajectories can be seen as the defining characteristic of a "black-box" probabilistic system. If we had complete knowledge of this probability measure, then the system under consideration would not be a black box to us. This leads us to make the following definition.

Definition 1 ("Black-box" probabilistic system). A stochastic discrete event system for which the probability measure *JJL* over sets of trajectories with common prefix is unknown and cannot even be sampled from is called a "black-box" probabilistic system.

A measurable space is a set *ft* with a a-algebra *FQ* of subsets of *ft* (Halmos 1950). *A probability space* is a measurable space (fi,^h) and a probability measure *\i* that assigns a value in the interval [0,1] to the elements of *T^* with /x(0) = 0, /i(fi) = 1, and *ji(E)* = *Yli^i^i)* ^ ^b^2>... are countably many pairwise disjoint sets in *T^* and *E* is their union. When we say that a set *Q* must be measurable, we really mean that there must be a a-algebra for the set. The elements of this a-algebra are the measurable subsets
of a

A stochastic discrete event system is measurable if the sets 5 and *T* are measurable. We can show this by defining a cr-algebra over the set of trajectories with common prefix <j$<_T$ = {(50, to),..., *(s$_k$, t$_k$)}$_9$* denoted *Path(a$\leq_r$),* as follows. Let *Ts* be a a-algebra over the state space 5, and let *TT* be a a-algebra over the index set *T* of the stochastic process. Such a-algebras exist if *S* and *T* are measurable sets, which by assumption they are. Then C(a$<_r$, J$_{fe}$,Sfc+i,... ,/$_n$-i,5$_n$), with *Si* G *F$_s$* and *I$_{\{}$* *e* *T$_T$*, denotes the set of trajectories $\sigma = \{\langle s'_0, t'_0\rangle, \langle s'_1, t'_1\rangle, \ldots\}$ such that $s'_i = s_i$ for $i \leq k$, $s'_i \in S_i$ for $k < i \leq n$, $t'_i = t_i$ for $i < k$, $t'_k > t_k$, and $t'_i \in I_i$ for $k \leq i < n$. In other words, $C(\sigma_{\leq\tau}, I_k, S_{k+1}, \ldots, I_{n-1}, S_n)$ is a subset of $\mathit{Path}(\sigma_{\leq\tau})$. The sets $C(\sigma_{\leq\tau}, I_k, S_{k+1}, \ldots, I_{n-1}, S_n)$ are the elements of a σ-algebra over the set $\mathit{Path}(\sigma_{\leq\tau})$ with set operations applied element-wise, for example $C(\sigma_{\leq\tau}, I_k, S_{k+1}, \ldots, I_{n-1}, S_n) \cup C(\sigma_{\leq\tau}, I'_k, S'_{k+1}, \ldots, I'_{n-1}, S'_n) = C(\sigma_{\leq\tau}, I_k \cup I'_k, S_{k+1} \cup S'_{k+1}, \ldots, I_{n-1} \cup I'_{n-1}, S_n \cup S'_n)$.

3 Properties of Stochastic Discrete Event Systems

A stochastic discrete event system can be specified as a triple $\langle S, T, \mu\rangle$, where S is a set of states, T is a time domain, and μ is a probability measure over sets of trajectories with common prefix. We typically 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 PCTL for a factored stochastic discrete event system $\mathcal{M} = \langle S, T, \mu, SV, V\rangle$ as

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

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 PCTL formulae can be derived in the usual way. For example, $\bot \equiv (x = v) \wedge \neg(x = v)$ for some $x \in SV$ and $v \in D_x$, $\top \equiv \neg\bot$, $\Phi \vee \Psi \equiv \neg(\neg\Phi \wedge \neg\Psi)$, $\Phi \rightarrow \Psi \equiv \neg\Phi \vee \Psi$, $\mathcal{P}_{\bowtie\theta}\left[\Phi\,\mathcal{U}\,\Psi\right] \equiv \mathcal{P}_{\bowtie\theta}\left[\Phi\,\mathcal{U}^T\,\Psi\right]$, and $\mathcal{P}_{<\theta}\left[\varphi\right] \equiv \neg\mathcal{P}_{\geq\theta}\left[\varphi\right]$.

The standard logic operators have their usual meaning. $\mathcal{P}_{\bowtie\theta}\left[\varphi\right]$ 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 PCTL formula, relative to a factored stochastic discrete event system $\mathcal{M}$, is defined in terms of a satisfaction relation $\models_{\mathcal{M}}$ between trajectory prefixes and PCTL formulae:

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

The above definition relies on a satisfaction relation $\sigma, \tau \models_{\mathcal{M}} \varphi$ such that $\langle \sigma, \tau, \varphi\rangle \in\ \models_{\mathcal{M}}$ iff σ satisfies φ starting at time τ. This satisfaction relation for path formulae is defined as follows:

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

Note that 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 defined by Alur et al. (1991). Depending on the probability measure μ, Φ may very well 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, except in special cases, to verify $ and \& at discrete points along a trajectory.

If $ and \£ are both free of any probabilistic operators, then it is always sufficient to verify the two formulae once in each state along a trajectory in order to verify <3> U^1 *. The same holds true if

(2) $$\hat{}Path(\{(s_0, \mathrm{t_o}>, \ldots, (s_k, **)\})) = n(Paih(\{(s_k,0)\}))$$

for all trajectory prefixes *{(so, to),..., (s_k, t_k)}*. This is the case if *M* is a Markov chain as (2) simply is a formulation of the Markov property. Our semantics for PCTL interpreted over general stochastic discrete event systems therefore coincides with the semantics for PCTL interpreted over discrete-time Markov chains (Hansson and Jonsson 1994) and CSL interpreted over continuous-time Markov chains (Baier et al. 2003), provided we choose the time domain *T* appropriately.

A PCTL model checking problem is typically specified as a triple *(M,* s, <E>) with the problem being to verify if <E> holds for *M* provided that execution starts in state s, i.e. *{(s,* 0)} \=*M* $• We often use *s* \= $ as a short form for the latter, leaving out *M.* when it is clear from the context which system is involved in the model checking problem.

4 Statistical Model Checking for "Black-Box" Stochastic Systems

We refer to a stochastic discrete event system At as a "black-box" system if we lack an exact definition of the probability measure *x* over sets of trajectories of *M.* We assume that we cannot even sample trajectories according to */JL* as earlier stated in Definition 1. Thus, in order to solve a model checking problem *s* \= $ for a "black-box" system *M,* we must rely on an external source to provide us with a set of trajectories for *M* that start in state *s.* We assume that trajectories cannot be generated on demand, but that we are provided with a finite set of *n* trajectories. This sample of size *n* must of course be representative of the probability measure */j,(Path({(s,Q)})),* and we must trust our external source to provide us with a representative set of trajectories. We further assume that we are only provided with *truncated* trajectories, because infinite trajectories would require infinite memory to store.

We will use statistical hypothesis testing to solve a model checking problem *s* (= $ 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 below computes a *p*-value for a model checking result, which is a value in the interval [0,1] with values closer to 0 representing higher confidence in the result and a *p*-value of 0 representing certainty (Hogg and Craig 1978, pp. 255-256). We start by assuming that $ is free of nested probabilistic operators. Later on, we consider PCTL formulae with nested probabilistic operators, which as it turns out cannot be handled in a meaningful way without making rather strong assumptions regarding the dynamics of the "black-box" system.

4.1 PCTL without Nested Probabilistic Operators

Given a state s, verification of a PCTL formula *x* ~ *v* is trivial. We consider the remaining three cases in more detail, starting with the probabilistic operator *V^e* [•]• Recall that the objective is to produce a Boolean result annotated with a *p*-value.

4.1.1 Probabilistic Operator

Consider the problem of verifying the PCTL formula *V^e* M in state *s* of a stochastic discrete event system *M.* Let *Xi* be a random variable representing the verification of the path formula *cp* over a trajectory for

M drawn according to the probability measure *fJ>(Path({(s,0)})).* If we choose *Xi* -= 1 to represent the fact that <*p* holds over a random trajectory, and *Xi* = 0 to represent the opposite fact, then *Xi* is a *Bernoulli variate* with parameter p = ^({cr G *Path({(s,0)})* | j,0 (= </?}), i.e. PrpQ = 1] = *p* and Pr[X^ = 0] = 1 - *p.* In order to verify *V^e [<p]*$_9$ we can make observations of *Xi* and use statistical hypothesis testing to determine if *p* ixi *8* is likely to hold. An observation of *X^* denoted *xu* is the verification of *ip* over a specific trajectory a*. If a^ satisfies the path formula <£>, then 2^ = 1, otherwise #$ = 0.

In our case, we are given *n* truncated trajectories for a "black-box" system that we can use to generate observations of *X*. Each observation is obtained by verifying the path formula <*p* over one of the truncated trajectories. This is straightforward given a truncated trajectory {(s$_{\mathrm{Oj}}$ *to),* • • •, (sfc-i,*fc-i), «&}> provided that <? does not contain any probabilistic operators. For <*p* = *X*^{1} $, we just check if *to* G / and *s*± |= $. For <? = $ Z^^{7} 'E>, we traverse the trajectory until we find a state s$_2$ such that one of the following conditions holds, with *T* defined as in (1) to be the time at which state *Si* is entered:

1. **(*si* t= -.*) A((*Ti* £ /) v (*si* f= ¬Ψ))**
2. *(Ti el) A (si* \= *)
3. $((T_i, T_{i+1}) \cap I$ /\) $\wedge (s_i \models \Phi) \wedge (s_i \models \Psi)$

In the first case, *&U*^{1} ^ does not hold over the trajectory, while in the second two cases the time-bounded until formula does hold. Note that we may not always be able to determine the value of *(p* over all trajectories because the trajectories that are provided to us are assumed to be truncated.

We consider the case *V>_$ [<p]* in detail, noting that *V≤ o [<p]* can be handled in the same way simply by reversing the value of each observation. We want to test the hypothesis *Ho* : *p* ≥ *8* against the alternative hypothesis *Hi* : *p* < *6* by using the *n* observations *x*,..., *x*$_n$ of the Bernoulli variates *Xi*,...,*X*$_n$. To do so, we specify a constant c. If *YH=i* ^{x}*i* *^{s} g$^{\mathrm{reater}}$ than c, then hypothesis *Ho* is accepted, i.e. *V>_e [f]* is determined to hold. Otherwise, if the given sum is at most c, then hypothesis *Hi* is accepted meaning that *V≥ $ [(p]* is determined not to hold. The constant c should be chosen so that it becomes roughly equally likely to accept *Ho* as if i if *p* equals *6.* The pair (n, c) is typically called a *single sampling plan* in the quality control literature (Montgomery 1991).

The probability distribution of a sum of *n* Bernoulli variates with parameter *p* is a binomial distribution with parameters *n* and p, denoted *B(n,p).* The probability of 5ZILi *Xi* being at most c is therefore given by the cumulative distribution function for *B(n,p):*

(3) *F^n^^* $\sum_{t=o}^{c} \binom{n}{i} p^i (1-p)^{n-i}$

Thus, with probability *F(c;n,p)* we accept hypothesis *Hi* using a single sampling plan *(n,c),* and consequently hypothesis ifo is accepted with probability 1 - *F(c;n,p)* by the same sampling plan. Ideally, we should choose *c* such that *F(c;* n, *9)* = 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, *6)* - 0.5| is minimized. We can readily compute the desired *c* using (3).

We now have a way to decide whether to accept or reject the hypothesis that *V>_Q* [<?] holds, but we also want to report a value reflecting the confidence in our decision. For this purpose, we compute the *p*-value for a 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 if$_0$ when £?=i *x*$_{\{}$ = *d* is PrEILi *X*$_{\{}$ ≥ *d* \ *p* < *0]* < *F(n - d]n,l - 8)* = *1* - *F(d* - l;n,0), while

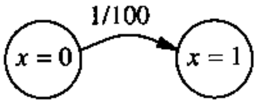

Figure 2: A simple two-state continuous-time Markov chain.

the p-value for accepting *Hi* is PrE^$_{=1}$ $X_{\{}$ <_*d* \ *p* ≥ *9]* ≤ F(d;n,0). The following theorem provides justification for our choice of the constant c.

Theorem 1 (Minimization of p-value). *By choosing c to minimize* |F(c; n, 0) — 0.5| *when testing Ho* : p ≥ *9 against H\ : p < 9 using a single sampling plan* (*n^c*)$_f$ *the hypothesis with the lowest p-value is always accepted.*

Proof. Hypothesis *H* is only accepted if *d* ≤ c, which means that the p-value for *Hi* under these circumstances is at most F(c; n, 0). The p-value for *Ho* if *d* ≤ *c* would be at least 1 - *F(c* — 1; ra, 0). We know that F(c-l;n,0) < *F(c;n,9)* and by assumption that *F(c*- l;n,0)-O.5| > |F(c;n,0)-O.5|. It follows that F(c; n, *9)* < 1 - *F(c* - 1; n, *9)* as required. For d > c, the p-value for acceptance of *Hi* would be at least *F(c* + 1; n, *9).* The p-value for acceptance of *Ho* when d > c, on the other hand, is at most 1 - *F(c;* n, 0). We know that F(c+l;n,0) > *F(c;n,9)* and by assumption that |F(c+l;n,0) -0.5| > |F(c;n,0)-O.5|. Consequently, 1 - F(c; n, 0) < *F(c*+1; n, 0) and our choice of c ensures that the hypothesis with the lowest p-value is always accepted. •

In the analysis so far we have been assuming that the value of *cp* can be determined over all *n* truncated trajectories that we are given. Now, consider the case when we are unable to verify the path formula <₽ over some of the n truncated trajectories. This would happen if we are verifying $ U^1 ^ over a trajectory that has been truncated before either -•$ V ^ is satisfied or time exceeds all values in /. We cannot simply ignore such trajectories: it is assumed that the *entire* set of *n* trajectories is representative of the measure /i, but the subset of truncated trajectories for which we can determine the value of <*p* is not guaranteed to be a representative sample for this measure.

For example, consider the problem of verifying the PCTL formula $ = P^o.9 [T Z^$^{'0,100}$! *x=l]* in a state satisfying *x=0* for a "black-box" system that in reality is the continuous-time Markov chain shown in Figure 2. The probability measure of trajectories starting in state *x=0* and satisfying T i^I^{0,100}! *x=l* is 1 - e"^{1} « 0.63 for this system, so the PCTL formula does not hold, but we would of course not know this unless we had access to the model. Assume that we are provided with a set of 100 truncated trajectories for the system, and that all trajectories have been truncated before time 50. Some of these trajectories, on average roughly 39 in every 100, will satisfy the path formula T ZVl^{0,100}! *x=*l_9 while the remaining truncated trajectories will not contain sufficient information for us to determine the validity of the path formula 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 $ holds with p-value 1 - F(38; 39,0.9) « 0.0164, which implies a fairly high confidence in the result.

Let n^f be the number of observations whose value we can determine and let *d!* be the sum of these *n'* observations. We then know that the sum of all observations, d, is at least *d!* and at most *d!* + *n* — n', i.e. *d e [d!,*d^f + *n* — n^r\. If *d!* > c, then hypothesis *Ho* can be safely accepted. Instead of a single p-value, we associate an interval of possible y^-values with the result: *[F(n'—d'* n, 1—0), *F(n—*d^f*;* n, 1—0)]. Conversely, if *d!* + n - n^f ≤ c, then hypothesis *Hi* can be accepted with p-value in the interval *[F(d'* n, 0), *F{d!* + *n*-n'; n, 0)]. If, however, *d!* _< *c* and *d!* + *n-n'* > c, then it is not clear which hypothesis should be accepted.

We could in this case 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_o if *F(n - d!\ *n, 1 - *0)_< F(d^f + n-n';* n, *0)* and *H* otherwise. This strategy minimizes the maximum possible p-value. Alternatively, we could minimize the minimum possible p-value by instead choosing H_o if *F(n^f — d!\n,* 1 - *0) _< F(d\n,0)* and *H* otherwise. Note that this way of treating truncated trajectories makes our approach work even for unbounded until formulae \$ *U* \I>, although we would typically expect the result to be highly uncertain for such formulae.

Consider the same problem as before, with 39 positive and 61 undetermined observations and assuming the system behaves like the Markov chain shown in Figure 2. The p-value interval for accepting the PCTL formula \$ = *V>$_0$.9* [T *U^^{100}^ x=l]* as true is [F(0; 100,0.1),F(61,100,0.1)] « [2.65 • 10~^{5},1 - 3.77 • 10~^{15}]. For the opposite decision, we get the p-value interval [F(39; 100,0.9), F(100; 100,0.9)] w [1.59 • 10~^{35},1]. Both intervals are almost equally uninformative, so no matter what decision we make, we will have a high uncertainty in the result. We would accept 3> as true if we prefer to minimize the maximum possible *p*-value, and we would reject \$ as false if we instead prefer to minimize the minimum possible p-value, but in both cases we have a maximum p-value well above 0.5. This is in sharp contrast to the faulty analysis suggested earlier, which lead to an acceptance of \$ as true with a low *p*-value.

4.1.2 Negation

To verify ->\$, we first verify \$. If we conclude that \$ has a certain truth-value with p-value pv_9 then we conclude that ->\$ has the opposite truth-value with the same *p*-value. To motivate this, consider the case -^*V>o [<p].* To verify *V>e [ip],* we test the hypothesis *Ho : p_> 6* against *Hi : p < 0* as stated above. Note, however, that ~^*V>o [<p]* = *V$_<$e* [y>], which could be posed as the problem of testing the hypothesis *H'$_o$: p < 0* against *H[: p> 0.* Since *H$^f_0$ = Hi* and *H[= Ho,* we can simply negate the result of verifying *V> e [y>* while maintaining the same *p*-value.

4.1.3 Conjunction

For a conjunction \$ A \£, we have to consider four cases. First, if we verify \$ to hold with *p*-value *pv\$* and \I/ to hold with p-value *pv^,* then we conclude that \$ A \I> holds with p-value *Ynax(pv^$_J$pv^).* Second, if we verify \$ not to hold with *p*-value pv_9 while verifying that * holds, then we conclude that \$ A * does not hold with *p*-value *pv.* The third case is analogous to the second with \$ and ^ interchanged. Finally, if we verify \$ not to hold with *p*-value *pv\$* and * not to hold with *p*-value *pv^,* then we conclude that \$ A \I> does not hold with *p*-value min(pi/\$,pv^).

Before deriving the given expressions for the *p*-values associated with the verification result of a conjunction, let us give an intuitive justification. In order for \$ A * to hold, both \$ and * must hold, so we cannot be anymore confident in the result for \$ A ^ than we are in the result for the individual conjuncts, thus the maximum in the first case. To conclude that \$ A * does not hold, however, we only need to be convinced that one of the conjuncts does not hold. In case we think exactly one of the conjuncts holds, then the result for the conjunction will be based solely on this conviction and the p-value for the conjunct we think holds should not matter. This covers the second and third cases. In the fourth case, we have two sources (not necessarily independent) telling us that the conjunction is false. We therefore 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 in this case.

For a mathematical derivation of the given expressions, we consider the formula *V> e_1 [<Pi]* A *V> o_2* [^2]- Let *d{* denote the number of trajectories that satisfy *ty.* Provided we accept the conjunction as true, which means we accept each conjunct as true, the p-value for this result is

$$\Pr[\sum_{i=1}^{n} X_i^{(1)} \geq d_1 \wedge \sum_{i=1}^{n} X_i^{(2)} \geq d_2 \mid p_1 < \theta_1 \vee p_2 < \theta_2] \ . \tag{4}$$

To compute this p-value, we consider the three ways in which $p_1 < \theta_1 \vee p_2 < \theta_2$ can be satisfied (cf. Sen et al. 2004). We know from elementary probability theory that

$$\Pr[A \cap B] \leq \min(\Pr[A], \Pr[B]) \tag{5}$$

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

We take the maximum over these three cases to obtain a bound for (4), 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

$$\Pr[\sum_{i=1}^{n} X_i^{(1)} \leq d_1 \wedge \sum_{i=1}^{n} X_i^{(2)} \leq d_2 \mid p_1 \geq \theta_1 \wedge p_2 \geq \theta_2] \ . \tag{6}$$

It follows immediately from (5) that $\min(pv_1, pv_2)$ is a bound for (6), which is the desired result.

4.2 PCTL with Nested Probabilistic Operators

If we allow nested probabilistic operators, PCTL model checking for "black-box" stochastic discrete event systems becomes much harder. Consider the formula $\mathcal{P}_{\geq \theta}\left[\top\ \mathcal{U}^{[0,100]}\ \mathcal{P}_{\geq \theta'}[\varphi]\right]$. 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 only use the subset of trajectories with a matching prefix to verify a nested probabilistic formula. This means that we will have very few trajectories available to use for the verification of nested probabilistic formulae, most likely only one if the prefix is long, 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 to make, 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, in the set provided to us by an external source, can be regarded as representative of the probability measure $\mu(\{\langle s, 0\rangle\})$.

Another complicating factor in the verification of V> $ [<p]9 where <p contains nested probabilistic operators, is that we cannot verify (p over trajectories without some uncertainty in the result. This means that we do no longer obtain observations of the random variables X\ as defined above, but instead we observe some other random variables Y{ with quite different distributions. We accept V>$ [<?] as true if ŶA=I ^ > c for some constant c, and we reject the same formula as false otherwise. We can choose c as previously, but what is the p-value of the decision?

To compute a p-value for nested verification we assume that Pr[Y; = 0 | a, r \= (p] < a and Pr[Yj = I | a, r \fi ip] ≤ (3. We can make this assumption if we introduce *indifference regions* in the verification of probabilistic formulae that are part of (p. Under the given assumption, we can use the total probability formula to derive bounds for Pr[^ = 1]: p(l - a) ≤ Pr[Y* = 1] < 1 - (1 - p)(l - /?). The p-value for accepting V>e [p] as true when the sum of the observations is d is PrE^Li Y% ≥ d \ p < 9] < F(n — d;n, (1-(9)(1-/3)). The p-value for the opposite decision is PT[Y%=1 Y%o < d \ p > 9] < F(d;n,9(l-a)). Since F{d\n,p) increases as p decreases, we see that the p-value increases as the error bounds a and (3 increase, which makes perfect sense. While we said that c can be chosen as previously, this choice does no longer guarantee that the hypothesis with the lowest p-value is accepted. To minimize the p-value of the result, we can simply compute the p-values of the two hypotheses and accept the hypothesis with the lowest p-value.

We can let the user specify a parameter So that controls the relative width of the indifference regions. A probabilistic formula V>$ [<p] is verified with indifference region of half-width 5 = 8$9 if 9 ≤ 0.5 and 5 = So(l - 9) otherwise. The verification is carried out using acceptance sampling as before, but with hypotheses Ho : p ≥ 9 + 5 and Hi : p ≤ 9 — 8. Instead of reporting a p-value, we report bounds for the type I error probability of the sampling plan in use if Hi is accepted and the type II error probability if Ho is accepted. The type I error of a sampling plan is defined as the maximum probability of accepting Hi when Ho holds, while the type II error is defined as the maximum probability of accepting Ho when Hi holds. In our case, assuming a sampling plan (n, c) is used, the type I error is F(c; n, 9 + 5) and the type II error is F(c; n,9 — 5). The error probabilities can be used in the same way as p-values to obtain error probabilities for compound state formulae. A path formula can be treated as a compound state formula, as suggested by Younes and Simmons (2002), which allows us to derive error bounds for the verification of path formulae over trajectories as well. 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 Related Work

The idea of using statistical hypothesis testing for probabilistic model checking of "black-box" systems was recently proposed by Sen et al. (2004). Their work is the inspiration for the current paper, although mostly for the wrong reasons. It is in fact the many hidden assumptions, outright errors, and misleading empirical evaluation of Sen et al.'s presentation that has prompted our interest in the subject.

First, consider the verification of a probabilistic formula V>e [ip]. Their approach is essentially the same as ours: given a constant c, accept if YH=i Xi > c and reject otherwise. Their choice of c is different, however, and is essentially based on De Moivre's (1738) normal approximation for the binomial distribution. Their acceptance condition is Y12=i %-i > n9, which corresponds to choosing c to be \n&\ — 1. The mean of the binomial distribution B(n, 9) is n0, so this would be the right thing to do if ŶA=I xi can be assumed to have a normal distribution. De Moivre showed that this is approximately the case for large n if X{ are Bernoulli variates, but the approximation is poor for moderate values of n or if 9 is not close to 0.5.

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 (The«. :em 1). Consider the formula P>o.oi M> for example, with n = 501 and *d* = 5. Our procedure would accept the formula as true with p-value 0.562, while the the algorithm of Sen et al. would reject the formula as false with p-value 0.614. The difference is not of great significance, but it is still worth pointing out because it demonstrates the danger of using the normal approximation for the binomial distribution. With today's fast digital computers, it is hard to motivate the use thereof. Our procedure is therefore an improvement over the algorithm of Sen etal.

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 <$> A \I/ when both conjuncts have been verified to be false. They state that the *p*-value is *pv$> + pv^*, but this is too conservative. There is no reason to believe that the confidence in the result for $ A ^ 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^^,^^), which intuitively makes more sense.

Sen et al.'s handling of nested probabilistic operators is just plain wrong. They confuse the p-value with the probability of accepting a false hypothesis (generally referred to as the type I or 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 a 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, so it breaks down completely. We have proposed a way to cope with this by introducing indifference regions for nested probabilistic operators.

In addition to getting the verification of nested probabilistic operators wrong, Sen et al. are very vague regarding the assumptions necessary to make their approach produce a reliable answer. 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 being studied is a Markov chain. It also appears as if they only consider truncated trajectories over which they can fully verify a path formula, and this can introduce a bias that very well may invalidate the conclusion they reach regarding the truth-value of a probabilistic formula. We have made this quite clear in our exposition, and we have presented a sound procedure for handling the fact that the value of a path formula may not be determined over all truncated trajectories that are presented to us.

Finally, the empirical analysis offered by Sen et al. is misleading. They give the reader the impression that a certain *p*-value can be guaranteed for a verification result simply by increasing the sample size. This violates the premise of a "black-box" system stated by the authors themselves earlier in their paper, namely that trajectories cannot be generated on demand. More important, though, is the fact that a certain p-value *never* can be guaranteed. The p-value is not a property of a test, but simply a function of a specific set of observations. If we are unlucky, we may make observations that give us a large *p*-value even in cases when this is unlikely. It is therefore misleading to say that their algorithm is "faster" than the statistical model checking algorithm used by Younes et al. (2004), as the latter algorithm is properly designed to realize a certain performance characteristic. Their empirical results can in fact not be replicated reliably because there is no fixed procedure by which they can determine the sample size required to achieve a certain accuracy. 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).

6 Discussion

Sen et al. (2004) were first to consider the problem of CSL verification for "black-box" systems. We have generalized this 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. We have extended the logic PCTL to enable the expression of properties of general stochastic discrete event systems. The algorithm we have presented for verifying PCTL properties of "black-box" systems is an improvement over a similar but flawed algorithm proposed by Sen et al.

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 (2002) and empirically evaluated by Younes et al. (2004). 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 enabled simulation, then we are better off with the alternative approach as it gives full control over the probability of obtaining an incorrect result.

References

Alur, Rajeev, Costas Courcoubetis, and David L. Dill. 1991. Model-checking for probabilistic real-time systems. In *Proceedings ofthe 18th International Colloquium on Automata, Languages and Programming,* edited by J. Leach Albert, B. Monien, and M. Rodriguez Artalejo, vol. 510 of *Lecture Notes in Computer Science,* 115-126, Madrid, Spain. Springer.

Alur, Rajeev and David L. Dill. 1994. A theory of timed automata. *Theoretical Computer Science* 126, no. 2: 183-235.

Aziz, Adnan, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. 2000. Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic* 1, no. 1: 162-170.

Baier, Christel, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. 2003. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering* 29, no. 6: 524-541.

De Moivre, A. 1738. A method of approximating the sum of the terms of the binomial $(a + b)^n$ expanded into a series, from whence are deducted some practical rules to estimate the degree of assent which is to be given to experiments. In *The Doctrine ofChances: or> A Method ofCalculating the Probabilities ofEvents in Play,* 235-243. 2nd ed. London, UK: H. Woodfall.

Doob, J. L. 1942. What is a stochastic process? *The American Mathematical Monthly* 49, no. 10: 648-653.

———. 1953. *Stochastic Processes.* New York, NY: John Wiley & Sons.

Halmos, Paul R. 1950. *Measure Theory.* New York, NY: Van Nostrand Reinhold Company.

Hansson, Hans and Bengt Jonsson. 1994. A logic for reasoning about time and reliability. *Formal Aspects ofComputing* 6, no. 5: 512-535.

Hoel, Paul G., Sidney C. Port, and Charles J. Stone. 1972. *Introduction to Stochastic Processes.* Boston, MA: Houghton Mifflin Company.

Hogg, Robert V. and Allen T. Craig. 1978. *Introduction to Mathematical Statistics.* 4th ed. New York, NY: Macmillan Publishing Co.

Infante López, Gabriel G., Holger Hermanns, and Joost-Pieter Katoen. 2001. Beyond memoryless distributions: Model checking semi-Markov chains. In *Proceedings of the 1st Joint International PAPM-PROBMIV Workshop,* edited by Luca de Alfaro and Stephen Gilmore, vol. 2165 of *Lecture Notes in Computer Science,* 57-70, Aachen, Germany. Springer.

Kwiatkowska, Marta, Gethin Norman, and David Parker. 2002. Probabilistic symbolic model checking with PRISM: A hybrid approach. In *Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems,* edited by Joost-Pieter Katoen and Perdita Stevens, vol. 2280 of *Lecture Notes in Computer Science,* 52-66, Grenoble, France. Springer.

Montgomery, Douglas C. 1991. *Introduction to Statistical Quality Control.* 2nd ed. New York, NY: John Wiley & Sons.

Sen, Koushik, Mahesh Viswanathan, and Gul Agha. 2004. Statistical model checking of black-box probabilistic systems. In *Proceedings of the 16th International Conference on Computer Aided Verification,* edited by Rajeev Alur and Doron A. Peled, vol. 3114 *of Lecture Notes in Computer Science,* 202-215, Boston, MA. Springer.

Younes, Håkan L. S., Marta Kwiatkowska, Gethin Norman, and David Parker. 2004. Numerical vs. statistical probabilistic model checking: An empirical study. In *Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems,* edited by Kurt Jensen and Andreas Podelski, vol. 2988 *of Lecture Notes in Computer Science,* 46-60, Barcelona, Spain. Springer.

Younes, Håkan L. S. and Reid G. Simmons. 2002. Probabilistic verification of discrete event systems using acceptance sampling. In *Proceedings of the 14th International Conference on Computer Aided Verification,* edited by Ed Brinksma and Kim Guldstrand Larsen, vol. 2404 of *Lecture Notes in Computer Science,* 223-235, Copenhagen, Denmark. Springer.