

# A Bounded Statistical Approach for Model Checking of Unbounded Until Properties

Ru He  
Department of Computer  
Science  
Iowa State University  
rhe@cs.iastate.edu

Samik Basu  
Department of Computer  
Science  
Iowa State University  
sbasu@cs.iastate.edu

Arka P. Ghosh  
Department of Statistics  
Iowa State University  
apghosh@iastate.edu

Huaiqing Wu  
Department of Statistics  
Iowa State University  
isuhwu@iastate.edu

## ABSTRACT

We study the problem of statistical model checking of probabilistic systems for PCTL unbounded until property  $P_{\bowtie p}(\varphi_1 \cup \varphi_2)$  (where  $\bowtie \in \{<, \leq, >, \geq\}$ ) using the computation of  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$ . The approach is first proposed by Sen et al. [13] but their approach suffers from two drawbacks. Firstly, the computation of  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  requires for its validity, a user-specified input parameter  $\delta_2$  which the user is unlikely to correctly provide. Secondly, the validity of computation of  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  is limited only to probabilistic models that do not contain loops. We present a new technique which addresses both problems described above. Essentially our technique transforms the hypothesis test for the unbounded until property in the original model into a new equivalent hypothesis test for bounded until property in our modified model. We empirically show the effectiveness of our technique and compare our results with those using the method proposed in [13].

## 1. INTRODUCTION

Statistical model checking techniques ([16, 9, 13]) based on Monte Carlo simulations have been proposed and developed for verifying probabilistic systems (CTMC, DTMC, semi-Markov Chains [3]) where traditional (numerical) verification may become unusable due to state-space explosion problem. The state-space explosion problem is addressed effectively by statistical methods because these methods only require some *succinct* representation [9] of the system being model checked instead of storing its state space. The central theme of statistical model checking is that the verification of properties is based on only a finite set of sample simulation paths of the system and the result is computed within a pre-specified error limit.

One of the main challenges in the effective application of the statistical model checking for probabilistic systems is the verification problem of PCTL/CSL ([7, 1]) until properties of the form  $P_{\bowtie p}(\varphi_1 \cup \varphi_2)$ , where  $\bowtie \in \{\leq, \geq, <, >\}$  and  $p \in [0, 1]$ . This is to verify whether a path starting from a state in the system satisfies  $\varphi_1 \cup \varphi_2$  with probability  $\bowtie p$ . A path satisfies  $\varphi_1 \cup \varphi_2$  if and only if there exists some state  $s_j$  in the path (of any length) that satisfies  $\varphi_2$  and in all the states before  $s_j$  in the path  $\varphi_1$  is satisfied. The challenge stems from the fact that statistical methods rely on pre-specified finite length simulations and this may lead to the existence of some finite simulation where every state satisfies  $\varphi_1$  but no states satisfy  $\varphi_2$ . As a result it cannot be inferred (*decided*) whether  $\varphi_1 \cup \varphi_2$  is satisfied or not along such a simulation path. In this paper, we will also refer to until path property  $\varphi_1 \cup \varphi_2$  as *unbounded* until path property.

To counter this problem, Sen et al. [13] propose a statistical model checking algorithm based on Monte Carlo simulation and hypothesis testing. The key aspect of their approach is the introduction of a basis procedure which is repeatedly used to verify  $P_{\bowtie p}(\varphi_1 \cup \varphi_2)$  within a given error bound. The basis procedure is essentially a specific statistical hypothesis test:  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$ , i.e., whether a state is the root of some path satisfying  $\varphi_1 \cup \varphi_2$ .

It is worth mentioning here that before Sen et al. propose their statistical-version solution, the numerical method for verifying PCTL property  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  [7] has already existed, which is equivalent to verifying  $\neg E(\varphi_1 \cup \varphi_2)$  (non probabilistic CTL property) using traditional (non-probabilistic) model checking method [4]. The underlying reason is because zero or non-zero probabilistic check depends only on the topology of the transitions and not on the actual values of the transition probabilities, which is similar to the fact mentioned by Hart et al. [8] in early 1980's. However, the traditional model checking algorithm for  $\neg E(\varphi_1 \cup \varphi_2)$  just suffers from state-space explosion problem, the main target that all the statistical methods intend to address. Therefore, since the statistical solution for  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  proposed by Sen et al. really avoids the corresponding state-space explosion problem, it has its importance and usefulness.

transformation of the model  $M$  where a terminal state is introduced such that it is reachable from every state in one step with a small *stopping* probability  $p_s$ . This transformation forces every simulation in the modified model to eventually terminate with a decided result, i.e.,  $\varphi_1 \cup \varphi_2$  is satisfied or not satisfied.

An important user-specified parameter  $\delta_2$ , necessary for the validity of the basis procedure as proposed in [13], depends directly on the actual probability of satisfying the given until property  $\varphi_1 \cup \varphi_2$  by any state. As it is impossible for the user to know this probability value a priori, it is also not possible to appropriately choose  $\delta_2$ . Furthermore, as the basis procedure is iteratively applied by the general procedure for verifying the general unbounded until property  $P_{\times p}(\varphi_1 \cup \varphi_2)$ , invalidity of the basis procedure will be accumulated and will result in the invalidity of the verification result for the general unbounded until property as well. Another important and more severe drawback of the basis procedure is that in general it is not valid for models containing loops; Younes and Simmons [17] point out this flaw in the validity of the proof of Theorem 2 in [13] (see Section 2.2 for our discussion).

We acknowledge the usefulness of the statistical method developed by Sen et al. [13] and address the above drawbacks by proposing a new method for solving the basis procedure. The main idea of our solution is to update the transition probabilities of the model  $M$  to obtain  $M'$  instead of introducing a terminal state and changing the model structure as proposed in [13]. For any state  $s$  with an outgoing branching factor  $d_s$ , our update amounts to changing the probability of each outgoing transition to  $1/d_s$ . We prove that the original hypothesis testing applied to verifying  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  at a state  $s$  in  $M$  is exactly equivalent to a modified hypothesis testing in  $M'$ . We establish that testing in  $M'$  requires every simulation to run till a finite bound  $N - 1$ , where  $N$  is the number of states in the model. In essence, we reduce the verification of unbounded until property in  $M$  to a verification problem for bounded ( $\leq N - 1$ ) until property in  $M'$ . We present a procedure for the hypothesis testing within any user-specified error limit. Our method eliminates the requirement of providing  $p_s$  or  $\delta_2$  (as in [13]) and is applicable to models with any structure. Furthermore, like other statistical methods [16, 9, 13], our proposed method only requires a succinct representation of the model  $M$ , does not store any state while exploration, and therefore avoids state-space explosion problem (at the cost of computing results within a pre-specified error limit).

### Contributions.

1. We present a new statistical procedure to check the PCTL unbounded until formula  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$ , which is applicable to any models and can be seamlessly used by the general procedure in [13] for the verification of general property  $P_{\times p}(\varphi_1 \cup \varphi_2)$ .
2. We transform a hypothesis testing problem of unbounded until properties in a model  $M$  into an exactly equivalent hypothesis testing problem in a modified model  $M'$  and ensure that each simulation path length is bounded by the number of states in the model. To the best of our knowledge, none of the existing statis-

tical methods are able to provide such a finite bound for verifying unbounded until property.

3. Our hypothesis testing is based on a modified model  $M'$  of  $M$  where the modification is independent of the transition probabilities of  $M$ . As such, the testing result for  $M'$  is applicable not only for the original model  $M$  but also for all other models (a) with the same transition structure as  $M$  and (b) with the same propositions labeling each state as in  $M$ . This aspect allows for the re-use of the verification result of our basis procedure for the class of *similar* models. Re-use, in turn, reduces the verification effort when a large number of these similar models are being verified against the same probabilistic until property.

**Organization.** Section 2 provides an overview of the preliminary concepts and definitions that will be used in the rest of the paper. Section 3 describes our solution methodology, while Section 4 discusses the theoretical bounds of simulations in our method. Section 5 presents our experimental results and empirically shows the effectiveness of our techniques. Finally, Section 6 discusses the related work and Section 7 discusses some future avenues of research.

## 2. BACKGROUND

For the purpose of explaining and establishing the theoretical results, we will consider the verification of probabilistic transition systems modeled as DTMC against the properties expressed by PCTL. The results are directly extensible to other modeling paradigms (CTMC and semi-Markov chains) and the corresponding property logics (CSL) representing until properties.

### 2.1 Probabilistic Transition Systems and PCTL

**Definition 1** (PROBABILISTIC TRANSITION SYSTEMS). *A probabilistic transition system [9] PTS = (S, s<sub>I</sub>, T, L), where S is a finite set of states, s<sub>I</sub> ∈ S is the initial or start state, T : S × S → [0, 1] is a transition probability function such that  $\forall s : \sum_{s' \in S} T(s, s') = 1$ , and L : S → P(AP) is the labeling function which labels each state with a set of atomic propositions  $\subseteq AP$  which hold in that state.*

**Paths and Probability Measures.** A path in PTS, denoted by  $\pi$ , is a finite or infinite sequence of states  $(s_0, s_1, s_2, s_3, \dots)$  such that for all  $i \geq 0 : s_i \in S$  and  $T(s_i, s_{i+1}) > 0$ . We denote the set of all infinite paths starting from  $s$  as  $Path(s)$ .  $\pi[i]$  denotes the  $i$ -th state in the path  $\pi$  and  $|\pi|$  is the length of  $\pi$  in terms of the number of transitions in  $\pi$ . For example, for an infinite path  $\pi$ ,  $|\pi| = \infty$ , while for a finite path  $\pi = (s_0, \dots, s_n)$ ,  $|\pi| = n, n \geq 0$ . The cylinder set, denoted by  $C_s(\pi)$ , for a state  $s$ , where  $\pi$  is a finite length path starting from  $s$ , is defined as  $C_s(\pi) = \{\pi' : \pi' \in Path(s) \wedge \pi \text{ is prefix of } \pi'\}$ . Essentially,  $C_s(\pi)$  is the set of all infinite paths  $\in Path(s)$  with the common finite length prefix  $\pi$ . For any finite path  $\pi$  with  $|\pi| = n$  we define

$$P(\pi) = \begin{cases} 1 & \text{if } n = 0 \\ T(\pi[0], \pi[1]) \times \dots \times T(\pi[n-1], \pi[n]) & \text{otherwise} \end{cases}$$

For a cylinder  $C_s(\pi)$ , we define  $Pr(C_s(\pi)) = P(\pi)$ . It is well-known that this probability measure  $Pr(\cdot)$  extends uniquely over all sets in the relevant  $\sigma$ -algebra.

**PCTL Syntax and Semantics.** Properties of PTS can be expressed using PCTL, an extension of standard CTL augmented with probabilistic specifications. Let  $\varphi$  represent a state formula and  $\psi$  represent a path formula. Then PCTL syntax is defined as follows:

$$\varphi \rightarrow tt \mid a \in AP \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbb{P}_{\bowtie r}(\psi)$$

$$\psi \rightarrow \varphi \mathbb{U} \varphi \mid \varphi \mathbb{U}^{\leq k} \varphi$$

In the above,  $\bowtie \in \{\leq, \geq, <, >\}$ ,  $r \in [0, 1]$  and  $k \in \{0, 1, \dots\}$ . We always use state formulas ( $\varphi$ ) to specify the properties of a PTS and path formulas ( $\psi$ ) only occur inside  $\mathbb{P}_{\bowtie r}(\cdot)$ . A state  $s$  (or a path  $\pi$ ) satisfying a state formula  $\varphi$  (or a path formula  $\psi$ ) is denoted by  $s \models \varphi$  (or  $\pi \models \psi$ ), and is inductively defined as:

$$\begin{aligned} s \models tt & \quad \text{for all } s \in S \\ s \models a & \quad \Leftrightarrow a \in L(s) \\ s \models \neg\varphi & \quad \Leftrightarrow s \not\models \varphi \\ s \models \varphi_1 \wedge \varphi_2 & \quad \Leftrightarrow s \models \varphi_1 \text{ and } s \models \varphi_2 \\ s \models \mathbb{P}_{\bowtie r}(\psi) & \quad \Leftrightarrow \text{Prob}(s, \psi) \bowtie r \end{aligned}$$

In the above,  $\mathbb{P}(s, \psi) = \text{Pr}(\{\pi \in \text{Path}(s) : \pi \models \psi\})$ . In other words,  $s \models \mathbb{P}_{\bowtie r}(\psi)$  holds if and only if the probability that  $\psi$  is true for an outgoing infinite path from state  $s$  is  $\bowtie r$ . For any infinite path  $\pi$ :

$$\begin{aligned} \pi \models \varphi_1 \mathbb{U}^{\leq k} \varphi_2 & \Leftrightarrow \exists 0 \leq i \leq k : \pi[i] \models \varphi_2 \wedge \forall j < i : \pi[j] \models \varphi_1 \\ \pi \models \varphi_1 \mathbb{U} \varphi_2 & \Leftrightarrow \exists i \geq 0 : \pi[i] \models \varphi_2 \wedge \forall j < i : \pi[j] \models \varphi_1 \end{aligned}$$

We will refer to  $\varphi_1 \mathbb{U}^{\leq k} \varphi_2$  as a bounded until path formula and refer to  $\varphi_1 \mathbb{U} \varphi_2$  as an unbounded until path formula (or an until path formula).

**Succinct Representation of PTS Model.** As in other statistical methods, our method only requires a succinct representation of the model being verified. The succinct representation (see [9]) of a PTS is such that given a state  $s$  in the PTS, the representation allows the generation of the set of its next states  $t$  such that  $T(s, t) > 0$ . Such representation typically has a space requirement in the same scale as the number of variables in the system. Liberatore in [12] explains a succinct representation of a probabilistic system capturing the behavior of flipping  $n$  coins. The DTMC model of this PTS system contains  $O(2^n)$  transitions; while the succinct representation of the same model only considers  $2n$  possible transition *relations* each capturing the event of randomly selecting one of the  $n$  coins and tossing it (while other coins remain unaltered).

## 2.2 Verifying $\mathbb{P}_{\bowtie p}(\varphi_1 \mathbb{U} \varphi_2)$ using $\mathbb{P}_{\leq 0}(\varphi_1 \mathbb{U} \varphi_2)$

We present the method proposed by Sen et al. [13] in Algorithms 1 and 2. Algorithm 1, which captures the main idea of the general procedure, computes the estimate of  $\mathbb{P}(s, \varphi_1 \mathbb{U} \varphi_2)$  by performing  $J$  simulations from state  $s$  and computing the proportion of the simulations that satisfy the given until property (Line 15). If a state in the simulation path does not satisfy  $\varphi_2$ , Algorithm 2 (the basis procedure) is invoked to decide whether the state is the root of any path satisfying  $\varphi_1 \mathbb{U} \varphi_2$  (Line 8). Algorithm 2 executes on a modified model  $M'$  where every state can reach a terminal state (which satisfies  $\neg\varphi_1 \wedge \neg\varphi_2$ ) in one step with a pre-specified probability  $p_s$ ; all the existing probabilities are accordingly modified by multiplying with  $(1 - p_s)$  such that the sum of

---

### Algorithm 1 Compute $\mathbb{P}(s, \varphi_1 \mathbb{U} \varphi_2)$

---

**Require:**  $J$  (the number of simulations starting from  $s$  in  $M$ )

- 1:  $X \leftarrow 0$
- 2: **for** (each simulation  $\pi$  of  $J$  simulations from  $s$  in  $M$ ) **do**
- 3:    $i \leftarrow 0$
- 4:   **while** (true) **do**
- 5:     **if** ( $\pi[i] \models \varphi_2$ ) **then**
- 6:        $X \leftarrow X + 1$
- 7:       **break**
- 8:     **else if** (Decide  $\pi[i] \models \mathbb{P}_{\leq 0}(\varphi_1 \mathbb{U} \varphi_2)$ ) **then**
- 9:       **break**
- 10:    **else**
- 11:      $i \leftarrow i + 1$
- 12:    **end if**
- 13:   **end while**
- 14: **end for**
- 15: **return**  $X/J$

---



---

### Algorithm 2 Decide $t \models \mathbb{P}_{\leq 0}(\varphi_1 \mathbb{U} \varphi_2)$

---

**Require:**  $\alpha$  (Type I error limit),  $\delta_2$

- 1:  $K_0 \leftarrow \log(\alpha)/\log(1 - \delta_2)$
- 2: **for** (each simulation  $\pi'$  of  $K_0$  simulations from  $t$  in  $M'$ ) **do**
- 3:   **if** ( $\pi' \models \varphi_1 \mathbb{U} \varphi_2$ ) **then**
- 4:     **return** false
- 5:   **end if**
- 6: **end for**
- 7: **return** true

---

the probabilities of the outgoing transitions of any state is equal to 1 (see Definition 1). If Algorithm 2 returns false, i.e., there exists a path from  $\pi[i]$  in  $M'$  that satisfies  $\varphi_1 \mathbb{U} \varphi_2$ , then the process is iterated starting from the next state of the simulation path (the index of  $\pi$  increments in Line 11).

There are two problems in Algorithm 2. *First*, the number of simulations  $K_0$  required for Algorithm 2 (Line 1) depends on  $\delta_2$ , a user-specified parameter. We show that the correct pre-specification of  $\delta_2$  is difficult, if not impossible. A pre-condition (condition C2 in [13]) necessary for the validity of the procedure in [13] states that for every state  $s$  in model  $M$ , the probability that a path starting from  $s$  satisfies  $\varphi_1 \mathbb{U} \varphi_2$  must not lie in the range  $(0, \delta_2/(1 - p_s)^N]$ , where  $N$  is the number of states in  $M$ , and  $\delta_2$  and  $p_s$  are user-specified positive parameters. In other words, if the probability of any path satisfying  $\varphi_1 \mathbb{U} \varphi_2$  is  $p$  ( $> 0$ ), then  $\delta_2$  must be selected such that  $p > \delta_2/(1 - p_s)^N$ . As it is unlikely for the user to know  $p$ , i.e.,  $\mathbb{P}(t, \varphi_1 \mathbb{U} \varphi_2)$  ( $t$  being the input state of Algorithm 2), it is not possible to choose a suitable  $\delta_2$  for the computation of the corresponding  $K_0$ . *Second*, Algorithm 2 is not valid in general for models with loops. For the validity of Algorithm 2, Sen et al. prove in their Theorem 2 that for any value of  $p_s$  ( $0 < p_s < 1$ , pre-specified by user)

$$p' \geq (1 - p_s)^N \cdot p \quad (1)$$

where  $N$  is the number of states in the original model  $M$ ;  $p$  and  $p'$  are the probabilities that a path satisfies  $\varphi_1 \mathbb{U} \varphi_2$  from any state  $t$  in the original and modified model respectively.

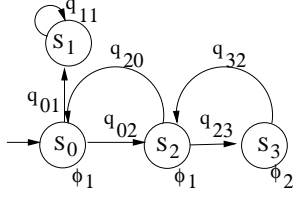


Figure 1: Model showing invalidity of [13]

Here we present a model (Fig. 1) proving the invalidity of the above claim. The states  $s_0$  and  $s_2$  satisfy  $\varphi_1$  and the state  $s_3$  satisfies  $\varphi_2$ ; transitions are annotated with the corresponding probabilities. Let  $p_i = \mathbf{P}(s_i, \varphi_1 \cup \varphi_2)$ . Observe that for Fig. 1,  $p_0 = q_{02} \cdot p_2 + q_{01} \cdot p_1$ ,  $p_1 = 0$ ,  $p_2 = q_{23} \cdot p_3 + q_{20} \cdot p_0$ , and  $p_3 = 1$ . Therefore,  $p_0 = q_{02} \cdot q_{23} / (1 - q_{02} \cdot q_{20})$ . For the modified model (with stopping probability  $p_s$ ),  $p'_0 = q_{02} \cdot (1 - p_s) \cdot p'_2 + q_{01} \cdot (1 - p_s) \cdot p'_1$ ,  $p'_1 = 0$ ,  $p'_2 = q_{23} \cdot (1 - p_s) \cdot p'_3 + q_{20} \cdot (1 - p_s) \cdot p'_0$ , and  $p'_3 = 1$ . Therefore,  $p'_0 = \frac{q_{02} \cdot q_{23} \cdot (1 - p_s)^2}{1 - q_{02} \cdot q_{20} \cdot (1 - p_s)^2}$ . For Equation 1 to hold, i.e., for satisfying  $p'_0 \geq (1 - p_s)^4 \cdot p_0$  for any value of  $p_s$ , the following must hold for any  $p_s$ :  $q_{02} \cdot q_{20} \leq \frac{1}{1 + (1 - p_s)^2}$ . Clearly, this is not true for all  $p_s$  and as such, Theorem 2 in [13] does not hold. In the following section, we will present an alternative for Algorithm 2 which solves the above two problems. That is, our method does not depend on the actual probability of satisfying  $\varphi_1 \cup \varphi_2$ , and is valid for all kinds of models.

### 3. VERIFYING $\mathbf{P}_{\leq 0}(\varphi_1 \cup \varphi_2)$ : ALTERNATE FOR ALGORITHM 2

#### 3.1 Balanced Model

We proceed with the definition of graph-structured equality between PTS models. Informally, two PTSs are said to be graph-structured equal if they differ only in the probabilities ( $\notin \{0, 1\}$ ) associated with each transition.

**Definition 2** (GS-EQUALITY). *For any two PTSs  $M = (S, s_I, T, L)$  and  $M' = (S', s'_I, T', L')$ ,  $M$  is said to be graph-structured equal to  $M'$ , denoted as  $M \equiv_{GS} M'$ , if and only if the following holds:  $S = S'$ ,  $s_I = s'_I$ ,  $L = L'$ ; and for all  $s, t \in S$  or  $S'$ :  $T(s, t) = 0 \Leftrightarrow T'(s, t) = 0$ ,  $T(s, t) = 1 \Leftrightarrow T'(s, t) = 1$ , and  $0 < T(s, t) < 1 \Leftrightarrow 0 < T'(s, t) < 1$ .*

It can be immediately shown that the above relation  $\equiv_{GS}$  is an equivalence relation (reflexive, symmetric and transitive).

##### 3.1.1 GS-Equivalent Transformation.

The central theme of our technique relies on a specific transformation of the original PTS model  $M$  such that the transformed model  $M'$  is GS-equivalent to  $M$ . Given a model  $M = (S, s_I, T, L)$ , the transformed model  $M'$  is a tuple  $(S, s_I, T', L)$  such that the domain of  $T'$  is the same as the domain of  $T$  and  $\forall s, t \in S$ ,  $T(s, t) > 0 \Rightarrow T'(s, t) = 1/d_s$ ; where  $d_s = |\{t : T(s, t) > 0\}|$ , i.e.,  $d_s$  is equal to the number of states reachable from  $s$  in one step with the original transition probability  $> 0$ . This transformation results in  $M'$

which is GS-equivalent to  $M$  (Definition 2). Note that there are infinitely many GS-Equivalent models of  $M$ . We will refer to  $M'$ , generated by the above transformation, as the *balanced* GS-equivalent of  $M$  (see Remark 1 in Subsection 3.2), one where it is equally likely to traverse in all possible directions from any state in  $M'$ . We will show that such a property of the model  $M'$  is ideal for our method.

### 3.2 Derivation of Equivalent Hypothesis Test

Our method for model checking the PCTL property  $\mathbf{P}_{\leq 0}(\varphi_1 \cup \varphi_2)$  in a PTS model  $M$  is based on a hypothesis testing problem (referred to as  $\mathcal{H}$ ) where

$H_0$  (Null Hypothesis) :  $p > 0$  versus

$H_1$  (Alternative Hypothesis) :  $p = 0$

In the above,  $p = \mathbf{P}(s, \varphi_1 \cup \varphi_2) = Pr_s(\{\pi \in Path(s) : \pi \models \varphi_1 \cup \varphi_2\})$ . As noted in Sections 1 and 6, the conventional statistical methods which run Monte Carlo simulations to directly perform the above test  $\mathcal{H}$  can suffer from the drawback that some simulations may not lead to a decided result for any pre-specified simulation path length. Therefore, the results of  $\mathcal{H}$  may not be restricted appropriately within the user-specified error range. In our approach we first obtain a GS-equivalent  $M'$  using the transformation described in Section 3.1.1. Let  $N = |S|$  be the number of the states in  $M$  and  $d_{max}$  be the maximum (or some upper bound) of the out-degrees of all the states in  $M$ .<sup>1</sup> Then we consider a new hypothesis testing problem (referred to as  $\mathcal{H}'$ ) in  $M'$  as follows:

$$H'_0 : p' \geq (1/d_{max})^{N-1} \text{ versus } H'_1 : p' = 0$$

In the above,  $p'$  is defined as the probability that a finite random path of length  $N - 1$  starting from  $s$  satisfies  $\varphi_1 \cup \varphi_2$  in  $M'$ . Note that the length bound  $N - 1$  has been explicitly imposed in our new  $\mathcal{H}'$ . In short,

$$p' = \mathbf{P}(s, \varphi_1 \cup \varphi_2 \leq^{N-1}) \quad (2)$$

**Proposition 1.** *For any path  $\pi$  in  $M$  (or  $M'$ ) satisfying PCTL path formula  $\varphi_1 \cup \varphi_2$ , there exists a corresponding path  $\pi_{simple}$  which satisfies the same formula and  $\pi_{simple}$  does not contain any repetition of states.*

**PROOF.** The proposition holds due to the fact that  $\pi \models \varphi_1 \cup \varphi_2$  iff  $\exists j \geq 0 : \pi[j] \models \varphi_2 \wedge \forall i < j : \pi[i] \models \varphi_1$ , i.e., there exists a finite path  $\pi'$  which is a prefix of  $\pi$  such that  $|\pi'| = j$  and  $\pi' \models \varphi_1 \cup \varphi_2$ . Proceeding further,  $\pi_{simple}$  can always be constructed from  $\pi'$  in the following fashion. For every  $0 \leq k < l \leq j$ , if  $\pi'[k] = \pi'[l]$ , every state between  $\pi'[k + 1]$  and  $\pi'[l]$  (both inclusive) is removed. Note that the path after each removal still satisfies the invariant  $\varphi_1 \cup \varphi_2$  since  $\varphi_1$  and  $\varphi_2$  are both state formulas according to the PCTL syntax. Finally, the path obtained after all the removals results in  $\pi_{simple}$ , the one where there is no repetition of states.  $\square$

**Theorem 1.** *The hypothesis testing problem  $\mathcal{H}$  in  $M$  is equivalent to the hypothesis testing problem  $\mathcal{H}'$  in  $M'$ .*

<sup>1</sup>Note that  $N$  and  $d_{max}$  are the same across  $M$  and  $M'$  since  $M \equiv_{GS} M'$ .

PROOF. (1) To prove:  $H_0$  is true in  $M \Rightarrow H'_0$  is true in  $M'$ .

$$\begin{aligned}
& H_0 \text{ is true in } M \\
\Rightarrow & p > 0 \\
\Rightarrow & Pr(\{\pi \in Path(s) \mid \pi \models \varphi_1 \cup \varphi_2\}) > 0 \\
\Rightarrow & \text{There exists a } \pi \text{ in } M : \pi[0] = s \wedge \pi \models \varphi_1 \cup \varphi_2 \\
\Rightarrow & \pi_{simple} \text{ in } M \wedge \pi_{simple} \models \varphi_1 \cup \varphi_2 \text{ (Proposition 1)} \\
\Rightarrow & \pi_{simple} \text{ in } M' \wedge \pi_{simple} \models \varphi_1 \cup \varphi_2 \text{ (Definition 2)}
\end{aligned}$$

We have established the existence of a simple path  $\pi_{simple}$  that satisfies  $\varphi_1 \cup \varphi_2$  and  $|\pi_{simple}| \leq N - 1$  (since  $\pi_{simple}$  contains no repetition of states). Next, let us denote  $|\pi_{simple}|$  as  $j_0$  and consider the probability of any random path  $\pi$  in  $M'$  of length  $N - 1$ , which has  $\pi_{simple}$  as its prefix.

$$\prod_{i=0}^{j_0-1} T'(\pi[i], \pi[i+1]) = \prod_{i=0}^{j_0-1} (1/d_{\pi[i]}) \geq (1/d_{max})^{N-1} \quad (3)$$

In the above  $d_{\pi[i]}$  denotes the outgoing branching factor of the  $i$ -th state in the path  $\pi$  and  $d_{max}$  is the maximum outgoing branching factor. Next, consider the following events.

$$\begin{aligned}
E_1 &= \{\pi_1 : \pi_1 \text{ is a random path of length } N - 1 \text{ in } M' \\
&\quad \text{which has the prefix } \pi_{simple}\} \\
E_2 &= \{\pi_2 : \pi_2 \text{ is a random path of length } N - 1 \text{ in } M' \\
&\quad \text{which starts at } s \text{ and satisfies } \varphi_1 \cup \varphi_2\}
\end{aligned}$$

From Equation (3),  $Pr(E_1) \geq (1/d_{max})^{N-1}$ . Observe that paths in  $E_1$  and  $E_2$  satisfy  $\varphi_1 \cup \varphi_2$  and paths in  $E_1$  have the additional constraint that they must have  $\pi_{simple}$  as the prefix. Therefore,  $E_1 \subseteq E_2$  and in turn,  $Pr(E_2) \geq Pr(E_1) \geq (1/d_{max})^{N-1}$ . Recall that  $Pr(E_2) = p'$  (Equation 2). Proceeding further,

$$\begin{aligned}
& H_0 \text{ is true in } M \\
\Rightarrow & \pi_{simple} \text{ in } M' \wedge \pi_{simple} \models \varphi_1 \cup \varphi_2 \\
\Rightarrow & p' \geq (1/d_{max})^{N-1} \\
\Rightarrow & H'_0 \text{ is true in } M'
\end{aligned}$$

(2) To prove:  $H'_0$  is true in  $M' \Rightarrow H_0$  is true in  $M$ :

$$\begin{aligned}
& H'_0 \text{ is true} \\
\Rightarrow & p' \geq (1/d_{max})^{N-1} \\
\Rightarrow & \text{There exists a random path } \pi' \text{ in } M' : |\pi'| = N - 1 \wedge \\
&\quad \pi'[0] = s \wedge \pi' \models \varphi_1 \cup \varphi_2 \\
\Rightarrow & \pi' \text{ is in } M \wedge \pi' \models \varphi_1 \cup \varphi_2 \text{ (Definition 2)} \\
\Rightarrow & P(s, \varphi_1 \cup \varphi_2) > 0 \Rightarrow p > 0 \Rightarrow H_0 \text{ is true in } M
\end{aligned}$$

(3) To prove:  $H_1$  is true in  $M \Leftrightarrow H'_1$  is true in  $M'$ .

The proof is straightforward.  $H_1$  is true in  $M$  if and only if  $p = 0$ , which in turn ensures that there exists no path starting from  $s$  that satisfies  $\varphi_1 \cup \varphi_2$ . As  $M \equiv_{GS} M'$  (Definition 2), there exists no path in  $M'$  starting from  $s$  that satisfies  $\varphi_1 \cup \varphi_2$ . Therefore,  $H'_1$  is true in  $M'$ . The reverse can be proved in a similar fashion.  $\square$

There are two important consequences of the validity of Theorem 1. First, it allows us to transform the  $\mathcal{H}$  problem in  $M$

to an equivalent  $\mathcal{H}'$  problem in  $M'$  based on the existence of a simple path  $\pi_{simple}$  in  $M$  and  $M'$ . As  $|\pi_{simple}| \leq N - 1$ , where  $N$  is the number of states in the model, simulation runs of length  $N - 1$  is sufficient to test  $\mathcal{H}'$  and therefore,  $\mathcal{H}$ . Observe that, we do not care whether each simulation path of length at most  $N - 1$  is a simple path or not. In the proof of Theorem 1 we have already shown that the probability that a finite path of length  $N - 1$  from  $s$  satisfies  $\varphi_1 \cup \varphi_2$  in  $M'$  is at least  $(1/d_{max})^{N-1}$ . This addresses the problem of identifying the simulation length bounds for statistically verifying unbounded until properties of the form  $\mathbb{P}_{\leq 0}(\varphi_1 \cup \varphi_2)$ . Second, the family of models which are graph-structured equivalent to  $M$  will be mapped to one  $M'$  using our transformation. The result of testing  $\mathcal{H}'$  for  $M'$  is, therefore, applicable to testing  $\mathcal{H}$  for every member of the family.

**Remark 1.** We choose  $M'$ , the balanced GS-equivalent of  $M$ , because this is an important feature to have for both the correctness and the effective application of our statistical technique. First, for the correctness, only when the original model  $M$  is modified in such a balanced way, the particular real number  $(1/d_{max})^{N-1}$  can be derived for our new hypothesis such that our new hypothesis test is exactly equivalent to the original hypothesis test. In other words, the validity of proof for Theorem 1 relies on our balanced way of modification. Second, our balanced way of modification can safeguard against some worst-case simulations. For example, if the property is true in the model  $M$  with a small probability  $p$  due to certain valuations of transition probabilities, a typical statistical method needs to run many times until the path witnessing the satisfiability is found in the simulations. Our most “balanced” GS-equivalent  $M'$  is able to safeguard against this extreme but important possibility. Of course, it is possible that our method has higher computation cost compared to any method that deals with  $M$  directly (e.g., Sen’s method in [13]) when the true  $p$  in  $M$  is large. But since this value of  $p$  is not known a priori, we introduce our method based on  $M'$  which performs equally well for all possible values of unknown  $p$ .

### 3.3 Testing for the Transformed Model

The hypothesis testing of  $\mathcal{H}'$  in  $M'$  is performed as follows.

Let  $p'$  be the probability of a finite random path  $\pi$  (with  $|\pi| \leq N - 1$  and  $\pi \models \varphi_1 \cup \varphi_2$ ) starting from a given state  $s$ . We require  $K$  simulations of length at most  $N - 1$  starting from state  $s$  in  $M'$ . For each simulation  $i \in \{1, 2, \dots, K\}$ , we define the corresponding binary random variable  $X_i$  as follows: if simulation  $i$  satisfies  $\varphi_1 \cup \varphi_2$  then  $X_i = 1$ ; otherwise (i.e.,  $\varphi_1 \cup \varphi_2$  is either not satisfied or undecided for simulation  $i$  within  $N - 1$  steps),  $X_i = 0$ . Note that each  $X_i$  is independently and identically distributed as a Bernoulli random variable with parameter  $p'$ . If  $\sum_{i=1}^K X_i = 0$ : we reject  $H'_0$  and conclude  $H'_1$  of  $\mathcal{H}'$  (and equivalently we reject  $H_0$  and conclude  $H_1$  of  $\mathcal{H}$ ). On the other hand, if  $\sum_{i=1}^K X_i > 0$ : we conclude  $H'_0$  of  $\mathcal{H}'$  (and equivalently we conclude  $H_0$  of  $\mathcal{H}$ ).

**Stop-Early Strategy.** A common computation saving strategy used in this type of testing is the *stop-early* strategy. While conducting  $K$  simulations, if the  $m$ -th simulation path satisfies  $\varphi_1 \cup \varphi_2$  (i.e.,  $X_m = 1$ ), the process can stop-

early, immediately infer that  $\sum_{i=1}^K X_i > 0$  and conclude  $H'_0$ . (This stop-early saving strategy is also applied to the basis procedure in [13].)

**Samples from succinct representation.** The above method of testing does not require the construction of the original model  $M$  or its transformation  $M'$ . Instead we only require that some succinct representation of the original model  $M$  be available (see Section 2). The simulation runs can be easily obtained as follows. For each simulation path, at each currently visited state  $s$  along the path, we select the next state  $t$  to proceed according to the evenly distributed transition probability  $1/d_s$  instead of the original transition probability  $T(s, t)$ , where  $d_s$  is the number of states reachable from  $s$  in one step. The value of  $d_s$  is directly available from the succinct representation of  $M$ . For instance, for the coin-flipping example discussed in Section 2 ([12]), for any state  $d_s = 2n$ . For the property  $\mathbb{P}_{\leq 0}(\varphi_1 \cup \varphi_2)$ , the simulation of a path is terminated (a) when  $\varphi_2$  is satisfied in the current state (i.e., the path satisfies  $\varphi_1 \cup \varphi_2$ ); (b) when  $\neg\varphi_1 \wedge \neg\varphi_2$  is satisfied in the current state (i.e., the path does not satisfy  $\varphi_1 \cup \varphi_2$ ); or (c) when the length of sample path is equal to  $N - 1$  ( $N$  being the upper-bound of the number of states in  $M$ ). This upper bound can be estimated from the number of variables present in the succinct representation of the model.

**Type I and Type II Errors.** Consider that the user specifies that the Type I Error (the error that  $H_0$  is true but the test incorrectly rejects  $H_0$ ) must be no greater than  $\alpha$ .

$$\begin{aligned} \alpha &= \max \text{Prob}(\text{reject } H'_0 \mid H'_0 \text{ is true}) \\ &= \max \text{Prob}(\sum_{i=1}^K X_i = 0 \mid p' \geq (1/d_{max})^{N-1}) \\ &= (1 - (1/d_{max})^{N-1})^K \end{aligned}$$

Therefore, to control Type I Error within  $\alpha$ , we could set  $K$  to  $\lceil \log(\alpha) / \log[1 - (1/d_{max})^{N-1}] \rceil$ . For controlling the Type II Error (the error that  $H_0$  is false but the test incorrectly accepts  $H_0$ ) within  $\beta$ , similarly we have

$$\begin{aligned} \beta &= \max \text{Prob}(\text{conclude } H'_0 \mid H'_0 \text{ is false}) \\ &= \max \text{Prob}(\sum_{i=1}^K X_i > 0 \mid p' = 0) \\ &= 0 \end{aligned}$$

Therefore, Type II Error of our test is always 0.

In our analysis, we assume that  $\varphi_1 \cup \varphi_2$  does not include any nested probabilistic operators, i.e., the checking of  $\varphi_1$  and  $\varphi_2$  can be done without any error. For the nested probabilistic operators, the mechanism to control the error propagation has already been described in [16] and later used in the checking framework presented in [13]. As our procedure is an alternative basis procedure that can replace the one proposed in [13], the same mechanism to control error propagation can be applied to our method.

## 4. THEORETICAL BOUNDS ON SIMULATION

The product of mean simulation length and sample size (the number of simulations) is a unified measurement of the computation cost of a simulation-based statistical method. We

will compare between the computation costs of the method proposed by Sen et al. [13] (**Method A**) and the method proposed in this paper (**Method B**).

**Bound on the Simulation Length.** As noted before, Method A ensures the finiteness of every simulation because in every simulation, each step has the probability  $p_s$  to end in the terminating state. As a result, the number of steps (i.e., the simulation length) has a geometric distribution with the parameter no less than  $p_s$ . Though it is not possible to provide an upper bound on the simulation length, the average length of simulation can be inferred to be less than or equal to  $1/p_s$ . In [13], the authors suggested that  $p_s$  be set to  $c/N$  where  $c$  is a positive constant and  $N$  is the number of states in the model. Therefore, on average, the length of the simulations for Method A is  $O(N)$ . On the other hand, as per Method B, each simulation may proceed up to  $N - 1$  steps. Therefore, in the worst case, the length of the simulations for Method B is  $O(N)$ .

**Bound on the Number of Simulations.** The worst case of the sample size for any method occurs when  $H_1 : p = 0$  is true (i.e.,  $s \models \mathbb{P}_{\leq 0}(\varphi_1 \cup \varphi_2)$ ). For Method A, Sen et al. [13] state that the required sample size  $K_A = \lceil \log(\alpha) / \log[1 - \delta_2] \rceil$ , where  $\delta_2$  is a user-specified positive parameter (Recall that  $\delta_2$  cannot be chosen correctly due to the invalidity of the Method; see Section 2.2). On the other hand, for Method B, the sample size  $K_B = \lceil \log(\alpha) / \log[1 - (1/d_{max})^{N-1}] \rceil$ .

When  $H_0 : p > 0$  is true (i.e.,  $s \not\models \mathbb{P}_{\leq 0}(\varphi_1 \cup \varphi_2)$ ), both Method A and Method B may still require all the simulation runs. However, for Method B, the probability that the worst-case sample size occurs is close to  $\alpha$ , the tolerance limit for Type I Error. In this situation, we find it important to analyze the expected number of simulations at which point the testing procedure can stop early before the pre-specified (worst-case) number of simulations is exhausted. Let  $\pi_m$  be the first simulation path that satisfies  $\varphi_1 \cup \varphi_2$ , i.e., the testing procedure can be stopped after  $m$  simulation runs ( $m \leq K$ ,  $K \in \{K_A, K_B\}$ ). For Method A,  $m$  has a geometric distribution with parameter  $\theta$ . Therefore, the expectation of  $m$ ,  $E(m) = 1/\theta$ . As claimed in [13],  $\theta$  lies in  $[p(1 - p_s)^N, p]$ , where  $p = \mathbb{P}(s, \varphi_1 \cup \varphi_2)$  in  $M$ . However, due to the invalidity of Method A for general models (see Section 2.2), the lower bound of  $\theta$  cannot be specified as  $p(1 - p_s)^N$ , i.e., the only conclusion that can be made is  $E(m) \geq 1/p$ .

For Method B,  $m$  also has a geometric distribution with parameter  $p'$  so that  $E(m) = 1/p' \leq d_{max}^{N-1}$ , where  $p' = \mathbb{P}(s, \varphi_1 \cup^{\leq N-1} \varphi_2)$  in our modified model  $M'$ . Therefore, when  $p$  is small enough so that  $1/p > d_{max}^{N-1}$ , on average, Method B will obtain a simulation path satisfying  $(\varphi_1 \cup \varphi_2)$  earlier than Method A.

**Complexity of our proposed method.** We have established that our method requires simulation length  $O(N)$  and the sample size in the worst case is  $K_B$ . Therefore the computation cost of our method is bounded by  $O(N K_B)$ .

## 5. CASE STUDY

We validate our theoretical results empirically using a case study of IPv4 zeroconf protocol modeled as PTS (Fig. 2); the

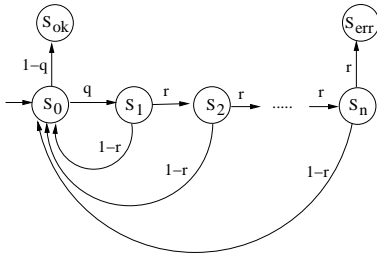


Figure 2: Model of Ipv4 zeroconf protocol.

same model is used in [13]. The model contains  $N = n + 3$  states  $\{s_0, s_1, \dots, s_n, s_{ok}, s_{err}\}$  where  $s_0$  is the start state and the proposition **error** holds only in the state  $s_{err}$ . We will verify whether  $s_0 \models P_{\leq 0}(tt \text{ U error})$ . Observe that, for positive values of  $q$  and  $r$ , the correct inference is  $s_0 \not\models P_{\leq 0}(tt \text{ U error})$ . Furthermore, the real probability  $p$  (i.e.  $P(s_0, tt \text{ U error})$ ) is a function of  $N$ ,  $q$  and  $r$ . For fixed  $q$  and  $r$ ,  $p$  is a decreasing function of  $N$ . For fixed  $N$ ,  $p$  is an increasing function of  $q$  and  $r$ .

**Experimental Setup & Evaluation Parameters.** We use the simulation engine in PRISM [10] to get all the simulation traces and then use our own Java programs<sup>2</sup> to parse all the traces and perform the corresponding analysis. We set the parameter  $p_s$  to be 0.1 needed for Method A across various cases, which is the same as the value used by Sen et al. in their experiments in [13]. For both methods, we keep running the simulations until we obtain one simulation which satisfies  $(tt \text{ U error})$  (recall that in the example case study  $s_0 \not\models P_{\leq 0}(tt \text{ U error})$ ). For Method A, it is equivalent to assuming that a sufficiently small  $\delta_2$  is provided from the user so that its corresponding sample size is large enough to include the first simulation path satisfying  $(tt \text{ U error})$ .

We evaluate the computation cost of the methods with respect to  $m$ , the first simulation that satisfies  $(tt \text{ U error})$ ; and TCS (Total Computation Steps), the total number of transitions of all the required simulations in the test. The quantity  $m$  provides an estimate regarding the number of simulations after which the test procedure can be stopped with a valid result (stop-early), while the quantity TCS provides valuable insights into the computation cost of the test procedure. Note that both quantities are random variables in the test procedure, i.e., they can take different values if the same test is performed multiple times.

**Validity Analysis of Test Procedures.** We set  $n = 6$ ,  $q = 0.3$ , and  $r = 0.3$  in the example model (Fig. 2) and fix the Type I error bound  $\alpha = 1 \times 10^{-4}$ . We evaluate the validity of Method A and Method B. Figure 3 presents the results of our experiments. The histograms present the probability (proportion in 100 runs) of  $m$  values for both methods. We proceed with the discussion on the impact of the selection of  $\delta_2$  for the validity of the results obtained using Method A. Following the suggestion from [13], if we set  $\delta_2 = 0.01$ , then the maximum number of simulations that will be used in Method A is  $K_A = \lceil \log(\alpha) / \log(1 - \delta_2) \rceil = 917$ . In Fig. 3(a), the vertical line marks this value.

<sup>2</sup>Available at <http://www.cs.iastate.edu/~rhe/probupdate/>.

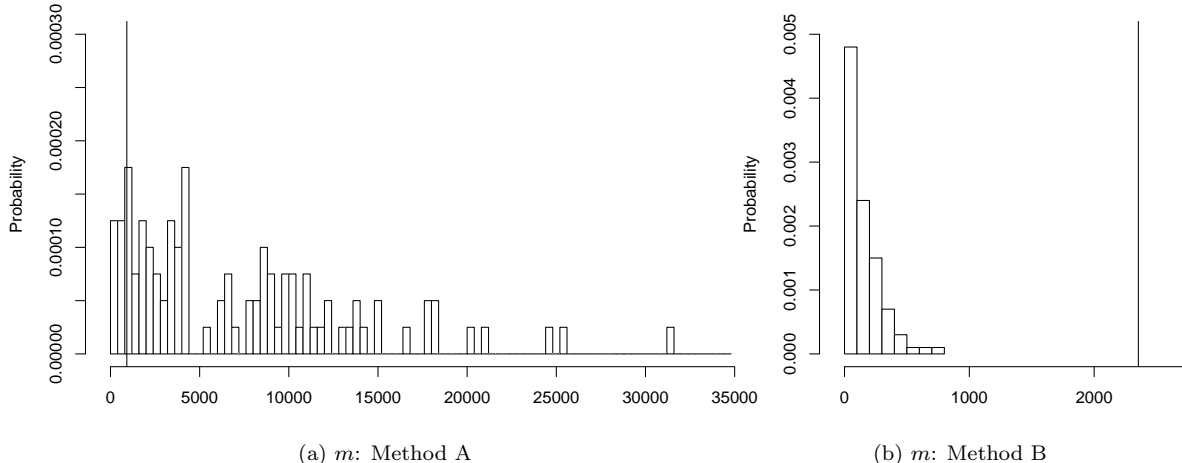
Observe that the valuation of  $m$  (the first simulation that satisfies  $(tt \text{ U error})$ ) is greater than 917 in 89 out of 100 test experiments. Therefore, if  $\delta_2$  (0.01 in this case) provided by the user is not small enough, then insufficient number of simulations (917) will be examined in Method A so that the majority of the test runs would have resulted in the incorrect inference, i.e.,  $p = P(s, tt \text{ U error}) = 0$ .

On the other hand, in our method, Method B, the appropriate number of simulations  $K_B$  is calculated using the pre-specified Type I error limit,  $\alpha$ , the branching factor of the model-states,  $d_{max}$ , and the number of states  $N$ . Here  $N = n + 3 = 9$  and  $d_{max} = 2$  in the model. Therefore,  $K_B = \lceil \log(\alpha) / \log[1 - (1/d_{max})^{N-1}] \rceil \leq 2,354$  (marked by the vertical line in Fig. 3(b)). Observe that, all the 100 test experiments has the valuation of  $m \leq 2,354$ . That is, with the choice of 2,354 simulations, all the experiments with Method B can lead to the valid inference  $P(s_0, tt \text{ U error}) > 0$ . Also note that each  $m$  is far below the worst-case value  $K_B$  so that there is really a lot of computation saving from our *stop-early* strategy.

**Efficiency & Stability Analysis.** In order to compare the efficiency and the stability of Method A and Method B, as before, we conduct experiments with sufficiently large sample size for Method A such that it always includes the first simulation path satisfying  $(tt \text{ U error})$  (i.e., Method A is always valid).

First, we obtain five different test cases by fixing  $q = 0.3$ ,  $r = 0.3$ , and increasing  $n$  from 1 to 10. Table 1(a) shows the values of  $p = P(s_0, tt \text{ U error})$ , and the corresponding results for the sample means and standard deviations (SDs) of  $m$  and TCS for both methods (with 10 runs per each test case). Observe that the sample means and standard deviations of  $m$  and TCS of Method B (our method) are smaller than those of Method A. This experiment empirically shows the *lower average* and the *higher stability* of the computation cost of Method B for the case study. The results can be explained as follows. As discussed in Section 4,  $E(m)$  for Method A is at least  $1/p$ , while  $E(m)$  for Method B is  $1/p' \leq d_{max}^{N-1}$ . With the increase in  $n$  and for the fixed valuations of  $q = 0.3$  and  $r = 0.3$ , the decrease in the valuation of  $p$  in Method A is more than the decrease in the valuation of  $p'$  in Method B. In fact, the valuation of  $p'$  is independent of various valuations of  $q$  and  $r$  (recall that testing based on modified model  $M'$  does not consider  $q$  and  $r$ , Section 3.1.1).

In the second set of experiments (Table 1(b)), we fixed  $n = 5$  and used different  $q$  and  $r$  values between 0.1 and 0.9 (with 10 runs per each test case). The sample means and standard deviations of  $m$  and TCS for Method B remain unchanged. This is because our hypothesis test does not depend on  $q$  and  $r$ ; in fact, one test is sufficient to infer the results for all the models with different valuations of  $q$  and  $r$  ( $\in (0, 1)$ ). On the other hand, the valuation of  $p$  decreases with the decrease in the valuations of  $q$  and  $r$ , and as a result, the means and standard deviations of  $m$  and TCS increase in Method A. Observe that, for large values of  $q$  and/or  $r$  ( $> 0.5$ ), the sample mean of  $m$  ( $\text{mean}(m)$ ) for Method A is smaller than that for Method B as the simulations have higher chance to reach the state  $s_{err}$  (Fig. 2) and satisfy  $(tt \text{ U error})$ . In other words, for models where the probability  $p$  for satisfying a



(a)  $m$ : Method A (b)  $m$ : Method B  
**Figure 3:  $m$  valuations for 100 runs of test procedure.**

desired property decreases dramatically with the transition probability variations, Method A may suffer from the higher mean and variability of computation cost. On the other hand, our method is based on a transformed balanced model (see Remark 1), one where the valuation of  $p'$  is not affected by the transition probabilities of the original model. As such, the efficiency and the stability of the computation in Method B do not depend on the transition probabilities.

## 6. RELATED WORK

### 6.1 Model Checking Probabilistic Systems

Younes and Simmons [16] introduce a statistical method based on the Monte Carlo simulation and sequential hypothesis testing [14] for verifying CSL formulas in CTMC. However, their method excludes the unbounded until properties, and is only applicable to (time) bounded until properties. Techniques for bounded until properties are also discussed in [15].

Herauld et al. [9] also propose a statistical method based on Monte Carlo simulation. The technique uses estimation from Chernoff-Hoeffding inequality [11] to verify a subset of LTL formula and includes the checking of unbounded until properties. However, it fails to completely control the error in the procedure. The reason is as follows. The simulation path length used in the procedure has a pre-specified upper bound. If a simulation reaches that bound and fails to infer a decided result, the technique assumes that the simulation, if allowed to proceed, will eventually have results defending the null hypothesis  $H_0$  of the testing. This assumption allows the procedure to control Type I Error within a pre-specified limit. However, as admitted in [9], the proposed technique cannot determine the appropriate upper bound of simulation path length to control the number of the undecided simulations. As such, the method loses the control of Type II Error. The popular probabilistic model checker PRISM [10] includes a variation of the method described in [9]. The distinguishing feature is that, unlike [9] which allows the undecided simulations, PRISM requires that every simulation terminates with a decided result.

To control both Type I Error and Type II Error of checking unbounded until properties, Sen et al. [13] introduce the new statistical model checking algorithm (discussed in Section 1). However, the technique is invalid in general due to the invalidity of their basis procedure (see Section 2). Zapreev [18] also proposes a statistical technique for verifying unbounded until properties. This technique assumes some prior knowledge of the model structure and requires the users to provide the appropriate parameters such as simulation length.

Unlike the above techniques, we present a statistical model checking technique which is valid for any model structure and also does not require prior knowledge of the probability of the satisfaction of the property under consideration. We establish an upper bound (with respect to the size of the model  $M$ ) on the simulation path length that is sufficient for checking specific unbounded until property of the form  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$ . In essence, we reduce the original hypothesis test for the unbounded until property in a model  $M$  to the equivalent hypothesis test for a bounded until property in a transformed model  $M'$ .

### 6.2 Probabilistic Methods for Model Checking Non-Probabilistic Properties

We have presented a method that can be used as the basis procedure to verify the probabilistic property  $P_{\bowtie p}(\varphi_1 \cup \varphi_2)$  following the methodology proposed by Sen et al. [13]. The basis procedure is essentially verifying  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  or a pure non-probabilistic property  $\neg E(\varphi_1 \cup \varphi_2)$ . As such, any method capable of verifying  $\neg E(\varphi_1 \cup \varphi_2)$  is a candidate alternate for the basis procedure.

In contrast to traditional model checking of  $\neg E(\varphi_1 \cup \varphi_2)$  [4] which has a space complexity polynomial to the number of states in the model, our method, being based on statistical sampling, has a considerable less space requirement (of the order of the size of the succinct representation of the model).

Grosu and Smolka [6] have discussed a statistical-sampling based model checking of non-probabilistic models against LTL properties. Their technique also has a space require-

$n$	$p$	Method A				Method B			
		mean( $m$ )	SD( $m$ )	mean(TCS)	SD(TCS)	mean( $m$ )	SD( $m$ )	mean(TCS)	SD(TCS)
1	0.1139	15.8	8.6	141.1	66.2	1.4	1.0	3.2	2.9
2	0.0371	37.1	33.7	405.8	397.5	8.8	6.0	34.2	24.1
3	0.0114	152.7	153.2	1,517.8	1,508.5	21.3	13.2	105.5	65.9
4	0.0035	439.1	389.7	4,310.8	3,820.2	23.2	22.4	138.2	134.3
5	0.0010	2,086.2	1,564.7	20,817.2	15,624.9	111.3	104.1	778.1	728.9
6	3.12E-04	5,734.8	4,386.8	57,116.2	43,616.5	129.2	150.7	1,032.6	1,205.7
7	9.37E-05	9,084.5	7,032.8	91,157.8	70,111.1	359.3	293.9	3,232.7	2,644.9
8	2.81E-05	65,091.4	55,533.2	651,267.0	555,451.7	414.6	438.2	4,145.0	4,382.5
9	8.44E-06	330,465.6	324,067.8	3,303,374.1	3,241,259.9	926.2	805.2	10,187.2	8,857.5
10	2.53E-06	1,922,723.0	1,789,107.0	19,228,101.0	17,892,495.0	2,490.0	2,011.1	29,879.0	24,133.1

(a)

$(q, r)$	$p$	Method A				Method B			
		mean( $m$ )	SD( $m$ )	mean(TCS)	SD(TCS)	mean( $m$ )	SD( $m$ )	mean(TCS)	SD(TCS)
(0.1, 0.1)	1.11E-6	847,073.4	1,319,903.0	8,470,795.4	13,199,276.0	111.3	104.1	778.1	728.9
(0.25, 0.1)	3.33E-6	739,477.3	888,451.2	7,395,761.8	8,884,764.1				
(0.15, 0.15)	1.34E-5	107,157.3	85,527.6	1,071,463.7	855,301.8				
(0.75, 0.1)	3.00E-5	54,476.7	56,948.4	544,470.2	569,270.5				
(0.2, 0.2)	8.00E-5	23,780.8	20,175.5	237,757.8	202,198.4				
(0.25, 0.25)	3.25E-4	6,819.8	6,207.3	68,424.9	62,713.4				
(0.25, 0.3)	8.09E-4	1,708.2	1,743.5	17,047.5	12,430.1				
(0.3, 0.3)	0.0010	2,086.2	1,564.7	20,817.2	15,624.9				
(0.35, 0.35)	0.0028	617.4	626.6	6,148.4	6,219.7				
(0.4, 0.4)	0.0068	227.5	247.4	2,297.9	2,479.9				
(0.75, 0.3)	0.0072	502.3	384.2	5,032.8	3,930.6				
(0.45, 0.45)	0.0149	182.5	161.9	1,780.3	1,617.1				
(0.5, 0.5)	0.0303	102.3	96.1	1,007.8	923.0				
(0.25, 0.7)	0.0531	28.4	29.0	309.8	342.1				
(0.75, 0.5)	0.0857	30.4	39.7	288.8	386.7				
(0.25, 0.5)	0.1031	83.3	69.4	844.0	733.6				
(0.6, 0.6)	0.1044	23.1	21.7	207.9	229.0				
(0.25, 0.9)	0.1644	12.3	9.8	112.2	78.6				
(0.7, 0.7)	0.2817	7.3	4.9	67.0	37.2				
(0.75, 0.7)	0.3352	6.8	2.9	65.9	31.2				
(0.8, 0.8)	0.5672	3.2	2.7	31.1	29.8				
(0.75, 0.9)	0.6392	2.2	1.7	15.6	13.2				
(0.9, 0.9)	0.8416	3.4	2.4	17.7	12.7				

(b)

**Table 1: Means and Standard Deviations of  $m$  and TCS for Method A and Method B for different valuations of (a)  $n$  and (b)  $(q, r)$ .**

ment linear to the size of the model, which is the same as the one required by the traditional method for model checking  $\neg E(\varphi_1 \cup \varphi_2)$  [4]. I.e., their technique will have the same kind of state-space explosion problem as the traditional model checking method.

Brim et al. [2] propose a probabilistic approach to achieve space reduction in the depth first search (DFS) based LTL model checking of non-probabilistic systems. The work is based on two main observations made in [5]: (a) that it is necessary to cache the states on the search stack for the correctness of DFS based LTL model checking and (b) that caching of states outside the DFS stack such that the overall size of cache is above some threshold (usually between 1/2 and 1/3 of the entire state space) is likely to gain the good space saving without the explosion in time. In this context, authors in [2] use randomness to decide the states that should be cached during DFS. Experiments on two examples (dining philosophers and Peterson algorithm) in [2] show that there is often a good tradeoff point where the space requirement can be reasonably reduced without incurring large penalty in terms of time. For example, experiments on Peterson algorithm show that 33% saving in space can be achieved with a 44% increase in time usage. However, it is difficult to identify a good tradeoff point for any given model in advance. Beyond the good tradeoff point, the in-

attention to further save space may dramatically increase the time cost. For instance, experiments on dining philosophers in [2] reveal that while a saving of 15% in space can be achieved at the cost of 83% increase in time, subsequent gains of 28% and 32% in space increase the time usage by 662% and 6,939% respectively. Finally, as the technique is based on DFS, minimum space that holds the stack is always required, and the maximum stack size depends on the model being explored.

While our technique has a considerably less space requirement compared to the above techniques, it suffers from high computation cost (in terms of the number of simulations and simulation path lengths). However, it should be noted that the worst computation cost corresponds to the case when probability of satisfying  $(\varphi_1 \cup \varphi_2)$  is actually 0. In all other situations, it is likely that the number of simulations ( $m$ ) considered until the first simulation satisfying  $(\varphi_1 \cup \varphi_2)$  is explored is much smaller than the total number of simulations ( $K_B$ ) required to be examined in the worst case (i.e., that our method will be able to stop early). Therefore, our technique is a viable alternate not only for the basis procedure but also for traditional model checking of  $\neg E(\varphi_1 \cup \varphi_2)$ , especially when the model under consideration is large and branching factor from each state is small.

## 7. CONCLUSION

To the best of our knowledge, our work for the first time provides a road-map to obtain the computation complexity of a statistical technique for verifying  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  under the requirement of controlling both Type I Error and Type II Error within the pre-specified limits and under the condition that only succinct representation of the model is available prior to the test. The worst-case computation complexity of our method is  $O(N K_B)$ , where  $K_B$  is of the order  $O(d_{max}^N)$ . We concur with the statement in [9], “Unfortunately, this bound might be exponential in the number of states”, and conjecture that the above computation complexity is the asymptotic lower bound for the complexity of a statistical technique for verifying  $P_{\leq 0}(\varphi_1 \cup \varphi_2)$  under the above requirement and condition.

In future, we will first investigate whether a better bound ( $\ll d_{max}^N$ ) on the number of simulations required for the validity of hypothesis testing can be obtained in the basis procedure by minimally relaxing the conditions. Especially, the objective will be to identify a minimal set of structural information of the system model that may be exploited for this purpose. Second, we intend to obtain a bound on the number of invocations to the basis procedure required for the statistical verification of the general  $P_{\bowtie p}(\varphi_1 \cup \varphi_2)$ .

## 8. REFERENCES

- [1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Verifying continuous time markov chains. In *Proceedings of Computer Aided Verification*, volume 1102, 1996.
- [2] L. Brim, I. Cerna, and M. Necesal. Randomization helps in LTL model checking. In *Proceedings of the Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, pages 105 – 119, London, UK, 2001. Springer-Verlag.
- [3] E. Cinlar. *Introduction to Stochastic Processes*. Prentice Hall, 1975.
- [4] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM TOPLAS*, 8(2), 1986.
- [5] P. Godefroid, G. J. Holzmann, and D. Pirottin. State-space caching revisited. *Formal Methods in System Design*, 7(3):227–241, 1995.
- [6] R. Grosu and S. A. Smolka. Monte carlo methods for process algebra. In *Electronic Notes in Theoretical Computer Science*, volume 162. Springer, 2006.
- [7] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
- [8] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent program. *ACM Trans. Program. Lang. Syst.*, 5(3):356–380, 1983.
- [9] T. Herault, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *5th International Conference on Verification, Model Checking, and Abstract Interpretation*, volume 2937. Springer, 2004.
- [10] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. Prism: A tool for automatic verification of probabilistic systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 3920, 2006.
- [11] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58, 1963.
- [12] P. Liberatore. On polynomial sized MDP succinct policies. *Journal of Artificial Intelligence Research*, 21:551–577, 2004.
- [13] K. Sen, M. Viswanathan, and G. Agha. On statistical model checking of stochastic systems. In *Proceedings of Computer Aided Verification*, volume 3576, 2005.
- [14] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2), 1945.
- [15] H. L. S. Younes. Error control for probabilistic model checking. In *7th International Conference on Verification, Model Checking and Abstract Interpretation*, volume 3855, 2006.
- [16] H. L. S. Younes and R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Proceedings of Computer Aided Verification*, volume 2404. Springer, 2002.
- [17] H. L. S. Younes and R. G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9), 2006.
- [18] I. S. Zapreev. *Model Checking Markov Chains: Techniques and Tools*. PhD thesis, University of Twente, The Netherlands, 2008.