Statistical Model Checking of Black-Box Probabilistic Systems

Koushik Sen, Mahesh Viswanathan, Gul Agha
Department of Computer Science,
University of Illinois at Urbana-Champaign.
`fksen,vmahesh,aghag@uiuc.edu`

Abstract. We propose a new statistical approach to analyzing stochastic systems against speci¯cations given in a sublogic of continuous stochastic logic (CSL). Unlike past numerical and statistical analysis methods, we assume that the system under investigation is an *unknown, deployed black-box*

and software from diverse vendors, making the construction of a formal model of the entire system often impossible and thus limiting the feasibility of numerical and symbolic methods. Second, for large network systems, meaningful experiments may involve dozens or even thousands of routers and hosts, which would mean that the system needs to be deployed before reasonable performance measures can be obtained. However, once they are deployed, such systems cannot be controlled to generate traces from any state, making it impossible to generate execution samples on a need basis as is required by the Younes *et. al*'s statistical information about relevant states. Examples of systems we can successfully analyze are Continuous-time Markov Chains (CTMCs) or systems whose relevant states are discrete, while we are unlikely to succeed for GSMPs in general.

A closely related approach to analyzing stochastic systems based on Monte Carlo simulation is by Herault et. al. [7], which can model-check discrete-time Markov chains against properties expressed in an expressively weak logic (\positive LTL").

We have implemented the whole procedure in Java as a prototype tool, called VeStA (Veri¯cation based on Statistical Analysis).[2] We have experimented with VeStA by applying it to some examples that have been previously analyzed in [15] and the results are encouraging. However, we suspect that VeStA would require a lot more space, because it stores the entire collection of samples it is analyzing. Even though space was not a problem for the examples we tried, we suspect that it may become an issue later.

The rest of the paper is organized as follows. Section 2 de¯nes the class of where s_0 is the unique initial state of the system, s_i is the state of the system after the ith event and t_i is the time spent in state s_i. If the kth state of this sequence is absorbing, then $s_i = s_k$ and $t_i = 1$ for all i ¸ k.

We denote the i

3 Algorithm

In what follows we say that

at s to test the *null hypothesis* $H_0 : p^0 < p$ against the *alternative hypothesis* $H_1 : p^0$ ¸ p. In the second experiment, we test the *null hypothesis* $H_0 : p^0$ ¸ p against the *alternative hypothesis* $H_1 : p^0 < p$.[4]

Let the number of sample execution paths having a state s

go for the second experiment, in which the null hypothesis $H_0: p$

g
<u>if</u> *zsum*

formula $\tilde{A}$ is very close to the threshold p in a formula $P_{./p}(\tilde{A})$ whose satisfaction we are checking at s.

To evaluate the performance and e®ectiveness of our implementation we did plot the results of our experiment. The graph shows that for n closer to 13 the running time and the number of samples required increases considerably to get a respectable p-value of around 10^{-8}. This is because at $n = 13$ the probability that $P_{\geq 0.5}(true\ U \leq T_2$

We could not compare the number of samples for these case studies as they are not available from [15]; theoretically sequential hypothesis testing should require a smaller sample size than simple hypothesis testing to achieve the same level of con¯dence. While in our case studies we never faced a memory problem, we suspect that this may be a problem in very big case studies. We observed

References

1. A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton. Verifying continuous-time Markov chains. In *8th International Conference on Computer Aided Verification (CAV'96)*, volume 1102, pages 269–276. Springer, 1996.
2.