Model Checking of CSL Steady-State Operator with Perfect Simulation*

D. El Rabih, LACL, Université Paris-Est
61, Av. du Général de Gaulle 94010 Créteil

1. Introduction

Stochastic model checking is a recent extension of traditional model checking techniques for the integrated analysis of both quantitative and qualitative system properties. It can be performed by numerical or statistical methods. Numerical model checking provides accurate checking procedure but it is computation intensive. On the other hand, statistical approach [4] is based on hypothesis testing and discrete event simulation, then with this approach we cannot guarantee that the verification result is correct but we can at least bound the probability of generating an incorrect answer. In this paper we propose to apply perfect simulation which is a Monte-Carlo method to study CSL steady-state operator. We illustrate the efficiency of this method when the underlying model is monotone and the state formula ϕ is increasing with respect to the component-wise order. In fact, these hypothesis are not so restrictive and satisfied in general for permance and reliability models.

2. Preliminaries

Perfect Simulation based on coupling from the past is a method to directly generate a sample according to the stationary distribution of Markov chains and avoids the burn-in time period [3]. In this algorithm, we consider the trajectories initiating from all possible initial values. If two sample paths are in the same state at any time t (we say that they couple), they will continue forever during all the simulation. When all the sample-paths have coupled, a sample state is obtained. We must run the simulation from a distant point in the past until the present in order to obtain an exact sample.

Perfect Simulator Ψ^2 proposed in [1] is a sampler designed for the steady state evaluation of various monotone queueing networks. It follows a sampler Ψ of Markov chains developed previously for the perfect sampling of Markov chains without monotonicity properties. In fact, Ψ^2 is a software package which permits to simulate stationary distribution or directly a cost function of large Markov chains by keeping only trajectories issued from the minimal and maximal states.

Continuous Stochastic Logic (CSL) is a powerful mean to state properties which refer to Continuous Time Markov Chains (CTMCs) [2]. It is useful to specify and to evaluate performance and dependability measures as logical formulas over CTMCs. The CSL steady-state operator $\mathcal{S}_{\triangleleft p}(\phi)$ refers to the probability of residing in a particular set of states of the CTMC (specified by a state formula ϕ) in the long run.

3. Checking CSL steady state operator with perfect simulation

The steady-state operator of CSL denoted by $\mathcal{S}_{\triangleleft p}(\phi)$ is related to the steady state probability to be in the set of states of CTMC $\mathcal{M}$ verifying ϕ which is the property to check. The verification of this operator requires the knowledge of the steady state distribution of the CTMC $\mathcal{M}$. We consider that $\mathcal{M}$ is ergodic so the steady state distribution exists and it is independent of the initial distribution. Thus to check this operator, we have to compute the steady-state probability to be in ϕ states, $\Pi^M(S_\phi) = \sum_{s' \models \phi} \Pi^M(s')$ where ϕ is a CSL state formula and s' is a state verifying ϕ. Consequently, $\mathcal{S}_{\triangleleft p}(\phi)$ is satisfied if and only if the expression $\Pi^M(S_\phi) \triangleleft p$ is verified where S_ϕ is the set of states verifying ϕ, p is a probability and $\triangleleft$ is a comparison operator with $\triangleleft \in (<, >, \leq, \geq)$. In other words, to check the steady state operator given by the underlying formula, we have to determine the set of states satisfying ϕ, then sum the probabilities of these states and then compare this sum to the probability p. Therefore we can say that the steady state operator is verified by the CTMC $\mathcal{M}$ if the comparison relation between the determined sum and p is verified.

3.1. Proposed Method

First, we must label states with atomic propositions that will be used to define the underlying property to check, ϕ. For instance, ϕ may represent different performance measures of the underlying model. Then we propose to estimate statistically the underlying stationary distribution Π^M by means of software Ψ^2. As told previously, the checking procedure consists in finding the probabilities for S_ϕ states and compare this sum with p.

In fact, this summation can be seen as a reward

* Joint-work with N. Pekergin. This work is supported by Checkbound,ANR-06-SETI-002

function defined on the state space E where r_ϕ : $E \longrightarrow$ (0,1).

Definition 1. *Let $s = (s_1, s_2, s_3, \cdots, s_n) \in E$ be the state vector where E is the state space of the considered system and n is the number of components in the underlying system. This reward function r_ϕ can be defined as below :*

$$r_\phi(s) = 1, \text{ if } s \models \phi$$
$$r_\phi(s) = 0, \text{ otherwise } s \not\models \phi$$

Let us remark here that the perfect simulation using Ψ^2 can be done when the underlying model (events of this model) is monotone. Moreover to generate reward values according to the steady state distribution, it is sufficient to stop the backward scheme when the trajectories issued from all maximal and minimal states collapse on the same reward function. This may lead to an important reduction of the coupling time. This method called as functional perfect simulation was proposed in [1] and in order to be able to apply it to check a given ϕ, we must prove the monotonicity of reward function r_ϕ. Let us first remind here that a reward function r is said to be monotone if it satisfies $\forall$ (x, y) $\in E^2$, x $\leq$ y $\Rightarrow$ r(x) $\leq$ r(y).

Proposition 1. *For a given ϕ, in order to apply the functional perfect simulation, one must show that $\forall x, y \in E$ such that $x \leq y$, if $r_\phi(x) = 1$ then $r_\phi(y) = 1$*

The following proposition states the checking of the steady-state operator by using a steady-state functional of the underlying Markov chain $\mathcal{M}$.

Proposition 2. *For a given formula ϕ, the steady-state operator $\mathcal{S}_{\triangleleft p}(\phi)$ is checked if the r_ϕ functional of $\mathcal{M}$ in the steady-state fullfils $\triangleleft p$ $(\sum_{s \in E} r_\phi(s).\Pi^{\mathcal{M}}(s) \triangleleft p)$.*

3.2. Model Checking Algorithm

We now present our proposed model checking algorithm. Remind that we consider an ergodic CTMC and it is supposed to be monotone in order to use Ψ^2.

Input : $\mathcal{M}$ (CTMC), ϕ (state formula), p (probability)

Output : Yes, No

1. Testing if the reward function r_ϕ is monotone (by using proposition 2).

2. **If** r_ϕ is monotone **then**

{a)Determine the steady state reward R_ϕ by perfect simulation with functional method using r_ϕ (see proposition 2).

b)If $R_\phi(1) \triangleleft$ p then return 'Yes', else return 'No'.}

Else

{a)Determine the steady state distribution Π^M of the model M by using Ψ^2 with classical method.

b)Find the sum $\sum_{s \models \phi} \Pi^M$ (s)

c)Check if this sum meets the bound p. If this sum $\triangleleft$ p then return 'Yes', else return 'No'.}

4. Case Study

We consider an interconnexion network model (delta network of 32 queue) with a maximum capacity of N_{max} for each queue and an homogeneous input traffic. Note that, in telecommunication networks, multi stage models are used for modelling switches. Let N_i be the number of clients in the i^{th} buffer of this delta network, then the state vector of the underlying model may be $(N_1 N_2 \cdots N_{32})$. This model has been shown to be monotone [1]. Let $a_i(k)$ be the atomic proposition which is true if $N_i > k$ where $0 \leq k \leq N_{max}$ and which is false otherwise. In fact, it is possible to define different interesting performance measures for the underlying system as the saturation probability in the i^{th} buffer, a_i (N_{max}) for example and then we can conclude the loss probability. Let ϕ_1 be the formula to check if at least a queue at third level is saturated. Thus ϕ_1 is defined as follows by means of atomic propositions $a_i(k)$ with k=N_{max}. $\phi_1 = a_{17}(k) \vee a_{18}(k) \vee a_{19}(k) \vee a_{20}(k) \vee a_{21}(k) \vee a_{22}(k) \vee a_{23}(k) \vee a_{24}(k)$. It can be easily shown that the saturation for each queue is monotone and so the union of $a_i(k)$s. We show that it is possible to check this formula with a significant confidence interval in a few seconds while it is not possible to do numerical model checking in this case due to the huge state space size $(O(N_i^{32}))$.

Bibliographie

1. J.-M. Vincent and J. Vienne. PSI2 a Software Tool for the Perfect Simulation of Finite Queueing Networks. In QEST, Edinburgh, September 2007.
2. A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, "Model Checking Continuous Time Markov Chains," *ACM Trans. on Comp. Logic*, 1(1), p. 162-170, 2000.
3. D. Propp, J.and Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. Random Structures and Algorithms, 9(1 and 2), pages 223-252,1996.
4. K. Sen, M. Viswanathan, G. Agha. On Statistical Model Checking of Stochastic Systems. LNCS 3576, 2005.