Stochastic Model Checking*

Marta Kwiatkowska, Gethin Norman, and David Parker

School of Computer Science, University of Birmingham
Edgbaston, Birmingham B15 2TT, United Kingdom

Abstract. This tutorial presents an overview of model checking for both discrete and continuous-time Markov chains (DTMCs and CTMCs). Model checking algorithms are given for verifying DTMCs and CTMCs against specifications written in probabilistic extensions of temporal logic, including quantitative properties with rewards. Example properties include the probability that a fault occurs and the expected number of faults in a given time period. We also describe the practical application of stochastic model checking with the probabilistic model checker PRISM by outlining the main features supported by PRISM and three real-world case studies: a probabilistic security protocol, dynamic power management and a biological pathway.

1 Introduction

Probability is an important component in the design and analysis of software and hardware systems. In distributed algorithms electronic coin tossing is used as a symmetry breaker and as a means to derive efficient algorithms, for example in randomised leader election [38,26], randomised consensus [3,18] and root contention in IEEE 1394 FireWire [37,47]. Traditionally, probability has also been used as a tool to analyse system performance, where typically queueing theory is applied to obtain steady-state probabilities in order to arrive at estimates of measures such as throughput and mean waiting time [30,61]. Probability is also used to model unreliable or unpredictable behaviour, as in e.g. fault-tolerant systems and multi-media protocols, where properties such as frame loss of 1 in every 100 can be described probabilistically.

In this tutorial, we summarise the theory and practice of stochastic model checking. There are a number of probabilistic models, of which we will consider two in detail. The first, discrete-time Markov chains (DTMCs), admit *probabilistic choice*, in the sense that one can specify the probability of making a transition from one state to another. Second, we consider continuous-time Markov chains (CTMCs), frequently used in performance analysis, which model *continuous real time* and probabilistic choice: one can specify the rate of making a transition from one state to another. Probabilistic choice, in this model, arises through *race conditions* when two or more transitions in a state are enabled.

* Partly supported by EPSRC grants EP/D07956X and EP/D076625 and Microsoft Research Cambridge contract MRL 2005-44.

M. Bernardo and J. Hillston (Eds.): SFM 2007, LNCS 4486, pp. 220–270, 2007.

Stochastic model checking is a method for calculating the likelihood of the occurrence of certain events during the execution of a system. Conventional model checkers input a description of a model, represented as a state transition system, and a specification, typically a formula in some temporal logic, and return 'yes' or 'no', indicating whether or not the model satisfies the specification. In common with conventional model checking, stochastic model checking involves reachability analysis of the underlying transition system, but, in addition, it must entail the calculation of the actual likelihoods through appropriate numerical or analytical methods.

The specification language is a *probabilistic* temporal logic, capable of expressing temporal relationships between events and likelihood of events and usually obtained from standard temporal logics by replacing the standard path quantifiers with a probabilistic quantifier. For example, we can express the probability of a fault occurring in a given time period during execution, rather than whether it is possible for such a fault to occur. As a specification language for DTMCs we use the temporal logic called Probabilistic Computation Tree Logic (PTCL) [29], which is based on well-known branching-time Computation Tree Logic (CTL) [20]. In the case of CTMCs, we employ the temporal logic Continuous Stochastic Logic (CSL) developed originally by Aziz et al. [4,5] and since extended by Baier et al. [10], also based on CTL.

Algorithms for stochastic model checking were originally introduced in [62,23,29,5,10], derive from conventional model checking, numerical linear algebra and standard techniques for Markov chains. We describe algorithms for PCTL and CSL and for extensions of these logics to specify reward-based properties, giving suitable examples. This is followed by a description of the PRISM model checker [36,53] which implements these algorithms and the outcome of three case studies that were performed with PRISM.

Outline. We first review a number of preliminary concepts in Section 2. Section 3 introduces DTMCs and PCTL model checking while Section 4 considers CTMCs and CSL model checking. Section 5 gives an overview of the probabilistic model checker PRISM and case studies that use stochastic model checking. Section 6 concludes the tutorial.

2 Preliminaries

In the following, we assume some familiarity with probability and measure theory, see for example [16].

Definition 1. *Let Ω be an arbitrary non-empty set and $\mathcal{F}$ a family of subsets of Ω. We say that $\mathcal{F}$ is a* field *on Ω if:*

1. *the empty set $\varnothing$ is in $\mathcal{F}$;*
2. *whenever A is an element of $\mathcal{F}$, then the complement $\Omega \setminus A$ is in $\mathcal{F}$;*
3. *whenever A and B are elements of $\mathcal{F}$, then $A \cup B$ is in $\mathcal{F}$.*

A field of subsets $\mathcal{F}$ is called a σ-algebra *if it is field which is closed under countable union: whenever $A_i \in \mathcal{F}$ for $i \in \mathbb{N}$, then $\cup_{i \in \mathbb{N}} A_i$ is also in $\mathcal{F}$.*