Probabilistic Model Checking

Marta Kwiatkowska (mzk@cs.bham.ac.uk)
www.cs.bham.ac.uk/ mzk; www.cs.bham.ac.uk/ dxp/prism/
University of Birmingham
School of Computer Science
Office 136, Computer Science Building
Edgbaston, Birmingham B15 2TT, United Kingdom

Abstract.

Probability is widely used in the design and analysis of software and hardware systems: as a means to derive efficient algorithms (e.g. the use of coin flipping and randomness in decision making); as a model for unreliable or unpredictable behavior (e.g. fault-tolerant systems, computer networks); and as a tool to analyse system performance (e.g. the use of steady-state probabilities in the calculation of throughput and mean waiting time). Probabilistic verification (or probabilistic model checking) refers to a range of techniques for calculating the likelihood of the occurrence of certain events during the execution of the system, and can be useful to establish properties such as "shutdown occurs with probability 0.01 or smaller" and "video frame will be delivered within 5ms with probability 0.97 or greater".

In Lecture 1 we will cover fully probabilistic models, known as Discrete Time Markov Chains (DTMCs). DTMCs can be represented as transition probability matrices. The probabilistic temporal logic PCTL will be introduced. It is derived from CTL with the addition of the probabilistic operator that enables one to specify that some path formula holds with a certain probability or greater (and dually, with a certain probability or less). Model checking for PCTL will be explained, which reduces to solving linear equation systems for Until properties.

In Lecture 2 we will introduce a model for representing distributed probabilistic computation, known as Markov Decision Processes (MDPs). An MDP differs from a DTMC in that, during execution, the system can select in each state one of possibly several probability distributions. The logic PCTL can also be defined for MDPs by quantifying over the adversaries. Model checking for PCTL reduces to solving linear programming problems.

Lecture 3 will consider Continuous Time Markov Chains (CTMCs). They do not exhibit non-determinism, but, in contrast to MDPs, can model real-time. The logic CSL can express time-bounded Until properties such as "with probability .98 or greater, the system will reach the target state within 4.5 miliseconds". Such properties can be model checked by reduction to transient probability calculations. By employing uniformisation, appropriately adapted to matrix-by-vector multiplication as opposed to vector-by-matrix, one can obtain O(N) improvement in complexity of the calculation.

Probabilistic models give rise to very large matrices, and so it is important to represent those compactly. In Lecture 4 we will discuss a symbolic representation for probability matrices based on Multi-Terminal Binary Decision Diagrams (MTBDDs) and the resulting model checking algorithms that have been implemented in the PRISM Probabilistic Symbolic Model Checker (www.cs.bham.ac.uk/ dxp/prism/).

Lecture 5 will be concerned with probabilistic timed automata, a model that extends the classical timed automata with probability distributions over the target states (for the discrete variant) and additionally resets of values drawn from

a continuous probability distribution (the continuous variant). Model checking of discrete probabilistic timed automata can be achieved by deriving an MDP over the regions or zones. Algorithms and issues will be discussed, as well as an example of the FireWire root contention protocol.