Abstraction of Probabilistic Systems

Joost-Pieter Katoen[1,2]

[1] RWTH Aachen University, Software Modeling and Verification Group, Germany
[2] University of Twente, Formal Methods and Tools, The Netherlands

1 Introduction

Probabilistic model checking enjoys a rapid increase of interest from different communities. Software tools such as PRISM [13] (with about 4,000 downloads), MRMC [12], and LiQuor [2] support the verification of Markov chains or variants thereof that exhibit nondeterminism. They have been applied to case studies from areas such as randomised distributed algorithms, planning and AI, security, communication protocols, biological process modeling, and quantum computing. Probabilistic model checking engines have been integrated in existing tool chains for widely used formalisms such as stochastic Petri nets [6], Statemate [5], and the stochastic process algebra PEPA [11], and are used for a probabilistic extension of Promela [2].

The typical kind of properties that can be checked is time-bounded reachability properties—"Does the probability to reach a certain set of goal states (by avoiding bad states) within a maximal time span exceed $\frac{1}{2}$?"—and long-run averages—"In equilibrium, does the likelihood to leak confidential information remain below 10^{-4}?" Extensions for cost-based models allow for checking more involved properties that refer to e. g., the expected cumulated cost or the instantaneous cost rate of computations. Intricate combinations of numerical or simulation techniques for Markov chains, optimisation algorithms, and traditional LTL or CTL model-checking algorithms result in simple, yet very efficient verification procedures. Verifying time-bounded reachability properties on models of tens of millions of states usually is a matter of seconds.

Like in the traditional setting, probabilistic model checking suffers from the *state space explosion* problem: the number of states grows exponentially in the number of system components and cardinality of data domains. To combat this problem, various techniques from traditional model checking have been adopted such as binary decision diagrams (multi-terminal BDDs) [10], partial-order reduction [8] and abstract interpretation [14]. We will focus on bisimulation minimisation for fully probabilistic models such as discrete-time and continuous-time Markov chains (DTMCs and CTMCs, for short), and variants thereof with costs. They are an important class of stochastic processes that are widely used in practice to determine system performance and dependability characteristics.

We first study the comparative semantics of branching-time relations for fully probabilistic systems. Strong and weak (bi)simulation relations are covered together with their characterisation in terms of probabilistic and continuous-time variants of CTL, viz. the temporal logics PCTL [9] and CSL[1,3]. PCTL is a

J.-F. Raskin and P.S. Thiagarajan (Eds.): FORMATS 2007, LNCS 4763, pp. 1–3, 2007.

discrete-probabilistic variant of CTL in which existential and universal path quantification have been replaced by a probabilistic path operator. CSL includes in addition means to impose time-bounds on (constrained) reachability problems. For instance, it allows one to stipulate that the probability of reaching a certain set of goal-states within a specified real-valued time bound, provided that all paths to these states obey certain properties, is at least/at most some probability value. The result of this study [4] is an overview of weak and strong (bi)simulations relations, including connections between discrete- and continuous-time relations.

In particular, strong probabilistic bisimulation preserves the validity of PCTL and CSL formulas. It implies ordinary *lumpability*, an aggregation technique for Markov chains that is omnipresent in performance and dependability evaluation since the 1960s. Quotient Markov chains can be obtained in a fully automated way. The time complexity of quotienting is logarithmic in the number of states, and linear in the number of transitions—as for traditional bisimulation minimisation—when using splay trees (a specific kind of balanced tree) for storing blocks [7]. Experimental results show that—as for traditional model checking—enormous state space reductions (up to logarithmic savings) may be obtained. In contrast to traditional model checking, in many cases, the verification time of the original Markov chain exceeds the quotienting time plus the verification time of the bisimulation quotient. This effect is stronger for bisimulations that are tailored to the property to be checked and applies to PCTL as well as CSL model checking.

Finally, we present a more aggressive abstraction technique for DTMCs and CTMCs that uses a three-valued interpretation, i.e., a formula evaluates to either true, false or indefinite. Abstract DTMCs, in fact Markov decision processes (MDPs), are obtained by replacing transition probabilities by intervals where lower and upper bounds act as under- and over-approximation, respectively. For CTMCs, we resort to uniform CTMCs, i.e., CTMCs in which all states have equal residence times and use transition probability intervals. Any CTMC can be efficiently turned into a weak-bisimilar uniform CTMC. Abstraction then amounts to just replace probabilistic transitions by intervals, and model checking can be reduced to determining (constrained) time-bounded reachability probabilities in continuous-time MDPs. This abstraction is conservative for affirmative and negative verification results and allows to perform abstraction on models where bisimulation fails.

Acknowledgement. Thanks to my co-workers on these topics: Christel Baier, Holger Hermanns, David N. Jansen, Tim Kemna, Daniel Klink, Verena Wolf, and Ivan S. Zapreev.

References

1. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model-checking continuous time Markov chains. ACM TOCL 1, 162–170 (2000)
2. Baier, C., Ciesinski, F., Größer, M.: ProbMela and verification of Markov decision processes. Performance Evaluation Review 32, 22–27 (2005)

3. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: Model-checking algorithms for continuous-time Markov chains. IEEE TSE 29, 524–541 (2003)
4. Baier, C., Katoen, J.-P., Hermanns, H., Wolf, V.: Comparative branching-time semantics for Markov chains. Information and Computation 200, 149–214 (2005)
5. Böde, E., Herbstritt, M., Hermanns, H., Johr, S., Peikenkamp, T., Pulungan, R., Wimmer, R., Becker, B.: Compositional performability evaluation for Statemate. In: QEST, pp. 167–178. IEEE CS, Los Alamitos (2006)
6. D'Aprile, D., Donatelli, S., Sproston, J.: CSL model checking for the GreatSPN tool. In: Aykanat, C., Dayar, T., Körpeoğlu, İ. (eds.) ISCIS 2004. LNCS, vol. 3280, pp. 543–553. Springer, Heidelberg (2004)
7. Derisavi, S., Hermanns, H., Sanders, W.H.: Optimal state-space lumping in Markov chains. IPL 87, 309–315 (2003)
8. Groesser, M., Baier, C.: Partial order reduction for Markov decision processes: a survey. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2005. LNCS, vol. 4111, pp. 408–427. Springer, Heidelberg (2006)
9. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing 6, 512–535 (1994)
10. Hermanns, H., Kwiatkowska, M., Norman, G., Parker, D., Siegle, M.: On the use of MTBDDs for performability analysis and verification of stochastic systems. J. of Logic and Alg. Progr. 56, 23–67 (2003)
11. Hillston, J.: A Compositional Approach to Performance Modelling. Cambridge University Press, Cambridge (1996)
12. Katoen, J.-P., Khattri, M., Zapreev, I.S.: A Markov reward model checker. In: QEST, pp. 243–244. IEEE CS, Los Alamitos (2005)
13. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic symbolic model checking with PRISM: a hybrid approach. Int. J. on STTT 6, 128–142 (2004)
14. Monniaux, D.: Abstract interpretation of programs as Markov decision processes. Science of Computer Programming 58, 179–205 (2005)