Model Checking Gigantic Markov Models

Joost-Pieter Katoen

Software Modelling and Verification, RWTH Aachen University, Germany
Formal Methods and Tools, University of Twente, The Netherlands

Probabilistic model checking – the verification of models incorporating random phenomena – has enjoyed a rapid increase of interest. Thanks to the availability of mature tool support and efficient verification algorithms, probabilistic model checking has been successfully applied to case studies from various areas, such as randomized (distributed) algorithms, planning and AI, security, hardware, stochastic scheduling, reliability analysis, and systems biology [9]. In addition, model-checking techniques have been adopted by mainstream model-based performance and dependability tools as effective analysis means. Probabilistic model checking can thus be viewed as a viable alternative and extension to traditional model-based performance analysis [1].

Typical properties that are checked are quantitative reachability objectives, such as: does the probability to reach a certain set of goal states (by avoiding illegal states) exceed $\frac{1}{2}$? Extra constraints can be incorporated as well that e.g., require the goal to be reached within a certain number of transitions, within a certain budget, or within a real-time deadline. For models exhibiting both transition probabilities and non-determinism, maximal and minimal probabilities are considered. Intricate combinations of numerical (or simulation) techniques for Markov chains, optimization algorithms, and traditional CTL or LTL model-checking algorithms result in simple, yet very efficient verification procedures [2, 10]. Verifying time-bounded reachability properties on continuous-time models of tens of millions of states usually is a matter of seconds. Using symbolic representation techniques such as multi-terminal BDDs, much larger systems can be treated efficiently as well. A gentle introduction can be found in [5].

Like in the traditional setting, probabilistic model checking suffers from the curse of dimensionality: the number of states grows exponentially in the number of system components and cardinality of data domains. This hampers the analysis of real-life systems such as biological models involving thousands of molecules [12], and software models of on-board aerospace systems that incorporate probabilistic error models of various system components on top of the "nominal" system behaviour [3].

This talk considers the theory and practice of aggressive abstraction of discrete-time and continuous-time Markov models. Our abstraction technique is based on a partitioning of the concrete state space that is typically much coarser than e.g., bisimulation minimisation. We exploit three-valued abstraction [4] in which a temporal logic formula evaluates to either true, false, or indefinite. In this setting, abstraction is conservative for both positive and negative verification results; in our setting this means that the analysis yields bounds on the desired probability measures. If the verification of the abstract model yields an indefinite answer (dont know), no conclusion on the validity in the concrete model can be drawn. States in abstract Markov models are groups of concrete states and transitions are either equipped with intervals or modeled as non-deterministic choices. The resulting abstraction is shown to preserve a simulation relation: concrete states are simulated by their corresponding abstract ones.

We present the theoretical foundations of aggressive abstraction of Markov models [6] and show how this technique can be applied in a compositional way. This enables the component-wise abstraction of large models [7, 11]. We present two case studies, one from systems biology and one from queueing theory, illustrating the power of this technique. This includes strategies of which states to group, verification times of the abstract models, and the resulting accuracies of the quantitative results. We show that this abstraction technique enables the verification of models larger than 10^{250} states by abstract models of a few hundred thousands states while obtaining results with an accuracy of 10^{-6} [8].

Acknowledgement. This work is funded by the EU FP7-projects SENSATION and MEALS, the STW project ArRangeer, and the Excellence Program of the German Federal Government.

References

1. C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Performance evaluation and model checking join forces. *Commun. ACM*, 53(9):76–85, 2010.
2. C. Baier and J.-P. Katoen. *Principles of Model Checking.* MIT Press, 2008.
3. M.-A. Esteve, J.-P. Katoen, V. Y. Nguyen, B. Postma, and Y. Yushtein. Formal correctness, safety, dependability, and performance analysis of a satellite. In *ICSE*, pages 1022–1031. IEEE, 2012.
4. M. Huth, R. Jagadeesan, and D. A. Schmidt. Modal transition systems: A foundation for three-valued program analysis. In *ESOP*, volume 2028 of *LNCS*, pages 155–169. Springer, 2001.
5. J.-P. Katoen. Model checking meets probability: A gentle introduction. In *Engineering Dependable Software Systems*, volume 34 of *NATO Science for Peace and Security Series - D*, pages 177–205. IOS Press, 2013.
6. J.-P. Katoen, D. Klink, M. Leucker, and V. Wolf. Three-valued abstraction for probabilistic systems. *J. Log. Algebr. Program.*, 81(4):356–389, 2012.
7. J.-P. Katoen, D. Klink, and M. R. Neuhäußer. Compositional abstraction for stochastic systems. In *FORMATS*, volume 5813 of *LNCS*, pages 195–211. Springer, 2009.
8. D. Klink, A. Remke, B. R. Haverkort, and J.-P. Katoen. Time-bounded reachability in tree-structured qbds by abstraction. *Perform. Eval.*, 68(2):105–125, 2011.
9. M. Z. Kwiatkowska. Model checking for probability and time: from theory to practice. In *LICS*, pages 351–. IEEE Computer Society, 2003.
10. M. Z. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *SFM*, volume 4486 of *LNCS*, pages 220–270. Springer, 2007.
11. S. Shoham and O. Grumberg. Compositional verification and 3-valued abstractions join forces. *Inf. Comput.*, 208(2):178–202, 2010.
12. V. Wolf, R. Goel, M. Mateescu, and T. A. Henzinger. Solving the chemical master equation using sliding windows. *BMC Systems Biology*, 4:42, 2010.