Automated Analysis of Probabilistic Infinite-state Systems

Dominik Wojtczak
University of Liverpool, UK
d.wojtczak@liverpool.ac.uk

In this talk we survey the state of the art in the automated analysis of several classes of finitely-presentable infinite-state probabilistic models which combine probabilistic, recursive and in many cases also controlled or more general game behaviour. Unreliability is inherent to almost all physical systems, e.g. telecommunication networks, distributed systems, railway connections. Software systems, designed with robustness and efficiency in mind, often make explicit use of randomness, thus exhibiting probabilistic behaviour. Also, even if a system or a program behaves deterministically, the environment were it operates is in many cases either unknown or very complex. In such situations the best we can hope for is to statistically quantify the behaviour of the environment and analyse the performance of our model in such a probabilistic setting. It is common to model purely probabilistic systems as finite-state Markov Chains or as Markov Decision Processes if the system is both probabilistic and controlled. However, to faithfully represent the behaviour of many naturally occurring systems one really needs to represent them as infinite-state Markov Chains or Markov Decision Processes. This is because finite-state probabilistic models differ significantly from infinite-state ones, e.g. in finite-state systems no *null recurrent* states can occur, i.e. states that are revisited with probability 1, but with infinite expected time before this happens.

The finitely-presentable infinite-state probabilistic models studied in the literature include Multi-Type Branching Processes (see, e.g. [12, 10]), Stochastic Context-Free Grammars (see, e.g. [13, 1]), Quasi-Death-Birth processes (see, e.g. [14, 15]), "random walks with back-buttons" [9], Recursive Markov Chains [8] and probabilistic Pushdown Systems [4]. These models found applications in domains as diverse as population dynamics, nuclear chain reactions, natural language processing, biological sequence analysis, red blood cells formation, queueing theory, modelling web surfing and finally model checking of probabilistic procedural programs. The relationship of the expressive power of these particular models have been extensively studied, e.g. in [7] where a tight connection was established to the the models used in the queueing theory, but the first natural model that subsumes all of the mentioned models was defined in [11].

The fundamental quantitative analysis of these models can be done by constructing a specific system of monotone polynomial equations and finding its least nonnegative fixed point. In order to find this fixed point efficiently we can use, e.g. numerical approximation methods such as Jacobi iteration, Gauss-Seidel iteration or a decomposed variant of the Netwon's method as described in [8]. A detailed analysis of the theoretical performance of the Newton's method when applied in this setting was done in [3, 5]. Here we look instead on how well it performs in practice and describe a tool called PReMo [17] (Probabilistic Recursive Models analyser) which implements highly optimised versions of these approximation algorithms in Java. Using simple tailored-made input languages PReMo is able to accept Recursive Markov Chains and Stochastic Context-Free grammars as an input, while the other models can be easily translated into these ones. PReMo is able to perform all kind of analyses such as computing the termination probability (as described in [8]), the expected termination time (as in [6]) or the standard deviation of that time (as in [2]) among others. PReMo's performance was tested on Stochastic Context-Free Grammars derived from the Penn Treebank corpora in [17], on queuing models in [7] and on many other examples in [16].

Submitted to:
INFINITY 2013

The general domain of finitely-presented infinite-state probabilistic models is a rich and fascinating field of study that is getting more and more attention. Many important questions regarding the computational complexity of their analysis were already addressed and there is a tool that is able to analyse them efficiently. However, many theoretical questions still remain open, and there is still a potential of improving the performance of PReMo, the range of quantitative analyses and input models it supports.

References

[1] Richard Durbin (1998): *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.

[2] J. Esparza, A. Kucera & R. Mayr (2005): *Quantitative analysis of probabilistic pushdown automata: expectations and variances*. In: 20th Annual IEEE Symposium on Logic in Computer Science, 2005. LICS 2005. Proceedings, pp. 117–126, doi:10.1109/LICS.2005.39.

[3] Javier Esparza, Stefan Kiefer & Michael Luttenberger (2010): *Computing the Least Fixed Point of Positive Polynomial Systems*. SIAM Journal on Computing 39(6), pp. 2282–2335, doi:10.1137/090749591.

[4] Javier Esparza, Antonin Kucera & Richard Mayr (2004): *Model Checking Probabilistic Pushdown Automata*. In: Proceedings of the 19th Symposium on Logic in Computer Science, LICS '04, IEEE Computer Society, Washington, DC, USA, pp. 12–21, doi:10.1109/LICS.2004.23.

[5] Kousha Etessami, Alistair Stewart & Mihalis Yannakakis (2012): *Polynomial Time Algorithms for Multi-type Branching Processesand Stochastic Context-free Grammars*. In: Proceedings of the 44th Symposium on Theory of Computing, STOC '12, ACM, New York, NY, USA, pp. 579–588, doi:10.1145/2213977.2214030.

[6] Kousha Etessami, Dominik Wojtczak & Mihalis Yannakakis (2008): *Recursive Stochastic Games with Positive Rewards*. In: Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part I, ICALP '08, Springer-Verlag, Berlin, Heidelberg, pp. 711–723, doi:10.1007/978-3-540-70575-8_58.

[7] Kousha Etessami, Dominik Wojtczak & Mihalis Yannakakis (2010): *Quasi-Birth-Death Processes, Tree-Like QBDs, Probabilistic 1-Counter Automata, and Pushdown Systems*. Perform. Eval. 67(9), pp. 837–857, doi:10.1016/j.peva.2009.12.009.

[8] Kousha Etessami & Mihalis Yannakakis (2009): *Recursive Markov Chains, Stochastic Grammars, and Monotone Systems of Nonlinear Equations*. J. ACM 56(1), pp. 1–66, doi:10.1145/1462153.1462154.

[9] Ronald Fagin, Anna R. Karlin, Jon Kleinberg, Prabhakar Raghavan, Sridhar Rajagopalan, Ronitt Rubinfeld, Madhu Sudan & Andrew Tomkins (2000): *Random Walks with "Back Buttons"*. In: Proceedings of the 32nd Symposium on Theory of Computing, STOC '00, ACM, New York, NY, USA, pp. 484–493, doi:10.1145/335305.335362.

[10] Theodore E. Harris (2002): *The theory of branching processes*. Courier Dover Publications.

[11] Stefan Kiefer & Dominik Wojtczak (2011): *On Probabilistic Parallel Programs with Process Creation and Synchronisation*. In: Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'11, Springer-Verlag, Berlin, Heidelberg, pp. 296–310.

[12] Andrei Nikolaevitch Kolmogorov & B. A. Sevastyanov (1947): *The calculation of final probabilities for branching random processes*. Doklady 56, pp. 783–786.

[13] Christopher D. Manning & Hinrich Schtze (1999): *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

[14] Marcel F. Neuts (1981): *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Dover Publications.

[15] Marcel F. Neuts (1989): *Structured stochastic matrices of M/G/1 type and their applications*. 5, CRC Press.

[16] Dominik Wojtczak (2009): *Recursive probabilistic models: efficient analysis and implementation*. Ph.D. thesis, University of Edinburgh.

[17] Dominik Wojtczak & Kousha Etessami (2007): *PReMo: An Analyzer for Probabilistic Recursive Models*. In: Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'07, Springer-Verlag, Berlin, Heidelberg, pp. 66–71, doi:10.1007/978-3-540-71209-1_7.