Abstraction-Based Petri Net Reachability Checking

Peep Küngas
peep@idi.ntnu.no
Norwegian University of Science and Technology
Department of Computer and Information Science
Knowledge Systems Group

Petri nets and related graph models have been proposed for a wide variety of applications. These models are particularly suitable for representing concurrent hardware and software systems. A fundamental basis for studying the dynamic properties of systems described with Petri nets is reachability property [7].

Unfortunately the complexity of Petri net reachability checking has been proven to be EXPSPACE-hard [6] in the general case. Although several less complex classes of nets have been determined, there still are classes of problems, which can only be presented with "unconstrained" Petri nets. Therefore there is a need for tools and algorithms coping with that EXPSPACE-hard complexity.

We have studied the use of problem abstraction techniques in conjunction with Petri net reachability checking. Abstraction techniques have been used extensively in AI planning (especially in case-based and analogical reasoning), human problem solving and theorem proving.

In [4] it has been shown that while using abstraction it is possible to reduce the expected search space from $O(n)$ to $O(log\ n)$. This improvement makes combinatorial problems tractable. For instance, if n is an exponential function of problem size, then $log\ n$ is just linear.

This result is based on the premise that with abstraction techniques we may cut solution search space from b^d to $kb^{d/k}$, where b is the branching factor of a search, d is the depth of the search space and k is the ratio of abstraction space to base space.

The essential reason why abstraction reduces complexity is that the total complexity is the sum of complexities of the multiple searches, not their product. Thus abstraction may be an helpful tool for reducing the effect of the state space explosion problem associated with Petri net reachability checking.

While combining abstraction with Petri net reachability checking we take advantage of a fact that Petri nets are representable [1] as sets of Linear Logic [2] (LL) sequents. Thus we can reuse an abstraction framework for LL theorem proving [5], which is adaption from a technique used successfully in AI planning [3].

Although abstraction of Petri nets has been already studied for instance in [9, 8], these approaches are based on analysing only structural properties of a net and in some cases also an initial marking has been taken into account.

The methodology we propose, in the other hand, considers structural properties as well as an initial and a final marking. While preparing the net for reachability checking we use the final marking to determine transitions not relevant for the given reachability problem. Thereby it becomes possible to remove some transitions from a net, without changing the reachability result for the given final marking.

The initial marking helps to recognise a lack of reachability solution. Thus, in some cases we can determine whether the final marking is reachable from the initial one even before we start with reachability checking.

Additionally we propose another method for reachability checking with abstraction and present experimental results.

References

[1] U. Engberg, G. Winskel. Completeness Results for Linear Logic on Petri Nets. Annals of Pure and Applied Logic, Vol. 86, pp. 101–135, 1997.

[2] J.-Y. Girard. Linear Logic. Theoretical Computer Science, Vol. 50, pp. 1–102, 1987.

[3] C. A. Knoblock. Automatically Generating Abstractions for Planning. Artificial Intelligence, Vol. 68, pp. 243–302, 1994.

[4] R. E. Korf. Planning as Search: A Quantitative Approach. Artificial Intelligence, Vol. 33, pp. 65–88, 1987.

[5] P. Küngas. Linear Logic Theorem Proving with Abstraction. In Proceedings of 14th European Summer School in Logic, Language and Information, ESSLLI'2002, Trento, Italy, 5–16 August, 2002.

[6] R. J. Lipton. The Reachability Problem Requires Exponential Space. Department of Computer Science, Research Report 62, Yale University, 1976.

[7] T. Murata. Petri Nets: Properties, Analysis and Applications. Proceedings of IEEE, Vol. 77, No. 4, pp. 541–580, 1989.

[8] J.-S. Song, S. Satoh, C. V. Ramamoorthy. The Abstraction of Petri net. In Proceedings of TENCON'87, 1987 IEEE Region 10 Conference "Computers and Communication Technology Toward 2000", August 25–28, Seoul, Korea, pp. 467–471, 1987.

[9] I. Suzuki, T. Murata. A Method for Stepwise Refinement and Abstraction of Petri Nets. Journal of Computer and System Sciences, Vol. 27, pp. 51–76, 1983.