On the Reachability Problem for Dynamic Networks of Concurrent Pushdown Systems

Mohamed Faouzi Atig and Ahmed Bouajjani

LIAFA, CNRS & Univ. Paris Diderot (Paris 7)
{atig,abou}@liafa.jussieu.fr

We consider the problem of checking safety properties for concurrent programs. We assume that programs may have (potentially recursive) procedure calls as well as (unbounded) dynamic creation of parallel threads. Each procedure can have a finite number of local variables, and there is a finite number of global variables that can be accessed by all parallel threads. We assume that these variables range over a finite data domain (e.g., booleans).

We consider concurrent pushdown dynamic networks as a formal model for this class of programs. In fact, sequential programs can naturally be modeled as pushdown systems, and then, concurrent programs can be modeled as networks where each process can behave as a pushdown systems (i.e., it can modify the global store and operate on the stack representing its local context), and additionally it can create new processes in the network. At each point in time, only one process is running (and can act on the global store) and all the others are idle. A scheduling policy is used along the computations to switch the contexts, i.e., to freeze the execution of some process at some point and resume the execution of some idle one. The most liberal scheduling policy is the one which may introduce context switches at any point in time and without any distinction between processes. For this policy, a computation of the program may have an infinite number of context switches, and the number of context switches in each of the potential computations of the program is in general unbounded. It is easy to see that this model is Turing powerful.

Other policies can be defined by imposing various conditions on the occurrences of context switches. These conditions can concern, e.g., the size of the stacks (for instance in asynchronous programs switches can occur only if the stack of the active thread is empty), the classes of the processes (for instance priorities between threads may be considered), the number of allowed context-switches globally, or per thread, or per class of threads, etc. We show that by considering special scheduling policies, it is possible to obtain models for significant classes of programs/applications for which the reachability problem is decidable. The presented results cover recent work published in [ABT08] and [ABQ09] showing that the considered reachability problems can be reduced to reachability/coverability problems in some classes of Petri nets.

O. Bournez and I. Potapov (Eds.): RP 2009, LNCS 5797, pp. 1–2, 2009.
© Springer-Verlag Berlin Heidelberg 2009

References

[ABQ09] Atig, M.F., Bouajjani, A., Qadeer, S.: Context-Bounded Analysis for Concurrent Programs with Dynamic Creation of Threads. In: 15th Intern. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2009). LNCS, vol. 5505, pp. 107–123. Springer, Heidelberg (2009)

[ABT08] Atig, M.F., Bouajjani, A., Touili, T.: Analyzing Asynchronous Programs with Preemption. In: IARCS Ann. Intern. Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2008), Leibniz Intern. Proc. in Informatics, vol. 2. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2008)