Analysis of Systems with Infinite State Space

Javier Esparza
Universität Stuttgart, Germany

Model-checking has focused mostly on the analysis of systems with a possibly large but finite state space. While the finiteness constraint have proved to be very appropriate for hardware, even the simplest software systems may have an infinite state space. The sources of infinity can be roughly classified into three groups:

- Control: (possibly recursive) procedures and process creation require to enrich the notion of state with unbounded stacks or sets.
- Data: the use of any datatype with an infinite range, like integers, reals, lists, queues, etc. potentially leads to an infinite state space.
- Parameterization: distributed algorithms (like algorithms for mutual exclusion, leader election, byzantine agreement, etc.) are often designed to work for an arbitrary number of processes, i.e., a distributed algorithm is in fact an infinite infinite family of systems. Checking that all elements of the family satisfy a property can be reduced to checking that one single infinite-state system satisfies it.

In the lectures I will introduce some of the techniques used to obtain verification algorithms in the infinite-state case. I will consider four classes of systems: timed automata, broadcast protocols, pushdown systems, and FIFO-automata.

The course will be based on the following four papers:

References

1. R. Alur, D.L. Dill. *A Theory of Timed Automata*, Theoretical Computer Science 126(2), pp:183-235, 1994
2. J. Esparza, A. Finkel, R. Mayr. *On the Verification of Broadcast Protocols*, In: Proc. of LICS 1999, 1999
3. J. Esparza, D. Hansel, P. Rossmanith, S. Schwoon. *Efficient Algorithms for Model Checking Pushdown Systems*, In: Proc. of CAV 2000, 2000
4. A. Vardhan, K. Sen, M.Viswanathan, G. Agha. *Actively Learning to Verify Safety for FIFO Automata*, In: Proc. of FSTTCS 2004, 2004

The papers are available online at, for instance, the home pages of Rajeev Alur, Javier Esparza, and Abhay Vardhan. Students are invited to have a look at them and try to understand the models of computation.