Finite countermodels as invariants. A case study in verification of parameterized mutual exclusion protocol

Alexei Lisitsa
alexei@csc.liv.ac.uk

The University of Liverpool

In [5, 6] we proposed a simple but powerful approach to the verification of safety properties of parameterized and infinite state systems. Consider encoding $e : s \mapsto \varphi_s$ of states of a transition system $\mathcal{S} = \langle S, \rightarrow \rangle$ by formulae of first-order predicate logic satisfying the folowing property. The state s' is reachable from s, i.e. $s \rightarrow^* s'$ if and only if $\varphi_{s'}$ is the logical consequence of φ_s, that is $\varphi_s \models \varphi_{s'}$ or $\varphi_s \vdash \varphi_{s'}$.

Under such assumptions establishing reachability amounts to theorem proving, while deciding non-reachability, becomes theorem disproving. To verify a safety property, i.e non-reachability of *unsafe* states, it is sufficient to disprove a formula of the form $\phi \rightarrow \psi$. We proposed in [5, 6] to delegate the latter task to generic finite model finding procedures for first-order predicate logic [3]. We show in [5] that the parallel composition of a complete finite model finder and a complete theorem prover is a decision procedure for safety properties of lossy channel systems [1] under appropriate encoding. Using a finite model finder Mace4 coupled with a theorem prover Prover9 [7] we successfully applied the method to the verification of alternating bit protocol, specified by a lossy channel system; all parameterized cache coherence protocols from [4]; series of coverability and reachability tasks for Petri Nets; parameterized Dining Philosophers Problem (DPP) and to parameterized linear systems (arrays) of finite automata.[1]

When the safety is verified, the method produces a finite countermodel, which is a concise representation of a system *invariant*. We discuss the invariants produced for some of the mentioned examples, focussing on the one case study. This case study is the verification of parameterized mutual exclusion protocol, which was used as a running example in [2]. The protocol is specified as a parameterized system of finite automata arranged in the linear array.

We conclude with a general claim of relative completeness of the proposed method with respect to the verification methods presented in [1, 4, 2]. In the ongoing work we aim to formally support this claim.

References

[1] Parosh Aziz Abdulla, Jonsson B. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91-101, June 15, 1996.

[2] Parosh Aziz Abdulla, Giorgio Delzanno, Noomene Ben Henda, Ahmed Rezine. Monotonic Abstraction: on Efficient Verification of Parameterized Systems. *Int. J. Found. Comput. Sci.* 20(5): 779-801 (2009)

[3] R. Caferra, A. Leitsch, N. Peltier, *Automated Model Building*, Applied Logic Series, 31, Kluwer, 2004.

[4] G. Delzanno. Constraint-based Verification of Parametrized Cache Coherence Protocols. *Formal Methods in System Design*, 23(3):257–301, 2003.

[5] A. Lisitsa Reachability as deducibility, finite countermodels and verification. In preProceedings of AVOCS 2009, Technical Report of Computer Science, Swansea University, CSR-2-2009, pp 241-243. A conference version, 14pp, is submitted

[1] In all cases, except DPP, the working time of Mace4 was within a few seconds, for DPP it was within 3000 seconds, on a laptop of average specification.

[6] A. Lisitsa Verfication via countermodel finding
http://www.csc.liv.ac.uk/~alexei/countermodel/
[7] W. McCune Prover9 and Mace4 http://www.cs.unm.edu/~mccune/mace4/