Abstraction for System Verification

Susanne Graf
VERIMAG, Grenoble, France

Use of appropriate abstraction is a key for successful verification of properties of programs and systems. Solving a general verification problem $M \models \psi$ is of high complexity – which is sometimes called "state explosion". In an abstraction-based approach, we calculate for some appropriate abstraction function α at a reasonable cost an "abstract interpretation" $\alpha(M)$ of a program or system preserving the property ψ, such that checking $\alpha(M) \models \psi$ can be done with a reasonable effort – because both $\alpha(M)$ and ψ are sufficiently "small" or have sufficiently "close structure" so as to make this check feasible. Despite the fact that this method is not complete when α is chosen from a restricted set, this approach proved to be practical for real applications.

Good news is that abstraction is compositional for almost any usual notion of composition (which we denote $\|$). That means, $\|_i \alpha(M_i) \models \psi$ guarantees that $\|_i M_i \models \psi$. But the bad news is that this is not enough for successful verification of large systems: it is generally the case that (1) for sufficiently preservative α, $\|_i \alpha(M_i) \models \psi$ poses still the state explosion problem, (2) whereas if α provides enough complexity reduction, then $\|_i \alpha(M_i) \models \psi$ does not hold.

Several methods for overcoming this problem have been proposed, such as:

- for systems with a very regular structure, such as a set of (almost) identical M_i, it may be possible to define an additional abstraction α' , such that $\alpha'(\|_i \alpha(M_i))$ is simple, yet strong enough for successfully checking $\alpha'(\|_i \alpha(M_i)) \models \psi$.
- decomposition of ψ into "local guarantees" ψ_i such that $\wedge \psi_i \Rightarrow \psi$ and $\alpha(M_i) \models \psi_i$ holds is sometimes successful; it may fail because local information is not sufficient to ensure ψ_i.
- iterative composition and abstraction avoids providing local guarantees $\ldots \alpha_{123}(\alpha_3(M_3) \| \alpha_{12}(\alpha_2(M_2) \| \alpha_1(M_1)))$. It may fail because the complexity of intermediate expressions explodes; the reason is also here that local information is insufficient.
- iterative computation of abstractions M_i^A taking into account increasingly stronger context information may ensure ψ_i; it generally fails when there are strong mutual dependencies amongst components.

We present a general framework of abstraction and show how to use abstractions for reasoning meaningfully about implementations of large composed systems. We also introduce a general contract framework and show that the combination of such top-down design constraints with bottom-up abstractions allows proving stronger properties.

References

1. K. Baukus, S. Bensalem, Y. Lakhnech, K. Stahl. *Abstracting ws1s Systems to Verify Parameterized Networks.* TACAS 2000; LNCS 1785; 2000.

2. T. Ball, S. K. Rajamani. *The SLAM Toolkit.* CAV 2001; LNCS 2102; 2001.

3. P. Cousot, R. Cousot. *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints.* 4th POPL; 1977.

4. L. de Alfaro, T. A. Henzinger. *Interface Automata.* FSE'01; ACM Press; 2001.
5. S. Graf, G. Lüttgen, B. Steffen. *Compositional Minimisation of Finite State Systems using Interface Specifications.* Formal Aspects of Computation, Vol. 8; 1996.
6. S. Graf. *Characterization of a Sequentially Consistent Memory and Verification of a Cache Memory by Abstraction.* Distributed Computing, Vol. 12; 1999.
7. C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, S. Bensalem. *Property Preserving Abstractions for the Verification of Concurrent Systems.* Formal Methods in System Design, Vol. 6, Issue 1; 1995.
8. S. Quinton, S. Graf. *Contract-based Verification of Hierarchical Systems of Components.* SEFM'08; IEEE Computer Society Press; 2008.
9. S. Quinton, S. Graf, R. Passerone. *Contract-based Reasoning for Component-Systems with Complex Interactions.* submitted for publication; 2010.

`http://www-verimag.imag.fr/~graf/?link=Publications`