High Quality Abstractions For The Verification Of Asynchronous Distributed Systems

Advisors : Jérôme Leroux & Grégoire Sutre
Telephone : 05 40 00 35 09 & 05 40 00 33 36
e-mail : {Jerome.Leroux, Gregoire.Sutre}@labri.fr
Team : Formal Methods
Group : Modeling and Verification

Keywords : verification, CEGAR, interpolation, infinite-state systems, symbolic representations, learning, acceleration, abstract interpretation

Scientific context :

Asynchronous distributed systems (such as multi-threaded programs, client/server applications, ...) are notoriously difficult to design correctly. This is due to the complex (and sometimes unforeseen) interactions resulting from asynchronous communications. Therefore, there is a growing need for automatic verification tools capable of analyzing such systems. This need is amplified with the current development of multicore architectures.

The verification of safety properties reduces to the effective computation of inductive invariants. Abstract interpretation [3] provides a general framework for computing these invariants. Basically, in this approach a concrete system is abstracted away into an abstract one that can be automatically verified. If the abstract system is correct, then by construction, the concrete system is known to be correct. Since the converse is not true, when the abstract system is incorrect, the abstraction must be refined.

The CEGAR approach, for Counter Example Guided Abstract Refinement [4], provides a general way for improving too coarse abstractions. Basically, the CEGAR approach relies on a finite set of predicates that are computed inductively. New predicates are needed when the abstraction provided by the set of predicates computed so far is not precise enough.

Abstract interpretation and CEGAR both compute invariants but with different priorities. The former puts the emphasis on the efficiency at the cost of precision of the analysis, while the later cares about the quality of the analysis. In practice, the convergence of the CEGAR approach relies deeply on the quality of the abstraction.

The computation of high quality abstractions is a challenging problem. Recently, the Craig interpolation technique has provided promising results for the verification of sequential programs. Intuitively, from a proof witnessing the impreciseness of an abstraction, new predicates are computed. Extending this technique to asynchronous distributed systems is a challenging problem. State-of-the-art model-checkers (e.g., CSeq) for mutithreaded programs first apply a sequentialization up to a fixed number of context switches [2], and then apply classical verification techniques on the resulting sequential program.

Goals :

This proposal aims at providing new techniques for computing high quality abstractions for the analysis of asynchronous distributed systems based on various frameworks : program transformation, assume-guarantee, interpolation abstraction, abstact acceleration, learning [1,2,5,6].

References :

[1] Chen, Y.F., Farzan, A., Clarke, E.M., Tsay, Y.K., Wang, B.Y.: Learning minimal separating DFA's for compositional verification. In: Kowalewski, S., Philippou, A. (eds.) TACAS 2009. LNCS, vol. 5505, pp. 31–45. Springer, Heidelberg (2009)

[2] Akash Lal, Thomas W. Reps: Reducing concurrent analysis under a context bound to sequential analysis. Formal Methods in System Design 35(1): 73-97 (2009)

[3] Patrick Cousot, Radhia Cousot: Systematic Design of Program Analysis Frameworks. POPL 1979: 269-282

[4] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, Helmut Veith: Counterexample-guided abstraction refinement for symbolic model checking. J. ACM 50(5): 752-794 (2003)

[5] Philipp Rümmer, Hossein Hojjat, Viktor Kuncak: Disjunctive Interpolants for Horn-Clause Verification. CAV 2013: 347-363

[6] Jérôme Leroux, Grégoire Sutre: Accelerated Data-Flow Analysis. SAS 2007: 184-199