Information Societies Technology (IST) Programme

Future and Emerging Technologies - Open Domain

Project acronym: Advance

Project full title:

Advanced Validation Techniques
for Telecommunication Protocols

Contract no: IST-1999-29082

Deliverable 2

Methods and Algorithms for Parametric Reasoning

Year 1 : October 2000 - September 2001

Date of preparation: October 2001

Contributors: Paris, Edinburgh, Uppsala, Grenoble, Weizmann Inst.

1 Introduction

Parametrized systems arise naturally in the modeling of protocols. For instance, the specification of a protocol may be parametrized by the number of components which may participate in a given session of the protocol. In such a case, it is interesting to verify the correctness of the protocol, regardless of the number of participants in a particular session. Another source of parameterization, is that the description of the protocol may contain variables whose values may change in different runs of the protocol, such as the sizes of messages, length of time-outs and delays, number of message retransmissions, etc.

The problem of verification of general classes of parametrized systems is undecidable[1]. Therefore, this task has mainly been devoted to invent methods for enhancing the performance of symbolic reachability analysis in order to achieve termination more often in practical applications.

2 Regular Model Checking

The *Regular model checking* framework was advocated by the paper [KMM+01] as a uniform framework for the analysis of parametrized systems. The idea of regular model checking is to perform symbolic reachability analysis, using regular sets as a symbolic representation of the state space of the parametrized system. Examples of parametrized systems which can be analyzed in this manner are systems consisting of homogeneous finite-state processes connected in a linear of ring-formed topology.

A major problem in the use of regular model checking is that standard reachability analysis algorithms are in general not guaranteed to terminate. Several papers in the project are therefore devoted to developing *acceleration procedures* for making the algorithm converge more often in practical applications.

One such an approach [Tou01] is based on *widening*, in the spirit of the techniques used in abstract interpretation [2]. The paper studies widening techniques for regular languages and uses this for accelerating fixpoint computations in regular model checking.

Another approach reported in [DLS01] defines an equivalence relation on states and then collapses equivalent states into order to enhance the performance of the reachability algorithm. The equivalence is based on a novel behavioral equivalence on states based on *past* and *future* bisimulations.

The paper [Mai01] adopts a slightly different approach. More precisely, it characterizes a subclass of parameterized systems for which the reachability algorithm is always guaranteed to terminate. The subclass is sufficiently wide to cover existing results for token rings and broadcast protocols.

The paper [BMT01] considers regular model checking using a subclass of regular languages, called *Alphabetic Pattern Constraints (APC)*. In contrast to regular languages, APCs are closed under *permutation rewriting*, which is a common operation in the description of many classes of parametrized systems. Computing closures under permutations can be seen in this setting as applying meta-transitions in order to accelerate the calculation of the whole reachability set of a system.

[1] See K. Apt and D. Kozen, Limits for Automatic Verification of Finite-State Concurrent Systems, *Information Processing Letters*, vol. 22, 1986.

[2] See P. Cousot and R. Cousot, Abstract Interpretation: A Unified Model for Static Analysis of Programs by Construction or Approximation of Fixpoints, in POPL'77.

3 Invariant Generation

The papers [PRZ01] and [APR+01] consider verification of parametrized systems through invariant (induction) based proofs. A major challenge here is to invent induction hypotheses which are sufficiently powerful to verify the given property. The papers describe a method where finite state model checking is used both deriving candidates induction assertions, and to prove that the candidate assertions are indeed inductive. Both steps are performed automatically without the use of theorem provers. The efficiency of the method is demonstrated through verification of several types of parametrized systems such as mutual exclusion protocols and cache coherence protocols.

4 Liveness Properties

Symbolic reachability analysis often generate finite state abstractions which may be used for checking safety properties. However, such abstractions can not be used for checking liveness (progress) properties, since fairness conditions are often not preserved, making almost all liveness properties trivially false in the concrete model. The papers [BLS01] and [ABLS01] address this problem in the context of parametrized systems. The papers present nontrivial techniques for automatic synthesis of fairness conditions in the abstract model from those in the concrete model. These techniques are applied for verification of mutual exclusion protocols such as Szymanski's protocol [BLS01], and counter automata with parametric guards [ABLS01].

5 Non-regular languages

Although regular model checking is a powerful technique, there are several classes of parameterized systems whose behaviors cannot be captured using regular languages. The paper [FP01] considers symbolic model checking using classes of languages more expressive than the regular languages and provides model checking of non-regular parameterized systems (using classes of context-free languages)

Moreover, in many cases, the processes in a parametrized network are infinite-state because of unbounded or parametrized data structures.

The paper [AJ01] considers *timed networks*, a class of parametrized systems, where the individual components operate on real-valued clocks (and hence not finite-state). The model checking algorithm presented in the paper is used to verify a parametrized version of the well-known Fischer's protocol.

In [BSBL01], a method based on combining a deductive approach with abstraction techniques and finite-state model-checking is presented in order to deal with a class of multi-dimensional parametrized systems. This method is shown to be applicable to the verification of systems such as group membership or distributed shared memory consistency protocols where each processes in the network maintains a structure ranging over the set of participating processes.

6 Automata with parameters

As mentioned above, one important source of parameterization is the existence of parameter variables in the description of the program. The paper [BHM01] extends the model of context-free processes by an integer parameter, and presents a model checking procedure for such systems. The procedure is used for data flow analysis of recursive programs. The paper [ABLS01] studies model checking for counter automata extended with parametric guards supplied with fairness conditions (see the above paragraph on liveness properties).

7 Implementation Effort

A tool, called PAX has been implemented based on the work in [BLS01]. The tool has been used to verify several parametrized protocols, e.g. Szymanski's mutual exclusion algorithm

The paper [CAS01] describes the tool TREX which implements symbolic reachability analysis techniques for different classes of infinite-state systems. The tool is used for automatic synthesis of invariants in a parametrized version of the IEEE 1394 root contention protocol.

References

[ABLS01] A. Annichini, A. Bouajjani, Y. Lakhnech, and M. Sighireanu. Analysis of Fair Extended Automata. In *Proc. 8th Intern. Static Analysis Symposium (SAS'01)*, Paris, France, July 2001. Lecture Notes in Computer Science 2126, Springer-Verlag.

[AJ01] Parosh Aziz Abdulla and Bengt Jonsson. Model checking of systems with many identical timed processes, 2001. To appear in the journal of Theoretical Computer Science.

[APR+01] T. Arons, A. Pnueli, S. Ruah, J. Xu, and L. Zuck. Parameterized verification with automatically computed inductive assertions. In *CAV'01*, pages 221–234, 2001.

[BHM01] A. Bouajjani, P. Habermehl, and R. Mayr. Automatic Verification of Recursive Procedures with one Integer Parameter. In *Proc. 26th Intern. Symp. on Mathematical Foundations of Computer Science (MFCS 2001)*, Marianske Lazne, Czech Republic, August 2001. Lecture Notes in Computer Science, Springer-Verlag.

[BLS01] K. Baukus, Y. Lakhnech, and K. Stahl. Verification of parameterized networks. *Journal of Universal Computer Science*, 7(2), 2001.

[BMT01] A. Bouajjani, A. Muscholl, and T. Touili. Permutation Rewriting and Algorithmic Verification. In *Proc. 16th Symp. on Logic in Computer Science (LICS'01)*, Boston (MA), USA, 2001. IEEE.

[BSBL01] K. Baukus, K. Stahl, S. Bensalem, and Y. Lakhnech. Networks of processes with parameterized state space. *Electonic Notes of Theoretical Computer Science*, 50(4), 2001. Proc. Workshop on Verification of Parametrized Systems (VEPAS'01), Crete, July, 2001.

[CAS01] A Collomb-Annichini and M. Sighireanu. Parameterized Reachability Analysis of the IEEE 1394 Root Contention Protocol using TReX. In *Proc. Workshop on Real-Time Tools (RT-TOOLS'01)*, Aalborg, August 2001. Aalborg University.

[DLS01] D. Dams, Y. Lakhnech, and M. Steffen. Iterating transducers. In G. Berry, H. Comon, and A. Finkel, editors, *Computer Aided Verification*, volume 2102 of *LNCS*, 2001.

[FP01] Dana Fisman and Amir Pnueli. Beyond regular model checking. In *Proc. 21th Conference on the Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, December 2001. To appear.

[KMM+01] Y. Kesten, O. Maler, M. Marcus, A. Pnueli, and E. Shahar. Symbolic model checking with rich assertional languages. *Theoretical Computer Science*, 256:93–112, 2001.

[Mai01] M. Maidl. A unifying model checking approach for safety properties of parameterized systems. In *CAV'01*, pages 324–336, 2001.

[PRZ01] A. Pnueli, S. Ruah, and L. Zuck. Automatic deductive verification with invisible invariants. In *TACAS'01*, volume 2031, pages 82–97, 2001.

[Tou01] T. Touili. Regular Model Checking using Widening Techniques. *Electronic Notes in Theoretical Computer Science*, 50(4), 2001. Proc. Workshop on Verification of Parametrized Systems (VEPAS'01), Crete, July, 2001.