Bisimulator 2.0: An On-the-Fly Equivalence Checker based on Boolean Equation Systems

Radu Mateescu and Emilie Oudot*
INRIA / VASY project-team
Faculté des Sciences Mirande, bât. LE2I, F-21000 Dijon, France
{Radu.Mateescu,Emilie.Oudot}@inria.fr

Abstract

*Equivalence checking is a classical verification method determining if a finite-state concurrent system (*protocol*) satisfies its desired external behaviour (*service*) by comparing their underlying labeled transition systems (LTSs) modulo an appropriate equivalence relation. Local (or* on-the-fly*) equivalence checking explores the synchronous product of the LTSs incrementally, allowing an efficient detection of errors in complex systems. In this paper, we consider the technique based on translating the equivalence checking problem in terms of the local resolution of a boolean equation system (BES). We propose two enhancements of this technique in the case of equivalent LTSs: a new, faster BES encoding of weak equivalence relations, and a new local BES resolution algorithm with a good average complexity. These enhancements were incorporated into the BISIMULATOR 2.0 equivalence checker of the CADP toolbox, and led to significant performance improvements.*

1. Context

CADP (*Construction and Analysis of Distributed Processes*) [5] is a state-of-the-art verification toolbox for asynchronous concurrent systems. It accepts as input process algebraic descriptions in LOTOS or CHP, as well as networks of communicating automata. These descriptions are translated into LTSs, represented either implicitly (by their "successor function") as C programs using the OPEN/CÆSAR environment [4], or explicitly (by their list of transitions) as compact binary files encoded in the BCG (*Binary Coded Graph*) format. The toolbox provides a wide range of functionalities: compilation and rapid prototyping, interactive and guided simulation, random execution, model checking and equivalence checking, test generation, and performance evaluation. CADP was used to validate more than 100 case-studies (see http://www.inrialpes.fr/vasy/cadp/case-studies).

We present here two recent enhancements of the BISIMULATOR [9] equivalence checker of CADP, which compares two LTSs modulo various equivalence relations (strong, branching, weak, $\tau^*.a$, safety, trace, and weak trace). The tool follows the *on-the-fly* approach [1], exploring the synchronous product of two implicit LTSs incrementally and searching for mismatches indicating the non equivalence of their initial states. The alternative *global* approach [2] computes the equivalence classes of states in two explicit LTSs using partition refinement and then checks whether their initial states fall into the same class. The global approach is more effective when the LTSs are equivalent, whereas the on-the-fly approach is more suitable for showing non equivalence by quickly detecting counterexamples. Our objective is to improve on-the-fly equivalence checking for equivalent LTSs.

Existing techniques for on-the-fly equivalence checking include synchronous product exploration [3, 1], model checking of characteristic formulas [6], and HORNSAT resolution [11]. BISIMULATOR works by translating the on-the-fly equivalence checking problem to the local resolution of a boolean equation system (BES), which is carried out using specific algorithms [7, 9].

2. Two Enhancements

New BES encodings of equivalence relations. Weak equivalences can be encoded as maximal fixed point BESs by directly translating their mathematical definitions (see the table below, first row, for the BES of branching bisimulation [1]). Solving such BESs requires to compute transitive reflexive closures over internal steps (τ-closures) on both LTSs, which reveals to be the most time-consuming part of the verification process. The new encodings that we propose (see the table below, second row) shift the τ-closure

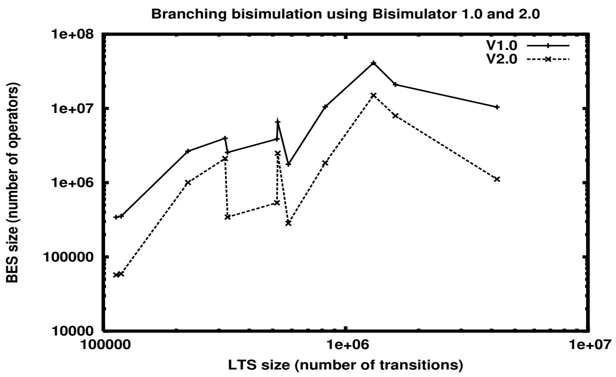
*Research funded by the EC-MOAN project no. 043235 of the FP6-NEST-PATH-COM European program.

computations into the boolean equations, which is about one order of magnitude faster than general τ-closure algorithms [8]. These encodings work only in the absence of τ-cycles, which are eliminated by applying τ-compression [8] on-the-fly on both LTSs simultaneously with their forward exploration underlying the local resolution of the BES.

$$
\begin{aligned}
X_{pq} &\stackrel{\nu}{=} \textstyle\bigwedge_{p \xrightarrow{a} p'} ((a = \tau \wedge X_{p'q}) \vee \\
&\qquad \textstyle\bigvee_{q \xrightarrow{\tau^*} q' \xrightarrow{a} q''} (X_{pq'} \wedge X_{p'q''})) \wedge \\
&\quad \textstyle\bigwedge_{q \xrightarrow{a} q'} ((a = \tau \wedge X_{pq'}) \vee \\
&\qquad \textstyle\bigvee_{p \xrightarrow{\tau^*} p' \xrightarrow{a} p''} (X_{p'q} \wedge X_{p''q'}))
\end{aligned}
$$

$$
\begin{aligned}
X_{pq} &\stackrel{\nu}{=} \textstyle\bigwedge_{p \xrightarrow{a} p'} Y_{pp'qa} \wedge \bigwedge_{q \xrightarrow{a} q'} Z_{pqq'a} \\
Y_{pp'qa} &\stackrel{\nu}{=} (a = \tau \wedge X_{p'q}) \vee U_{pp'qa} \\
Z_{pqq'a} &\stackrel{\nu}{=} (a = \tau \wedge X_{pq'}) \vee V_{pqq'a} \\
U_{pp'qa} &\stackrel{\nu}{=} \textstyle\bigvee_{q \xrightarrow{a} q'} W_{pp'qq'} \vee \bigvee_{q \xrightarrow{\tau} q'} U_{pp'q'a} \\
V_{pqq'a} &\stackrel{\nu}{=} \textstyle\bigvee_{p \xrightarrow{a} p'} W_{pp'qq'} \vee \bigvee_{p \xrightarrow{\tau} p'} V_{p'qq'a} \\
W_{pp'qq'} &\stackrel{\nu}{=} X_{pq} \wedge X_{p'q'}
\end{aligned}
$$

Local BES resolution based on suspend/resume DFS. We propose a new local BES resolution algorithm, which exhibits a smaller average complexity than previously published algorithms (see [7] for a survey). Our algorithm is based on a suspend/resume depth first search (sr-DFS) of the boolean graph representing the dependencies between boolean variables, and stops as soon as the BES portion explored contains a single example or counterexample for the boolean variable X_{p_0,q_0} to be solved (denoting the equivalence of the initial states p_0 and q_0 of the two LTSs), therefore being optimal from this point of view.



3. Experiments

The new BES encodings of weak equivalences were implemented in BISIMULATOR 2.0 and the sr-DFS algorithm was integrated to the generic CÆSAR_SOLVE library [9] for on-the-fly BES resolution, which serves as computing engine for BISIMULATOR and other tools of CADP. These two enhancements led to significant performance improvements w.r.t. BISIMULATOR 1.0, as shown by our experiments on LTSs coming from the demo examples of CADP (specifications of communication protocols and asynchronous circuits) or from the VLTS benchmark suite. As regards branching bisimulation (see the curves above), version 2.0 exhibits reductions of both the number of boolean variables and operators explored (up to a factor 9), which determine memory consumption and execution time, respectively.

4. Future Work

We plan first to extend the range of equivalences and preorders already available in BISIMULATOR 2.0 by devising BES encodings for testing equivalence and CFFD. Next, we will continue experimenting the sr-DFS algorithm and study its applicability for solving BESs coming from other verification problems, such as on-the-fly LTS reduction modulo partial order relations (e.g., τ-confluence, τ-inertness, etc.) as formulated in [10].

References

[1] R. Cleaveland and O. Sokolsky. Equivalence and Preorder Checking for Finite-State Systems. In *Handbook of Process Algebra*, chapter 6, pages 391–424, Elsevier, 2001.

[2] A. Dovier, C. Piazza, and A. Policriti. An efficient algorithm for computing bisimulation equivalence. *Th. Comp. Sci.*, 311(1–3):221–256, 2004.

[3] J-C. Fernandez and L. Mounier. Verifying bisimulations "on the fly". In *Proc. of FORTE'90*, 1990.

[4] H. Garavel. Open/cæsar: An open software architecture for verification, simulation, and testing. In *Proc. of TACAS'98*, LNCS vol. 1384, pp. 68–84, March 1998.

[5] H. Garavel, F. Lang, R. Mateescu, and W. Serwe. Cadp 2006: A toolbox for the construction and analysis of distributed processes. In *Proc. of CAV'2007*, LNCS vol. 4590, pp. 158–163, July 2007. See also http://www.inrialpes.fr/vasy/cadp.

[6] A. Ingolfsdottir and B. Steffen. Characteristic formulae for processes with divergence. *Information and Computation*, 110(1):149–163, 1994.

[7] A. Mader. *Verification of modal properties using boolean equation systems*. VERSAL 8, Bertz Verlag, Berlin, 1997.

[8] R. Mateescu. On-the-fly state space reductions for weak equivalences. In *Proc. of FMICS'05*, pages 80–89. ACM Computer Society Press, September 2005.

[9] R. Mateescu. Caesar_solve: A generic library for on-the-fly resolution of alternation-free boolean equation systems. *Springer Int. Journal on Software Tools for Technology Transfer (STTT)*, 8(1):37–56, February 2006.

[10] G. Pace, F. Lang, and R. Mateescu. Calculating τ-confluence compositionally. In *Proc. of CAV'2003*, LNCS vol. 2725, pp. 446–459, July 2003.

[11] S. K. Shukla and H. B. Hunt III and D. J. Rosenkrantz. Hornsat, model checking, verification and games. *Proc. of CAV'96*, LNCS vol. 1102, pages 99-110, 1996.