Bounded Model Checking for Knowledge and Linear Time [*]

(Extended Abstract)

A. Męski[†], W. Penczek[‡], M. Szreter
Institute of Computer Science, PAS, Poland
{meski,penczek,mszreter}@ipipan.waw.pl

B. Woźna-Szcześniak, A. Zbrzezny
Jan Długosz University, IMCS, Poland
{b.wozna,a.zbrzezny}@ajd.czest.pl

ABSTRACT

We investigate symbolic approaches to Bounded Model Checking (BMC) for the Linear Temporal Logic extended with epistemic components (LTLK), interpreted over Interleaved Interpreted Systems. We propose two BMC translations for LTLK - one is based on SAT and the other is based on BDD - which we have implemented and tested on several benchmarks. We report on our experimental results that reveal advantages and disadvantages of SAT- versus BDD-based BMC for LTLK.

Categories and Subject Descriptors

F.3.1 [**Specifying and Verifying and Reasoning about Programs**]: Specification techniques; D.2.4 [**Software/Program Verification**]: Model checking; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Modal logic, Temporal logic

General Terms

Verification, Theory, Performance

Keywords

SAT, BDD, Bounded Model Checking, Temporal Epistemic Logic (LTLK), Interleaved Interpreted Systems

1. INTRODUCTION

Several approaches based on model checking [1] have been put forward for verification of multi-agent systems (MAS) [2, 8, 9]. Typically, they employ combinations of epistemic logic with branching or linear temporal logic. Some approaches reduce the verification problem to the one for plain temporal logic, while others treat typical MAS modalities such as (distributed, common) knowledge as first-class citizens and introduce novel algorithms for them.

In an attempt to alleviate the state-space explosion problem (i.e., an exponential growth of the system state space with the number of the agents) two main BMC approaches have been proposed, based on either BDDs [3] or SAT [8]. However, these approaches deal with the properties expressed in CTLK (i.e., CTL extended the with epistemic component) only.

In this short paper we aim at completing the picture of applying mathods based on the BMC symbolic verification to MAS by looking at the existential part of LTLK (i.e., ELTLK) interpreted on *interleaved interpreted systems* (IIS) [6]. IIS are a special class of interpreted systems (IS) in which only one action at a time is performed in a global transition. Our original contribution consists in defining two novel model checking methods for LTLK, namely a SAT- and BDD-based BMC. The methods have been implemented, tested, and compared with each other as well as with the tool MCK [2] on three benchmarks for MAS. Our experimental results reveal advantages and disadvantages of SAT- versus BDD-based BMC for LTLK on MAS, which are consistent with comparisons for temporal logics. Although our methods are described for IIS, they can be applied to IS as well, which we will show in our future paper.

2. BMC FOR ELTLK

Our SAT- and BDD-based BMC methods for ELTLK are, to our best knowledge, the first ones formally presented in the literature; the manual for MCK states that the tool supports SAT-based BMC for CTL*K. Unfortunately, no theory behind this implementation has ever been published.

Let M be a model for a given IIS, φ - an ELTLK formula, and $k \geq 0$ - a bound. The problem of checking whether M is a model for φ can be translated to the problem of checking the satisfiability of the following propositional formula: $[M, \varphi]_k := [M^{\varphi,\iota}]_k \wedge [\varphi]_{M,k}$. The formula $[M^{\varphi,\iota}]_k$ constrains the finite number of symbolic k-paths to be valid k-paths of M, while the formula $[\varphi]_{M,k}$ encodes a number of constraints that must be satisfied on these sets of k-paths for φ to be satisfied. Once this translation is defined, checking satisfiability of an ELTLK formula can be done by means of a SAT-solver. In the case of the BDD-based approach we reduce the ELTLK model checking problem to the problem of the ELTL model checking. When processing the verified LTLK formula, the states of the model are labelled with the subformulae that hold in these states. This approach is similar to the approach proposed for CTL* [1]. To perform BMC using BDDs we interleave the fixed-point computation of the reachable states with executions of the state-labelling procedure for ELTLK. For details details we refer the reader to [7] and [10].

3. EXPERIMENTAL RESULTS

We consider three benchmarks for which we give performance evaluation of our two BMC algorithms and the BMC algorithm of MCK for the verification of several properties expressed in ELTLK.

[†] Also affiliated with: University of Łódź, FMCS, Poland.

[‡] Also affiliated with: University of Natural Sciences and Humanities, Institute of Informatics, Poland.

[*] Partly supported by the National Science Center under the grants No. 2011/01/B/ST6/05317 and 2011/01/B/ST6/01477.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

The tests have been performed on a computer with Intel Xeon 2 GHz processor and 4 GB of RAM, running Linux 2.6, with the default limits of 2 GB of memory and 2000 seconds of time. For every benchmark, each specification is given in the universal form, for which we verify the corresponding counterexample formula, i.e., the formula which is negated and interpreted existentially.

The presented approaches have been implemented as prototype modules of the tool VerICS [5]. All the benchmarks can be found at `http://verics.ipipan.waw.pl/bmcLTLK.zip`, together with instructions how to repeat our experiments.

Faulty Generic Pipeline Paradigm (FGPP) consists of Producer, Consumer, and a chain of n intermediate Nodes transmitting data, together with a chain of n Alarms enabled when some error occurs. We consider the following specifications:

$\varphi_1 = \mathrm{G}(ProdSend \rightarrow \mathrm{K}_C \mathrm{K}_P ConsReady)$,
$\varphi_2 = \mathrm{G}(Problem_n \rightarrow (\mathrm{F}Repair_n \vee \mathrm{G}Alarm_n Send))$,
$\varphi_3 = \bigwedge_{i=1}^{n} \mathrm{G}(Problem_i \rightarrow (\mathrm{F}Repair_i \vee \mathrm{G}Alarm_i Send))$,
$\varphi_4 = \bigwedge_{i=1}^{n} \mathrm{G}\mathrm{K}_P(Problem_i \rightarrow (\mathrm{F}Repair_i \vee \mathrm{G}Alarm_i Send))$.

A faulty train controller system (FTC) consists of a controller and n trains (for $n \geq 2$), one of which is dysfunctional. We consider the following specifications:

$\varphi_1 = \mathrm{G}(InTunnel_1 \rightarrow \mathrm{K}_{Train_1}(\bigwedge_{i=2}^{n} \neg InTunnel_i))$,
$\varphi_2 = \mathrm{G}(\mathrm{K}_{Train_1} \bigwedge_{i=1,j=2,i<j}^{n} \neg(InTunnel_i \wedge InTunnel_j))$.

Dining Cryptographers (DC) is a scalable anonymity protocol, which has been formalised and analysed in many works. Here we assume the formalisation of DC in terms of a network of automata [4], and we consider the following specifications:

$\varphi_1 = \mathrm{G}(odd \wedge \neg paid_1 \rightarrow \bigvee_{i=2}^{n} \mathrm{K}_1(paid_i))$,
$\varphi_2 = \mathrm{G}(\neg paid_1 \rightarrow \mathrm{K}_1(\bigvee_{i=2}^{n} paid_i))$,
$\varphi_3 = \mathrm{G}(odd \rightarrow \mathrm{C}_{\{1,\ldots,n\}} \neg(\bigvee_{i=1}^{n} paid_i))$.

Performance evaluation. An important difference in performance between the SAT- and BDD-based BMC reveals itself in the FTC benchmark, where the BDD-based method performs much better in terms of the total time and memory consumption. In the case of FGPP, BDD-BMC is still more efficient, but the difference is not that significant. Our SAT-based BMC significantly outperforms the BDD-based BMC for φ_2 of DC: SAT-BMC has computed the results for 3500 cryptographers, whereas BDD-BMC for 41. The reason is that there are at most two symbolic k-paths, and the length of the counterexamples is constant. This is also the case for φ_3 of FGPP. The efficiency of BDD-BMC improves for the formula φ_4 of FGPP comparing to φ_3, although they are similar. The reason is the presence of the knowledge operator that causes the partitioning of the problem to several smaller ELTL verification problems, which are handled much better by the implementation of the operations on BDDs. A noticeable superiority of SAT-BMC for φ_2 of DC follows from the long encoding times of the BDD for the transition relation. The reordering of the BDD variables does not cause any change of the performance in the case of FGPP and FTC, but for DC it reduces the memory consumption. This means that the fixed interleaving order we used can often be considered optimal, but the loss in the verification time to reorder the variables, in favour of reducing memory consumption, is also not significant and is often worth the tradeoff. In the case of φ_3 for DC, SAT-BMC was remarkably inferior to BDD-BMC, i.e., SAT-BMC managed to compute the results only for 3 cryptographers in the time of 5400 seconds, whereas BDD-BMC managed to compute the results for 17 cryptographers. This follows from the fact that φ_3 contains the common knowledge operator, which requires many symbolic k-paths to be analysed. For φ_1 of DC, our BDD-BMC has computed the results for 14 cryptographers, outperforming SAT-BMC (4 cryptographers). In most cases, BDD-BMC spends a considerable amount of time on encoding the system, whereas SAT-BMC on verifying the formula. Therefore, BDD-BMC may provide additional time gains when verifying multiple specifications of the same system.

We have compared MCK with our methods for the cases where the lengths of counterexamples scale correspondingly, thus minimising the factor played by different semantics. The comparison shows that for FGPP and FTC our methods are superior to MCK for all the tested formulae (sometimes by several orders of magnitude). There could be several reasons for this. While our approach is especially optimised for LTLK, it is likely that MCK treats LTLK formulae as CTL*K formulae, for which the translation is typically much less efficient. MCK consumes all the available memory even when formulae are surprisingly small (approx. 10^6 clauses and 10^5 variables) compared to those successfully tested in our SAT-BMC experiments (more than 10^8 clauses and variables in some cases). However, it should be noted that MCK implements different semantics of MAS, in which agents can perform independent actions simultaneously in a single step of the protocol, what may result in different counterexamples than given by IIS. This is the case of the DC benchmark, where MCK can profit from the strong locality and produces counterexamples of constant length, independently of the number of cryptographers, for all the formulae, being able to verify 15, 32, and 14 cryptographers for φ_1, φ_2, and φ_3, respectively. Using our approaches, we could verify, respectively, 14 cryptographers (BDD-BMC), 3500 (SAT-BMC), and 41 (BDD-BMC). We can conclude from our analysis that the BDD- and SAT-based BMC approches remain complementary and none of them is clearly superior in general, whereas in most cases MCK seems to be inferior to our BMC approaches.

4. REFERENCES

[1] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[2] P. Gammie and R. Meyden. MCK: Model checking the logic of knowledge. In *Proc. of CAV'04*, volume 3114 of *LNCS*, pp. 479–483. Springer-Verlag, 2004.

[3] A. Jones and A. Lomuscio. A BDD-based BMC approach for the verification of multi-agent systems. In *Proc. of CS&P'09*, volume 1, pp. 253–264. Warsaw University, 2009.

[4] M. Kacprzak, A. Lomuscio, A. Niewiadomski, W. Penczek, F. Raimondi, and M. Szreter. Comparing BDD and SAT based techniques for model checking Chaum's dining cryptographers protocol. *Fundam. Inform.*, 72(1-2):215–234, 2006.

[5] M. Kacprzak, W. Nabiałek, A. Niewiadomski, W. Penczek, A. Półrola, M. Szreter, B. Woźna, and A. Zbrzezny. VerICS 2007 - a Model Checker for Knowledge and Real-Time. *Fundam. Inform.* 85(1-4):313-328, 2008.

[6] A. Lomuscio, W. Penczek, and H. Qu. Partial order reduction for model checking interleaved multi-agent systems. In *AAMAS, IFAAMAS Press.*, pp. 659–666, 2010.

[7] A. Męski, W. Penczek, and M. Szreter. Bounded model checking linear time and knowledge using decision diagrams. In *Proc. of CS&P'11*, pp. 363–375, 2011.

[8] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundam. Inform.*, 55(2):167–185, 2003.

[9] F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via OBDDs. Journal of Applied Logic, 5(2):235–251, 2007.

[10] W. Penczek, B. Woźna-Szcześniak, and A. Zbrzezny. Towards SAT-based BMC for LTLK over interleaved interpreted systems. In *Proc. of CS&P'11*, pp 565–576, 2011.