Improved Bounded Model Checking for a Fair Branching-Time Temporal Epistemic Logic[*]

(Extended Abstract)

Xiaowei Huang, Cheng Luo, and Ron van der Meyden
University of New South Wales, Australia
xiaoweih,luoc,meyden@cse.unsw.edu.au

ABSTRACT

The paper develops a new approach to bounded model checking for a logic of knowledge and branching time. Experimental results are presented that demonstrate improved model checking performance, compared with previous approaches, on a range of examples.

Categories and Subject Descriptors

D.2.4 [Software Engineering]: Software/Program Verification

General Terms

Verification

Keywords

Bounded Model Checking, Temporal Epistemic Logics

1. INTRODUCTION

Bounded Model checking is a technique for verifying that a system satisfies a specification, based on a search for counter-examples to the validity of the specification using an encoding to propositional sastisfiability. A bounded model checking (BMC) encoding for a universal fragment of the branching time logic CTL, called ACTL, was proposed in [4], and improved in [6]. An extension of the [4] approach that adds epistemic operators, giving logic ACTLK$_n$, is given in [3]. We show in this paper that it is possible to significantly improve upon the efficiency of BMC for ACTLK$_n$. We develop an improved encoding for *fair* ACTLK$_n$ logic, which extends ACTLK$_n$ with a generalized Büchi fairness condition.

We show by both theoretical arguments and experimental results that our encoding yields an improved performance of BMC on a range o f examples. Theoretically, there are examples where the size of the encoding is reduced from exponential to quadratic. One such example is the “nested knowledge" formula $(K_aK_b)^n p$ expressing that two agents a, b have degree n mutual knowledge of the proposition p. In our experimental results, we have implemented two encoding functions, in the BDD-based epistemic model checker MCK [2], the encoding of [3] and our new encoding. The experimental results show that our BMC encoding yields a much better performance than the previous BMC encoding in all cases. Comparison with BDD model checking depends on the example.

2. PRELIMINARIES

Let *Prop* be a set of atomic propositions and $Ags = \{1, \ldots, n\}$ be a set of n agents. The syntax of ACTLK$_n$ is given by the following grammar: $\phi ::= \; p \,|\, \neg p \,|\, \phi{\vee}\phi \,|\, \phi{\wedge}\phi \,|\, AX\phi \,|\, AF\phi \,|\, AG\phi \,|A(\phi U \phi) \,|\, K_i\phi$. Intuitively $A\phi$ means ϕ holds in all futures, X, F, G, U are the next, some future time and all future time and until operators (respectively), and $K_i\phi$ says agent i knows ϕ. Its semantics is given in a standard way based on structures $M = (W, I, \Rightarrow, \sim^1, \ldots, \sim^n, \pi, \chi)$ where W is a (finite) set of global states, $I \subseteq W$ is the set of initial states, $\Rightarrow \subseteq W \times W$ is a serial temporal transition relation, each $\sim_i \subseteq W \times W$ is an equivalence relation representing epistemic accessibility for agent $i \in Ags$, $\pi : W \Rightarrow \mathcal{P}(Prop)$ is a propositional interpretation, and $\chi \in \mathcal{P}(\mathcal{P}(W))$ is a *generalised Büchi fairness condition*. ECTLK$_n$ is the dual language based on existential temporal branching E and epistemic possibility operators $\overline{K_i} = \neg K_i \neg$.

The model checking problem is the following: given a system M and a specification ψ in ACTLK$_n$, compute whether $M \models_A \psi$, or equivalently, whether not $M \models_E \phi$ for the ECTLK$_n$ formula ϕ corresponding to $\neg\psi$.

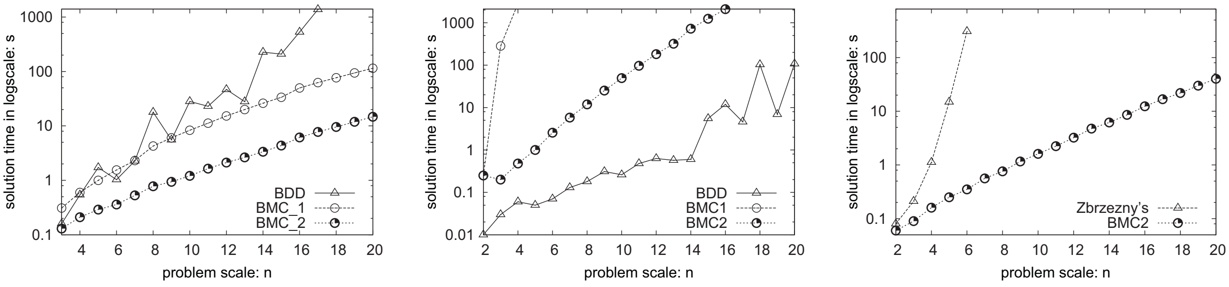
3. IMPROVED ENCODING FOR ACTLK$_n$

The basic idea underlying bounded model checking is to search for counter-examples to ψ, or equivalently, witnesses to ϕ, of increasingly large size k. The statement that the existential formula holds on a witness is encoded as a boolean satisfiability problem. In the approach of [3] for ACTLK$_n$, witnesses are collections R of cyclic runs of length k, and the encoding effectively evaluates all subformulas at all points of such runs, handling the search for a witness for an existential formula such as $EF\phi$ at a point (r, n) by means of a disjunctive formula, with a disjunct for each r' in R, where the disjuncts express that the point $(r, 0)$ has the same state as (r, n) and recursively calls the encoding for ϕ on $(r', 0)$. For the language ACTL, Zbrzezny [6]. shows that this disjunction can be eliminated by designating a specific run r' as providing the witness: this requires careful book-keeping to ensure that runs are not required to provide witnesses for multiple existential claims.

Our encoding for ACTLK$_n$ builds on Zbrzezny’s idea, but sharpens it by associating particular subformulas α with particular points (r, n) in the counter-example structure, and using atomic propositions $e_\alpha^{r,n}$ to represent the satisfaction of these subformulas at these points in a way that eliminates exponential blowups in previous encodings by means of structure-sharing. We illustrate this with some specific examples. The fact that the formula $EF\alpha$ holds at a

[*]Work supported by Australian Research Council Linkage Grant LP0882961 and Defence Research and Development Canada (Valcartier) contract W7701-082453.

Cite as: Improved Bounded Model Checking for a Fair Branching-Time Temporal Epistemic Logic (Extended Abstract), Xiaowei Huang, Cheng Luo and Ron van der Meyden, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1403-1404



(a) (b) (c)

Figure 1: Experimental Results

point (r, n) is expressed in the encoding by the atomic proposition $e^{r,n}_{EF\alpha}$. The encoding identifies a particular run r' as representing the branch from (r, n) that satisfies α, and expresses that the run r' satisfies α at some point. This is done by including in the encoding the formula

$$e^{r,n}_{\gamma} \Rightarrow b^{r,n,r',0} \wedge \bigvee_{i=0}^{k-1} e^{r',i}_{\alpha} \qquad (1)$$

where $b^{r,n,r',0}$ expresses that the states at (r, n) and at $(r', 0)$ are the same, and $e^{r',i}_{\alpha}$ (recursively) expresses that α holds at the point (r', i). This approach results in significant savings in encoding size compared to Zbrzezny's, particularly when dealing with nested modalities, where an exponential saving can be theoretically shown. Write $ef(r, n, \gamma, r')$ for formula (1). When encoding $\gamma = (EF)^h\alpha$ on point (r, n), our encoding has the form

$$ef(r, n, (EF)^h\alpha, r_1) \wedge \bigwedge_{j=1}^{h-1} \bigwedge_{i=0}^{k-1} ef(r_j, i, (EF)^{h-j}\alpha, r_{j+1})$$

which has size $O(hk^2)$. Zbrzezny's encoding has the form

$$H(r, n, r_1, 0) \wedge \bigvee_{i_1=0}^{k-1}(\ldots \bigvee_{i_{h-1}=0}^{k-1} (H(r_{h-1}, i_{h-1}, r_h, 0) \wedge \bigvee_{i_h=0}^{k-1}[\alpha]^{r_h,i_h}))$$

which has size $\Theta(k^h)$. Zbrzezny did not deal with epistemic operators. In our encoding these are handled by a new idea, viz, the inclusion of atomic propositions $b^{r,n,r',j}_{i}$ that represent that the point (r, n) is indistinguishable to agent i from the point (r', j). We may then express that $\overline{K_i}\alpha$ holds at the point (r, n), with the witness provided on run r', by including in the encoding the formula

$$e^{r,n}_{\gamma} \Rightarrow b^{r',0}_{I} \wedge \bigvee_{j=0}^{k-1}(b^{r,n,r',j}_{i} \wedge e^{r',j}_{\alpha})$$

where $b^{r',0}_{I}$ expresses that the state at $(r', 0)$ is initial. For nested knowledge formulas, this encoding results in an exponential saving over the approach of [3], which requires the construction of a nested disjunctive formula that grows exponentially, whereas our approach has linear growth.

4. EXPERIMENTAL RESULTS

We have implemented the old ($\mathtt{BMC}_1$) and new ($\mathtt{BMC}_2$) ACTLK$_n$ encodings as extensions to the epistemic model checker MCK (BDD, with sifting optimization) and performed a range of experiments that demonstrate improved performance of bounded model checking in all cases. We give a sample of the performance results in Figure 1. Each experiment measured runtime in seconds (s) as a function of some parameter n of the problem. Since the results show exponential growth patterns (as is to be expected for SAT problems of increasing size), we use a log-scale for run-times.

In all our experiments the new BMC procedure outperforms the old, decreasing the constant c in the model 2^{cn} for performance as a function of n, sometimes significantly, and increasing the scale of problems that can be solved in reasonable runtimes. Figure 1(a) shows results for Chaum's n agent Dining Cryptographers protocol and a formula of the form $AG(\alpha \Rightarrow K_1(\beta))$ where α and β are propositional. Here the formula is fixed and the number of states in the system grows exponentially with the number of agents. In this case, we find that our new BMC procedure outperforms BDD-based model checking. (In most other cases, BDD outperforms $\mathtt{BMC}_2$, but we note that BDD does not return counterexamples, so BMC remains important for this purpose.)

In Figure 1(b) the protocol is the two agent Byzantine Generals Problem where n is the number of messages sent, and the formula has the form $AG(\alpha_0 \Rightarrow \kappa(\beta))$ where α and β are propositional and κ is the nested sequence of operators $K_1K_2K_1K_2\ldots$ of length $n-1$. Here $\mathtt{BMC}_2$gives a dramatic improvement over $\mathtt{BMC}_1$.

To compare with [6], we conducted some experiments on pure temporal formulae. The protocol in this case is the two agent Byzantine Generals Problem and the formula has the form $AG(\alpha_0 \Rightarrow AG(\alpha_1 \Rightarrow \ldots AG(\alpha_n)\ldots)$ where the α_i are propositional and there are $n+1$ AG operators. The performance improvement is dramatic.

5. REFERENCES

[1] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[2] Peter Gammie and Ron van der Meyden. Mck: Model checking the logic of knowledge. In Rajeev Alur and Doron Peled, editors, *CAV*, volume 3114 of *Lecture Notes in Computer Science*, pages 479–483. Springer, 2004.

[3] Wojciech Penczek and Alessio Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. In *AAMAS*, pages 209–216. ACM, 2003.

[4] Wojciech Penczek, Bozena Wozna, and Andrzej Zbrzezny. Bounded model checking for the universal fragment of CTL. *Fundam. Inform.*, 51(1-2):135–156, 2002.

[5] Bozena Wozna. ACTLS properties and bounded model checking. *Fundam. Inform.*, 63(1):65–87, 2004.

[6] Andrzej Zbrzezny. Improving the translation from ECTL to SAT. *Fundam. Inform.*, 85(1-4):513–531, 2008.