Towards Abstract Interpretation of Epistemic Logic

Ajspur, Mai; Gallagher, John Patrick

Publication date:
2012

Document Version
Early version, also known as pre-print

Citation for published version (APA):
Ajspur, M., & Gallagher, J. P. (2012). *Towards Abstract Interpretation of Epistemic Logic*. Abstract from 8th Scandinavian Logic Symposium, Roskilde, Denmark.

Towards Abstract Interpretation of Epistemic Logic

Mai Ajspur and John P. Gallagher

CBIT, Building 43.2, Roskilde University, 4000 Roskilde, Denmark
{ajspur,jpg}@ruc.dk

Abstract. The model-checking problem is to decide, given a formula ϕ and an interpretation M, whether M satisfies ϕ, written $M \models \phi$. Model-checking algorithms for temporal logics were initially developed with finite models (such as models of hardware) in mind so that $M \models \phi$ is decidable. As interest grew in model-checking infinite systems, other approaches were developed based on approximating the model-checking algorithm so that it still terminates with some useful output.
In this work we present a model-checking algorithm for a multiagent epistemic logic containing operators for common and distributed knowledge. The model-checker is developed as a function directly from the semantics of the logic, in a style that could be applied straightforwardly to derive model-checkers for other logics. Secondly, we consider how to abstract the model-checker using abstract interpretation, yielding a procedure applicable to infinite models. The abstract model-checker allows model-checking with infinite-state models. When applied to the problem of whether $M \models \phi$, it terminates and returns the set of states in M at which ϕ might hold. If the set is empty, then M definitely does not satisfy ϕ, while if the set is non-empty then M possibly satisfies ϕ.

1 Syntax and semantics of the logic CMAEL(CD)

We consider the logic **CMAEL(CD)** [1, 7] whose formulas $\phi \in \Phi$ are defined by the following grammar.

$$\varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \mathbf{D}_A\varphi \mid \mathbf{C}_A\varphi.$$

The variable p ranges over the set $\mathtt{AP}$ of *atomic propositions*, typically denoted by $p, q, r, \ldots$; the variable A ranges over the set of *coalitions* $\mathcal{P}^+(\Sigma)$, which is the set of of non-empty subsets of Σ, where Σ is a finite, non-empty set of (names for) *agents*, typically denoted by $a, b, \ldots$. The epistemic operators $\mathbf{D}_A$ and $\mathbf{C}_A$ are read as *it is distributed knowledge among A that ...* and *it is common knowledge among A that ...* respectively. When A is a singleton $\{a\}$ we often write it as a subscript a instead of $\{a\}$, for example $\mathbf{D}_a$ instead of $\mathbf{D}_{\{a\}}$.
The semantics of **CMAEL(CD)** is given in terms of coalitional multiagent epistemic models (CMAEMs). A CMAEM is a tuple $(\Sigma, S, \{\mathcal{R}_A^D\}_{A\in\mathcal{P}^+(\Sigma)}, \{\mathcal{R}_A^C\}_{A\in\mathcal{P}^+(\Sigma)}, L)$,

1. Σ is a finite, non-empty set of *agents*;
2. $S \neq \emptyset$ is a set of *states*;
3. for every $A \in \mathcal{P}^+(\Sigma)$, $\mathcal{R}_A^D$ is an equivalence relation on S, satisfying the condition $\mathcal{R}_A^D = \bigcap_{a\in A}\mathcal{R}_a^D$;
4. for every $A \in \mathcal{P}^+(\Sigma)$, $\mathcal{R}_A^C$ is the transitive closure of $\bigcup_{a\in A}\mathcal{R}_a^D$;
5. $L : S \mapsto \mathcal{P}(\mathtt{AP})$ is a *labelling function*, assigning to every state s the set $L(s)$ of atomic propositions true at s.

Let S be a set. We define functions $pre : ((S\times S)\times\mathcal{P}(S)) \to \mathcal{P}(S)$ and $\widetilde{pre} : ((S\times S)\times\mathcal{P}(S)) \to \mathcal{P}(S)$.

- $pre(\mathcal{R})(X) = \{s \mid \exists s' \in X : (s, s') \in \mathcal{R}\}$ returns the set of states having at least one of their successors (in relation $\mathcal{R}$) in the set $X \subseteq S$;

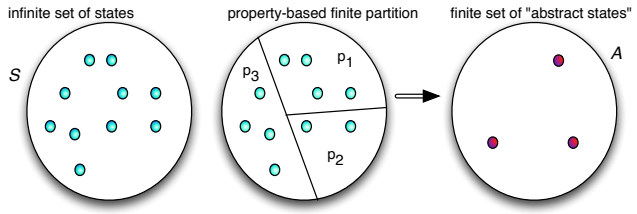

Fig. 1. Property Based Abstraction

– $\widetilde{pre}(\mathcal{R})(X) = S \setminus pre(\mathcal{R})(S \setminus X)$ returns the set of states all of whose successors are in the set $X \subseteq S$.

The functions pre and $\widetilde{pre}$ are defined by several authors (e.g. [6, 9]) and are also used with other names by other authors (e.g. they are called $pre_\exists$ and $pre_\forall$ by Huth and Ryan [8]).

*Semantic Function for **CMAEL(CD)**.* Let M be a CMAEM with states S; the following function $[\![.]\!]_M : \Phi \to \mathcal{P}(S)$ evaluates to the set of states of M where ϕ is true.

$$\begin{array}{lll} [\![p]\!]_M & = \{s \mid p \in L(s)\} & \\ [\![\neg\phi]\!]_M & = S \setminus [\![\phi]\!]_M & [\![\phi_1 \wedge \phi_2]\!]_M = [\![\phi_1]\!]_M \cap [\![\phi_2]\!]_M \\ [\![\mathbf{D}_A\phi]\!]_M & = \widetilde{pre}(\mathcal{R}_A^D)([\![\phi]\!]_M) & [\![\mathbf{C}_A\phi]\!]_M = \widetilde{pre}(\mathcal{R}_A^C)([\![\phi]\!]_M) \end{array}$$

This is closely related to the standard semantic relation $M, s \models \phi$ (ϕ holds at state s in M), which can be expressed as $s \in [\![\phi]\!]_M$. (We will transform the function above to eliminate the set complement operator, to avoid technical problems later when abstracting the function.)

1.1 Abstract Interpretation of CMAEL(CD)

What does it mean to perform "abstract model checking"? Informally, we check the satisfiability of a formula in a (possibly infinite) model using partial information about the model. The abstract interpretation framework [5] ensures that the result of the check is safe, in the sense that abstract checking returns false only when the formula is not satisfied by the model. A typical abstraction is based on a finite set of properties of interest $\{p_1, \ldots, p_k\}$ (see Figure 1), for example those atomic propositions appearing in the formula to be checked.[1] Suppose we have a model in which the set of states is infinite and in every state in S, exactly one p_i holds. Then the finite partition $A = \{d_1, \ldots, d_k\}$ is defined such that $d_i = \{s \in S \mid p_i \in L(s)\}$. In a property-based abstraction such as this, an abstract interpretation is constructed from two lattices $\langle \mathcal{P}(S), \subseteq \rangle$ and $\langle \mathcal{P}(A), \subseteq \rangle$ called the "concrete" and "abstract" domain respectively, and a *Galois connection* relating them. The Galois connection consists of monotonic functions $\alpha : \mathcal{P}(C) \to \mathcal{P}(A)$ and $\gamma : \mathcal{P}(A) \to \mathcal{P}(C)$ such that $\forall c \in \mathcal{P}(C), a \in \mathcal{P}(A), \alpha(c) \subseteq a \Leftrightarrow c \subseteq \gamma(a)$. In property-based abstractions, α and γ are defined as $\alpha(X) = \{d_i \mid d_i \cap X \neq \emptyset\}$ and $\gamma(Y) = \bigcup Y$. The concrete semantic function is $[\![.]\!]_M : \Phi \to \mathcal{P}(S)$ defined above. Given these components, the framework of abstract interpretation assists us in systematically deriving an abstract semantic function $[\![.]\!]^\sharp_M : \Phi \to \mathcal{P}(A)$ systematically from the concrete semantic function such that $[\![\phi]\!]_M \subseteq \gamma([\![\phi]\!]^\sharp_M)$, for all $\phi \in \Phi$ (or equivalently, $\alpha([\![\phi]\!]_M) \subseteq [\![\phi]\!]^\sharp_M$).
Thus in property-based abstract interpretation, the abstract semantic function returns a set of partitions (an element of $\mathcal{P}(A)$). The union of this set of partitions is a superset of the set of concrete states returned by the concrete semantic function. In particular, if $[\![\phi]\!]^\sharp_M$ is empty for some ϕ, then the result of the concrete computation $[\![\phi]\!]_M$ is also empty.

[1] Abstract interpretation is not limited to this kind of abstraction.

No Abstract Transition Relations In some approaches to abstract model checking [3, 4], a partition of the set of states is used to induce "abstract relations" in an "abstract model". This is not our approach, since it cannot simultaneously approximate both a formula and its negation. By contrast, the abstract semantic function derived below always returns an over-approximation of the set of states where any given formula holds. Other authors have developed "dual" approximations based on abstract relations to overcome the limitations of abstract models, but the framework of abstract interpretation offers a more direct solution, and was previously applied successfully to abstract model checking for CTL [2].

1.2 Abstract Semantic Function

The function $[\![.]\!]^\sharp_M : \Phi \to \mathcal{P}(A)$ is defined systematically from the concrete function $[\![\phi]\!]_M : \Phi \to \mathcal{P}(S)$ and the functions α and γ. We show it below. The transformation consists only of applying α to the base cases $\{s \mid p \in L(s)\}$ and $\{s \mid p \notin L(s)\}$ and replacing $pre(.)(.)$ (resp. $\widetilde{pre}(.)(.)$) by $\alpha(pre(.)(\gamma(.)))$ (resp. $\alpha(\widetilde{pre}(.)(\gamma(.)))$).

$$
\begin{array}{ll|ll}
[\![p]\!]^\sharp_M & = \alpha(\{s \mid p \in L(s)\}) & [\![\neg p]\!]^\sharp_M & = \alpha(\{s \mid p \notin L(s)\}) \\
[\![\phi_1 \wedge \phi_2]\!]^\sharp_M & = [\![\phi_1]\!]^\sharp_M \cap [\![\phi_2]\!]^\sharp_M & [\![\neg(\phi_1 \wedge \phi_2)]\!]^\sharp_M & = [\![\neg\phi_1]\!]^\sharp_M \cup [\![\neg\phi_2]\!]^\sharp_M \\
[\![\mathbf{D}_A\phi]\!]^\sharp_M & = \alpha(\widetilde{pre}(\mathcal{R}^D_A)(\gamma([\![\phi]\!]^\sharp_M))) & [\![\neg(\mathbf{D}_A\phi)]\!]^\sharp_M & = \alpha(pre(\mathcal{R}^D_A)(\gamma([\![\neg\phi]\!]^\sharp_M))) \\
[\![\mathbf{C}_A\phi]\!]^\sharp_M & = \alpha(\widetilde{pre}(\mathcal{R}^C_A)(\gamma([\![\phi]\!]^\sharp_M))) & [\![\neg(\mathbf{C}_A\phi)]\!]^\sharp_M & = \alpha(pre(\mathcal{R}^C_A)(\gamma([\![\neg\phi]\!]^\sharp_M))) \\
 & & [\![\neg\neg\phi]\!]^\sharp_M & = [\![\phi]\!]^\sharp_M
\end{array}
$$

Proposition 1. *For all formulas $\phi \in \Phi$, and CMAEM M, $[\![\phi]\!]_M \subseteq \gamma([\![\phi]\!]^\sharp_M)$.*

Proof. Proof is by structural induction on the formula ϕ and uses the properties of Galois connections and monotonic functions.

References

1. M. Ajspur, V. Goranko, and D. Shkatov. Tableau-based decision procedure. *CoRR*, abs/1201.5346, 2012.
2. G. Banda and J. P. Gallagher. Constraint-based abstract semantics for temporal logic: A direct approach to design and implementation. In E. M. Clarke and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 16th International Conference, LPAR-16, Dakar, Senegal, April 25-May 1, 2010, Revised Selected Papers*, volume 6355 of *Lecture Notes in Computer Science*, pages 27–45. Springer, 2010.
3. E. M. Clarke, O. Grumberg, and D. E. Long. Model checking and abstraction. In *Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Albuquerque, New Mexico, USA*, pages 342–354, 1992.
4. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.
5. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM Symposium on Principles of Programming Languages, Los Angeles*, pages 238–252, 1977.
6. P. Cousot and R. Cousot. Temporal abstract interpretation. In *Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, January 19 - 21 2000, Boston, Mass. USA*, pages 12–25, 2000.
7. V. Goranko and D. Shkatov. Tableau-based procedure for deciding satisfiability in the full coalitional multiagent epistemic logic. *CoRR*, abs/0902.2125, 2009.
8. M. R. A. Huth and M. D. Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge University Press, 2000.
9. P. Kelb. Model checking and abstraction: A framework preserving both truth and failure information. Technical report, Carl yon Ossietzky Univ. of Oldenburg, Oldenburg, Germany, 1994.