Hintikka's World: scalable higher-order knowledge

Tristan Charrier[1], **Sébastien Gamblin**[2], **Alexandre Niveau**[2,3] and **François Schwarzentruber**[4*]
[1]Univ Rennes, CNRS, IRISA, France
[2]Université de Caen Normandie, GREYC, Caen, France
[3]Université de Caen Normandie, GREYC, Caen, France
[4]Univ Rennes, CNRS, IRISA, France
tristan.charrier@irisa.fr, sebastien.gamblin@unicaen.fr, alexandre.niveau@unicaen.fr, francois.schwarzentruber@ens-rennes.fr

Abstract

Hintikka's World is a graphical and pedagogical tool that shows how artificial agents can reason about higher-order knowledge (agent a knows that agent b knows that...). In this demonstration paper, we present the implementation of symbolic models in Hintikka's World. They enable to the tool to scale, that is to face the state explosion, and to provide examples such as real card games, such as Hanabi.

1 Introduction

Constructing intelligent programs that play games with imperfect information is challenging: for instance, for Hanabi [Bard *et al.*, 2019], for Starcraft 2 [Hu *et al.*, 2018], etc. As far as we know, an important and missing ingredient is to reason about higher-order knowledge (an agent knows that another agents knows that...). In these systems, epistemic logic and its dynamic extension, Dynamic epistemic logic ([Baltag *et al.*, 1998], [van Ditmarsch *et al.*, 2008]) may offer formal tools for providing explanations in such AI programs. needs to be understood is relevant in AI, especially in strategic reasoning [Aumann, 1999].

The only pedagogical tool we are aware of explaining these models is *Hintikka's world* and was presented at ECAI-IJCAI 2018 [Schwarzentruber, 2018]. *Hintikka's world* is a proof of concept of a graphical user interface that represent Kripke models by comic strips, as shown in Figure 1. It enables to explore mental states of agents. The tool is available at the following address: http://hintikkasworld.irisa.fr/ and the source code is available here: https://gitlab.inria.fr/fschwarz/hintikkasworld

Kripke models are graphs, represented explicitly in memory in the first version of the tool. Explicit models are useful to learn how dynamic epistemic logic works by means of toy examples: muddy children, Sally and Anne [Wimmer and Perner, 1983], etc. However, the first version of Hintikka's world does not *scale*. For instance, in real card games, such as Hanabi, there are an exponential number of possible configurations of cards. For instance Hanabi has 50 cards total and each player has 4 cards and the order of the cards is important. Therefore with 4 players, the initial Kripke model features $50 \times 49 \times 48 \times \cdots \times 38$ configurations, that is 2.2×10^{21}. Thus, it is impossible to represent explicitly the full graph in memory.

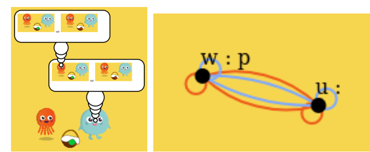

Figure 1: Graphical user interface of *Hintikka's world*.

That is why, in this demonstration, we propose to represent Kripke models symbolically by using the approach in [Charrier and Schwarzentruber, 2017] and [Charrier and Schwarzentruber, 2018]. The implementation relies on Binary Decision Diagrams (BDDs) [Bryant, 1986]. There is another implementation of symbolic epistemic models, called DEMO [van Benthem *et al.*, 2015], but their implementation is difficult to use in a web application and has no graphical user interface.

2 Demonstration Outline

In the demonstration we show a play of some variant of the game Hanabi. In this game, each agent has cards with a color and a number, but cannot see his own hand. In each turn, the user can play the role of one of the agent: he/she can either give the information to some other agent about a number or a color, or play a card. The goal is to play the cards in increasing order for each color. During the process, the system keeps track on the knowledge of the agents. More precisely, the system shows the real world (the real distribution of the cards). When the user clicks on an agent, it displays a *sampling* of some possible worlds for that agent (some possible distributions of cards he/she still consider as possible at some stage of the game). The agents also reason about knowledge of other agents, see in Figure 2 (two levels of knowledge are shown).

Note that in this demonstration, in order to explain models of DEL, we will continue to show simpler examples, that rely on explicit models. These examples were already presented in 2018 [Schwarzentruber, 2018]: Sally and Anne, muddy

*Contact Author

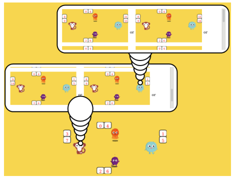

Figure 2: Screenshot of Hanabi in *Hintikka's World*.

children, consecutive numbers, etc.

3 Symbolic models

In our tool, we definitely emphasize on the use of model checking over theorem proving, as advocated in [Halpern and Vardi, 1991]. More precisely, we use the same ideas than in symbolic model checking, as defined for temporal logics [Burch *et al.*, 1990], adapted to DEL, as explained in [Charrier and Schwarzentruber, 2017] and [Charrier and Schwarzentruber, 2018]. Our model checking procedure relies now on symbolic Kripke models, aimed at representing succinctly so-called pointed Kripke models. A pointed Kripke model is a graph whose nodes are *possible worlds*, edges are labeled by agents and an edge $w \rightarrow^a u$ means that agent a considers world u as possible in world w. Each world w is equipped with a valuation telling which atomic propositions is true in w. A special world is called the pointed world and represents the true situation, while the other possible worlds are worlds imagined by the agents. The tool shows that graph in the right-part of the screen (in the example of Figure 1, the Kripke model has two possible worlds, w and u, p is true in w but not in u, and $\rightarrow_a$ is given in red and $\rightarrow_b$ in blue).

A symbolic model gives a Boolean formula $\chi(\vec{x})$ that succinctly describes the set of possible worlds: a world is a valuation over Boolean variables $\vec{x}$ satisfing $\chi(\vec{x})$. It also gives, for each agent a, a Boolean formula $\pi_a(\vec{x}, \vec{x}')$ that tells whether there is an edge labeled by agent a from a world described by a valuation over $\vec{x}$ and a world described by a valuation over $\vec{x}'$. All these Boolean formula are then classically converted in BDDs. Typically, for Hanabi, $\chi(\vec{x})$ tells that $\vec{x}$ describes an initial possible configuration. Formula $\pi_a(\vec{x}, \vec{x}')$ tells that the agents diferent from a have the same card in $\vec{x}$ and $\vec{x}'$ (it models the fact that agent a sees the cards of the other players).

Dynamic epistemic logic also provides so-called *event models* for describing actions (public announcements, public actions, private announcements/actions, etc.). The reader may refer to the textbook on DEL [van Ditmarsch *et al.*, 2008] and to [Charrier and Schwarzentruber, 2017] for symbolic event models, that we do not detail here.

4 System Description

Whereas the first version was written in Javascript In order to ease the development, the new version is written in TypeScript and Angular 7.

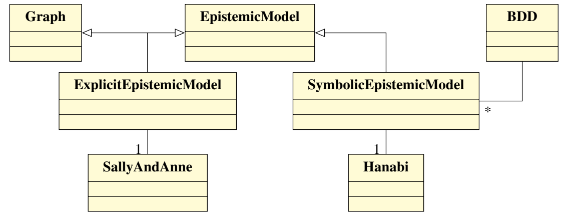

Figure 3: New architecture of *Hintikka's world*.

4.1 Binary decision diagrams

As shown in [Charrier and Schwarzentruber, 2017], the symbolic model checking of DEL is PSPACE-complete, thus is critical. We manipulate set of worlds, and relations by means of Binary decision diagrams. To this aim, we wrote a wrap-up in C of the library CUDD (Colorado University Decision Diagram Package) [Somenzi, 2001], that produces a Web Assembly library and a Javascript module.

In order to show possible worlds for a given agent a in some world w, we first construct the BDD of $\pi_a(descr(w), \vec{x}')$ where $descr(w)$ are the Boolean values of $\vec{x}$ corresponding to world w. We then count the number of possible valuations $\vec{x}'$ than makes for randomly values for $\vec{x}'$ that makes $\pi_a(descr(w), \vec{x}')$ true (BDDs are convenient for counting, see TODO). If the number of such valuations is small, we show all the possible worlds, otherwise we randomly generate valuations for $\vec{x}'$ that makes $\pi_a(descr(w), \vec{x}')$ true (we randomly select a branch that leads to the true-leaf in the BDD of $\pi_a(descr(w), \vec{x}')$).

4.2 Class Architecture

Figure 3 shows the new architecture of *Hintikka's world*. `EpistemicModel` is an abstract class used by the graphical user interface (GUI), that is independent from the current runnin example (muddy children, Sally and Anne, Hanabi, etc.) but more interestingly independent from the representation of the epistemic model itself. In particular, an epistemic model can be an `ExplicitEpistemicModel` (a graph) or a `SymbolicEpistemicModel` that relies on BDDs, depending on the examples: it suffices to implement the method `draw` of a class that inherits from class `World`.

4.3 Adding new examples

The system is easy to use to provide new examples. Explicit epistemic models are directly described (set of nodes and of edges). Symbolic epistemic models are described by a Boolean formula χ, or Boolean formulas for π_a. The system provides a way to easily describe how worlds are displayed in the comic strips.

5 Future Work

In the future, we plan to implement many examples. We plan to extend our implementation to Algebraic decision diagrams (ADD) [Bahar *et al.*, 1997] in order to tackle examples with numerical values, and not only Boolean values.

We aim at studying the use of counting and sampling specific techniques (see for instance [Meel *et al.*, 2016]) for generating possible worlds, in order to tackle even bigger examples.

References

[Aumann, 1999] Robert J. Aumann. Interactive epistemology I: knowledge. *Int. J. Game Theory*, 28(3):263–300, 1999.

[Bahar *et al.*, 1997] R. Iris Bahar, Erica A. Frohm, Charles M. Gaona, Gary D. Hachtel, Enrico Macii, Abelardo Pardo, and Fabio Somenzi. Algebraic decision diagrams and their applications. *Formal Methods in System Design*, 10(2/3):171–206, 1997.

[Baltag *et al.*, 1998] Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56. Morgan Kaufmann Publishers Inc., 1998.

[Bard *et al.*, 2019] Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The hanabi challenge: A new frontier for AI research. *CoRR*, abs/1902.00506, 2019.

[Bryant, 1986] Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Computers*, 35(8):677–691, 1986.

[Burch *et al.*, 1990] Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. Symbolic model checking: 10^20 states and beyond. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 428–439, 1990.

[Charrier and Schwarzentruber, 2017] Tristan Charrier and François Schwarzentruber. A succinct language for dynamic epistemic logic. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 123–131, 2017.

[Charrier and Schwarzentruber, 2018] Tristan Charrier and François Schwarzentruber. Complexity of dynamic epistemic logic with common knowledge. In *Advances in Modal Logic 12, proceedings of the 12th conference on "Advances in Modal Logic," held in Bern, Switzerland, August 27-31, 2018*, pages 103–122, 2018.

[Halpern and Vardi, 1991] Joseph Y. Halpern and Moshe Y. Vardi. Model checking vs. theorem proving: A manifesto. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991.*, 1991.

[Hu *et al.*, 2018] Yue Hu, Juntao Li, Xi Li, Gang Pan, and Mingliang Xu. Knowledge-guided agent-tactic-aware learning for starcraft micromanagement. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 1471–1477, 2018.

[Meel *et al.*, 2016] Kuldeep S. Meel, Moshe Y. Vardi, Supratik Chakraborty, Daniel J. Fremont, Sanjit A. Seshia, Dror Fried, Alexander Ivrii, and Sharad Malik. Constrained sampling and counting: Universal hashing meets SAT solving. In *Beyond NP, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 12, 2016.*, 2016.

[Schwarzentruber, 2018] François Schwarzentruber. Hintikka's world: Agents with higher-order knowledge. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 5859–5861, 2018.

[Somenzi, 2001] Fabio Somenzi. Efficient manipulation of decision diagrams. *STTT*, 3(2):171–181, 2001.

[van Benthem *et al.*, 2015] Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for dynamic epistemic logic. In *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan*, pages 366–378, 2015.

[van Ditmarsch *et al.*, 2008] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer, Dordecht, 2008.

[Wimmer and Perner, 1983] Heinz Wimmer and Josef Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128, 1983.

List of requirements/description of the demo setting

- Table, poster, monitor.