How to represent a Kripke model? From BDDs to Mental Programs

Haitian Wang*

May 10, 2024

Kripke models with explicit relations are the standard semantics in (dynamic) epistemic logic, but their sizes can become too large for practical model checking. To tackle this state-explosion, in the recent literature mostly two ways to encode the relations in Kripke models have been presented:

- *Binary Decision Diagrams* (BDDs) over a double-vocabulary, used for symbolic structures [Ben+18; Bry86],
- *Mental programs*, a variant of PDL used for succinct models [CS17].

Actual implementations of DEL model checking, such as SMCDEL [Ben+18], use BDDs[1], but only for succinct models a good theoretical complexity result is known. Therefore here we study how we can translate from BDDs to mental programs (without going via the potentially too large Kripke models) so that we can do model checking via this translation.

In ongoing work we have defined translations in both directions and proven them to be correct — here we focus on the direction from BDDs to mental programs. In this talk we will first recap both representations, discuss the differences with an example, and prove the correctness of the translation.

Boolean Formulas The symbolic structures from [Ben+18] encode the epistemic relations of agents using Boolean formulas (or their BDD) over a double vocabulary: $\Omega_i \in \mathcal{L}_B(V \cup V')$[2]. The intuition behind Ω_i is that it is a formula which is true at a pair of states iff the states are related.

Definition 1. *Throughout this abstract we fix a finite vocabulary V and denote a fresh copy of it by V'. A Boolean formula $\Omega_i \in \mathcal{L}_B(V \cup V')$ encodes a relation over $\mathcal{P}(V)$ given by $R_i st :\iff s \cup t' \vDash \Omega_i$ where s and t are states, i.e. subsets of V.*

For example, if $\Omega_{\text{Alice}} = p \wedge q'$ then from any state where p is true Alice will consider any other state where q is true possible and there are no other connections. This way of encoding relations is well-known and widely used, also for symbolic model checking of temporal logics[3]. In actual model checking we use the corresponding BDDs of boolean formulas but the idea is the same.

Mental Programs The succinct DEL models from [CS17] encode the relations for each agent using *mental programs*, a variant of Propositional Dynamic Logic (PDL) with the following syntax and semantics:

Definition 2. *A mental program π over a vocabulary V is recursively defined as:*

$$\pi ::= p \leftarrow \top \mid p \leftarrow \bot \mid \beta? \ \mid \pi \cup \pi \mid \pi ; \pi \mid \pi \cap \pi$$

where $p \in V$ and β is a Boolean formula over V.

*This is joint work with Malvin Gattinger (ILLC), Avijeet Gosh (Indian Statistical Institute) and Gregor Behnke (ILLC).

[1] For those unfamiliar with BDDs, each BDD can be viewed as a compact representation of a truth table. See the example below. More information about Binary Decision Diagrams can be found in [Ben+18] .

[2] They generalize knowledge structures (V, θ, O_i) which only work for S5.

[3] For further explanation, see for example [CG18, Section 8.3] where these formulas are also called the *characteristic function* of a relation.

Two states $s, t \subseteq V$ are related by a mental program π (written as $s \xrightarrow{\pi} t$) as follows:

$$\begin{aligned}
s \xrightarrow{p \leftarrow \top} t &\quad \leftrightarrow \quad t = s \cup \{p\} \\
s \xrightarrow{p \leftarrow \bot} t &\quad \leftrightarrow \quad t = s \setminus \{p\} \\
s \xrightarrow{\beta ?} t &\quad \leftrightarrow \quad s = t \text{ and } s \vDash \beta \\
s \xrightarrow{\pi_1 \cup \pi_2} t &\quad \leftrightarrow \quad s \xrightarrow{\pi_1} t \text{ or } s \xrightarrow{\pi_2} t \\
s \xrightarrow{\pi_1 ; \pi_2} t &\quad \leftrightarrow \quad \exists u \subseteq V : \ s \xrightarrow{\pi_1} u \text{ and } u \xrightarrow{\pi_2} t \\
s \xrightarrow{\pi_1 \cap \pi_2} t &\quad \leftrightarrow \quad s \xrightarrow{\pi_1} t \text{ and } s \xrightarrow{\pi_2} t
\end{aligned}$$

We define $R_\pi := \{(s, t) \in V \times V \mid s \xrightarrow{\pi} t\}$.

Translations We have defined translations between succinct and symbolic DEL models. Specifically, we provide a translation from BDDs to Mental Programs which somewhat surprisingly is in fact easier to define and implement compared to a translation from Boolean Formulas to Mental Programs. Here we give a toy example to illustrate the ideas:

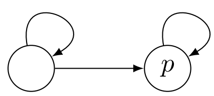

Figure 1: Relation R.

$$(?\top) \cup (p \leftarrow \top)$$

Figure 2: R's mental program.

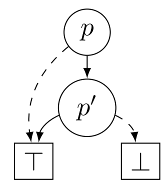

Figure 3: R's BDD.

Figure 1 shows the relation to be encoded using mental programs and BDDs. In Figure 2, $?\top$ corresponds to the two reflexive arrows and $p \leftarrow \top$ corresponds to the arrows from each world to its counterpart[4] in which p is true. Then we take the union of the relations and the resulting relations are the same as those in Figure 1. In Figure 3, the BDD represents the boolean formula $\neg p \vee p'$ [5]. Given the definition of belief structures, we can check it also represents the relation in Figure 1.

The translations we proposed would enable us to translate between mental programs and BDDs so that they both represent the same relation as this example illustrates. This might make DEL model checking more efficient as we can translate the BDDs to mental programs and do model checking on mental programs. However, this needs to be confirmed by benchmark results.

Future Work As a main theoretical application of the translation we aim to transfer the complexity results for succinct DEL model checking to symbolic knowledge structures. On the practical side, based on the code from [Har20] we are implementing a model checker directly working on succinct models with mental programs and will compare its performance with the already existing implementation using BDDs, on examples such as the Hanabi card game.

Bibliography

[Ben+18] Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. "Symbolic Model Checking for Dynamic Epistemic Logic — S5 and Beyond". In: *Journal of Logic and Computation* 28 (2 2018), pages 367–402. `https://doi.org/10.1093/logcom/exx038`.

[4] Given Definition 2, if p is false in a world w, then the arrow goes from w to the world in which p is true and everything else remains the same with w. If p is true in w, the arrow goes back to w itself.

[5] In this particular example, the only possible way to reach $\bot$ is by having p set to true and p' set to false. It can be viewed as a truth table in which the only row that has 0 in the last column is the one in which p is true and p' is false, therefore $\neg p \vee p'$.

[Bry86] Randal E Bryant. “Graph-based algorithms for boolean function manipulation”. In: *Computers, IEEE Transactions on* 100.8 (1986), pages 677–691.

[CG18] Sagar Chaki and Arie Gurfinkel. “BDD-Based Symbolic Model Checking”. In: *Handbook of Model Checking*. Edited by Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. Cham, 2018, pages 219–245. ISBN: 978-3-319-10575-8. https://doi.org/10.1007/978-3-319-10575-8_8.

[CS17] Tristan Charrier and François Schwarzentruber. “A Succinct Language for Dynamic Epistemic Logic”. In: *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*. Edited by S. Das, E. Durfee, K. Larson, and M. Winikoff. AAMAS ’17. International Foundation for Autonomous Agents and Multiagent Systems, 2017, pages 123–131. https://www.ifaamas.org/Proceedings/aamas2017/pdfs/p123.pdf.

[Har20] Maickel Hartlief. *Making Model Checking Scalable: Implementing Succinct Kripke Models for Public Announcement Logic*. BSc thesis, Groningen. Code at https://github.com/Maickel-Hartlief/SucExpModelCheckers. 2020. https://fse.studenttheses.ub.rug.nl/23607/.