Symbolic Model Checking for Dynamic Epistemic Logic

Johan van Benthem[1,2], Jan van Eijck[1,3], Malvin Gattinger[1], and Kaile Su[4,5]

[1] Institute for Logic, Language & Computation (ILLC), University of Amsterdam
[2] Department of Philosophy, Stanford University
[3] Centrum Wiskunde & Informatica, Amsterdam
[4] Institute for Integrated and Intelligent Systems, Griffith University
[5] Department of Computer Science, Jinan University

Abstract. Dynamic Epistemic Logic (DEL) can model complex information scenarios in a way that appeals to logicians. However, existing DEL implementations are ad-hoc, so we do not know how the framework really performs. For this purpose, we want to hook up with the best available model-checking and SAT techniques in computational logic. We do this by first providing a bridge: a new faithful representation of DEL models as so-called knowledge structures that allow for symbolic model checking. Next, we show that we can now solve well-known benchmark problems in epistemic scenarios much faster than with existing DEL methods. Finally, we show that our method is not just a matter of implementation, but that it raises significant issues about logical representation and update.

1 Introduction

We bring together two strains in the area of epistemic model checking. On one side, there are many frameworks for symbolic model checking on interpreted systems using temporal logics [24,30]. On the other hand, there are explicit model checkers for variants of Dynamic Epistemic Logic (DEL) like DEMO [15] with inferior performance but superior usability as they allow specification in dynamic languages directly. The goal of our work is to connect the two worlds of symbolic model checking and DEL in order to gain new insights on both sides.

Existing work on model checking DEL mainly focuses on specific examples, for example the Dining Cryptographers [28], the Sum and Product riddle [26] or Russian Cards [12]. Given these specific approaches, a general approach to symbolic model checking the full DEL language is desirable. A first step is [30] which presents symbolic model checking for temporal logics of knowledge. However, it does not cover announcements or other dynamics. The framework here extends these ideas with dynamic operators and a twist on the semantics.

Our knowledge structures are similar in spirit to hypercubes from [25], but of a different type: We do not use interpreted systems and temporal relations are not part of our models. Hence also our language does not contain temporal operators but primitives for epistemic events like announcements.

W. van der Hoek et al. (Eds.): LORI 2015, LNCS 9394, pp. 366–378, 2015.
DOI: 10.1007/978-3-662-48561-3_30

Related to our work is also [13] where DEL is translated into temporal epistemic logics for which symbolic model checkers exist. However, this method has not been implemented and the complexity and performance are not known. We do not translate to a temporal logic but check DEL formulas directly.

The paper is structured as follows. In Section 2 we recall standard semantics of DEL as in [11]. We then present knowledge structures in Section 3 and discuss the famous Muddy Children example in Section 4, together with experimental results in Section 5. Section 6 is a case study of the Russian Cards problem. Our main theoretical results are in Section 7: Knowledge structures are equivalent to S5 Kripke models. Moreover, S5 action models from [1] can be described in the same way. Section 8 gives a conclusion and suggestions for further research.

All source code can be found at `https://github.com/jrclogic/SMCDEL`.

2 Dynamic Epistemic Logic on Kripke Models

Definition 1. *Fix a set of propositions V and a finite set of agents I. The DEL language $\mathcal{L}(V)$ is given by*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C_\Delta\varphi \mid [\varphi]\varphi \mid [\varphi]_\Delta\varphi$$

where $p \in V$, $i \in I$ and $\Delta \subseteq I$. We also use the abbreviations $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$ and $\varphi \to \psi := \neg(\varphi \wedge \neg\psi)$. The boolean *formulas are $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi$.*

The formula $K_i\varphi$ is read as *"agent i knows φ"* while $C_\Delta\varphi$ says that φ is common knowledge among agents in Δ. The formula $[\psi]\varphi$ indicates that after a *public announcement* of ψ, φ holds. In contrast, $[\psi]_\Delta\varphi$ says that after announcing ψ to the agents in Δ, φ holds. The standard semantics for $\mathcal{L}(V)$ are given by means of Kripke models as follows.

Definition 2. *A* Kripke model *for n agents is a tuple $M = (W, \pi, \mathcal{K}_1, \cdots, \mathcal{K}_n)$, where W is a set of* worlds*, π associates with each world a truth assignment to the primitive propositions, so that $\pi(w)(p) \in \{\top, \bot\}$ for each world w and primitive proposition p, and $\mathcal{K}_1, \cdots, \mathcal{K}_n$ are binary accessibility relations on W. By convention, W^M, $\mathcal{K}_i^M$ and π^M are used to refer to the components of M. We omit the superscript M if it is clear from context. Finally, let $\mathcal{C}_\Delta^M$ be the transitive closure of $\bigcup_{i\in\Delta} \mathcal{K}_i^M$.*

A pointed Kripke model *is a pair (M, w) consisting of a Kripke model and a world $w \in W^M$. A model M is called an* S5 Kripke model *iff, for every i, $\mathcal{K}_i^M$ is an equivalence relation. A model M is called* finite *iff W^M is finite.*

Definition 3. *Semantics for $\mathcal{L}(V)$* on pointed Kripke models *are given inductively as follows.*

1. $(M, w) \models p$ *iff* $\pi^M(w)(p) = \top$.
2. $(M, w) \models \neg\varphi$ *iff not* $(M, w) \models \varphi$
3. $(M, w) \models \varphi \wedge \psi$ *iff* $(M, w) \models \varphi$ *and* $(M, w) \models \psi$
4. $(M, w) \models K_i\varphi$ *iff for all* $w' \in W$, *if* $w\mathcal{K}_i^M w'$, *then* $(M, w') \models \varphi$.

5. $(M, w) \models C_\Delta \varphi$ *iff for all* $w' \in W$, *if* $w \mathcal{C}_\Delta^M w'$, *then* $(M, w') \models \varphi$.
6. $(M, w) \models [\psi]\varphi$ *iff* $(M, w) \models \psi$ *implies* $(M^\psi, w) \models \varphi$ *where* M^ψ *is a new Kripke model defined by the set* $W^{M^\psi} := \{w \in W^M \mid (M, w) \models \psi\}$, *the relations* $\mathcal{K}_i^{M^\psi} := \mathcal{K}_i^M \cap (W^{M^\psi})^2$ *and the valuation* $\pi^{M^\psi}(w) := \pi^M(w)$.
7. $(M, w) \models [\psi]_\Delta \varphi$ *iff* $(M, w) \models \psi$ *implies that* $(M_\psi^\Delta, (1, w)) \models \varphi$ *where*
   (a) $W^{M_\psi^\Delta} := \{(1, w) \mid w \in W^M$ *and* $(M, w) \models \psi\} \cup \{(0, w) \mid w \in W^M\}$
   (b) *For* (b, w) *and* (b', w') *in* $W^{M_\psi^\Delta}$, *if* $i \in \Delta$, *let* $(b, w) \mathcal{K}_i^{M_\psi^\Delta} (b', w')$ *iff* $b = b'$ *and* $w \mathcal{K}_i^M w'$. *If* $i \notin \Delta$, *then let* $(b, w) \mathcal{K}_i^{M_\psi^\Delta} (b', w')$ *iff* $w \mathcal{K}_i^M w'$.
   (c) *For each* $(b, w) \in W^{M_\psi^\Delta}$, $\pi^{M_\psi^\Delta}((b, w)) := \pi^M(w)$.

Note that a group announcement $[\psi]_\Delta \varphi$ is private in the sense that only the agents in Δ obtain knowledge about ψ. However, the announcement is *not secret* because the other agents still learn that the agents in Δ might have learned ψ.

3 Knowledge Structures

While the preceding semantics is standard in logic, it cannot serve directly as an input to current sophisticated model-checking techniques. For this purpose, in this section we introduce a new format, *knowledge structures*. Their main advantage is that also knowledge and results of announcements can be computed via purely boolean operations. We first recapitulate some notions and abbreviations.

Given a set of propositional variables P, we identify a *truth assignment over* P with a subset of P. We say a formula φ is a formula *over* P if each propositional variable occurring in φ is in P. For convenience, we use the logical constants $\top$ and $\bot$ which are always true and always false, respectively. We also use $\models$ to denote the usual satisfaction relation between a truth assignment and a formula.

We use substitution and quantification as follows. For any formula φ and $\psi \in \{\top, \bot\}$, and any propositional variable p, let $\varphi(\frac{p}{\psi})$ denote the result of replacing every p in φ by ψ. For any $A = \{p_1, \ldots, p_n\}$, let $\varphi(\frac{A}{\psi}) := \psi(\frac{p_1}{\psi})(\frac{p_2}{\psi}) \ldots (\frac{p_n}{\psi})$, i.e. the result of substituting ψ for all elements of A. We use $\forall p \varphi$ to denote $\varphi\left(\frac{p}{\top}\right) \wedge \varphi\left(\frac{p}{\bot}\right)$. For any $A = \{p_1, \ldots, p_n\}$, let $\forall A \varphi := \forall p_1 \forall p_2 \ldots \forall p_n \varphi$.

Definition 4. *Suppose we have* n *agents. A* knowledge structure *is a tuple* $\mathcal{F} = (V, \theta, O_1, \ldots, O_n)$ *where* V *is a finite set of propositional variables,* θ *is a boolean formula over* V *and for each agent* i, $O_i \subseteq V$.

Set V *is the* vocabulary *of* $\mathcal{F}$. *Formula* θ *is the* state law *of* $\mathcal{F}$. *It determines the set of states of* $\mathcal{F}$ *and may only contain boolean operators. The variables in* O_i *are called agent* i*'s* observable variables. *An assignment over* V, *given as the set of true propositions, that satisfies* θ *is called a* state *of* $\mathcal{F}$. *Any knowledge structure only has finitely many states. Given a state* s *of* $\mathcal{F}$, *we say that* $(\mathcal{F}, s)$ *is a* scene *and define the* local state *of an agent* i *at* s *as* $s \cap O_i$.

Given a knowledge structure $(V, \theta, O_1, \cdots, O_n)$ *and a set* $\mathcal{V}$ *of subsets of* V, *we use* $\mathcal{E}_\mathcal{V}$ *to denote a relation between two assignments* s, s' *on* V *satisfying* θ *such that* $(s, s') \in \mathcal{E}_\mathcal{V}$ *iff there exists a* $P \in \mathcal{V}$ *with* $s \cap P = s' \cap P$. *We use* $\mathcal{E}_\mathcal{V}^*$

to denote the transitive closure of $\mathcal{E}_{\mathcal{V}}$. *Let* $\mathcal{V}_\Delta = \{O_i \mid i \in \Delta\}$. *We then have* $(s, s') \in \mathcal{E}_{\mathcal{V}_\Delta}$ *iff there exists an* $i \in \Delta$ *with* $s \cap O_i = s' \cap O_i$.

We now give alternative semantics for $\mathcal{L}(V)$ on knowledge structures. Definitions 5 and 6 run in parallel, both proceeding by the structure of φ.

Definition 5. *Semantics for DEL* on scenes *are defined inductively as follows.*

1. $(\mathcal{F}, s) \models p$ *iff* $s \models p$.
2. $(\mathcal{F}, s) \models \neg\varphi$ *iff not* $(\mathcal{F}, s) \models \varphi$
3. $(\mathcal{F}, s) \models \varphi \wedge \psi$ *iff* $(\mathcal{F}, s) \models \varphi$ *and* $(\mathcal{F}, s) \models \psi$
4. $(\mathcal{F}, s) \models K_i\varphi$ *iff for all* s' *of* $\mathcal{F}$, *if* $s \cap O_i = s' \cap O_i$, *then* $(\mathcal{F}, s') \models \varphi$.
5. $(\mathcal{F}, s) \models C_\Delta\varphi$ *iff for all* s' *of* $\mathcal{F}$, *if* $(s, s') \in \mathcal{E}^*_{\mathcal{V}_\Delta}$, *then* $(\mathcal{F}, s') \models \varphi$.
6. $(\mathcal{F}, s) \models [\psi]\varphi$ *iff* $(\mathcal{F}, s) \models \psi$ *implies* $(\mathcal{F}^\psi, s) \models \varphi$ *where* $\|\psi\|_\mathcal{F}$ *is given by Definition 6 and*
$$\mathcal{F}^\psi := (V, \theta \wedge \|\psi\|_\mathcal{F}, O_1, \cdots, O_n)$$
7. $(\mathcal{F}, s) \models [\psi]_\Delta\varphi$ *iff* $(\mathcal{F}, s) \models \psi$ *implies* $(\mathcal{F}^\Delta_\psi, s \cup \{p_\psi\}) \models \varphi$ *where* p_ψ *is a new propositional variable,* $\|\psi\|_\mathcal{F}$ *is given by Definition 6 and*
$$\mathcal{F}^\Delta_\psi := (V \cup \{p_\psi\}, \theta \wedge (p_\psi \to \|\psi\|_\mathcal{F}), O'_1, \cdots, O'_n)$$
where $O'_i := O_i \cup \{p_\psi\}$ *if* $i \in \Delta$ *and* $O'_i := O_i$ *otherwise.*

Before defining the boolean equivalents of formulas, we can already explain some similarities and differences between Definitions 3 and 5. The semantics of the boolean connectives are the same. For the knowledge operators, on Kripke models we use an accessibility relation $\mathcal{K}_i$. On knowledge structures this is replaced with the condition $s \cap O_i = s' \cap O_i$, inducing an equivalence relation on the states. We can already guess that knowledge structures encode S5 Kripke models.

Definition 6. *For any knowledge structure* $\mathcal{F} = (V, \theta, O_1, \cdots, O_n)$ *and any DEL formula* φ, *we define a boolean formula* $\|\varphi\|_\mathcal{F}$.

1. *For any primitive formula, let* $\|p\|_\mathcal{F} := p$.
2. *For negation, let* $\|\neg\psi\|_\mathcal{F} := \neg\|\psi\|_\mathcal{F}$.
3. *For conjunction, let* $\|\psi_1 \wedge \psi_2\|_\mathcal{F} := \|\psi_1\|_\mathcal{F} \wedge \|\psi_2\|_\mathcal{F}$.
4. *For knowledge, let* $\|K_i\psi\|_\mathcal{F} := \forall(V \setminus O_i)(\theta \to \|\psi\|_\mathcal{F})$.
5. *For common knowledge, let* $\|C_\Delta\psi\|_\mathcal{F} := \mathbf{gfp}\Lambda$ *where* Λ *is the following operator on boolean formulas given and* $\mathbf{gfp}\Lambda$ *denotes its greatest fixed point:*
$$\Lambda(\alpha) := \|\psi\|_\mathcal{F} \wedge \bigwedge_{i \in \Delta} \forall(V \setminus O_i)(\theta \to \alpha)$$
6. *For public announcements, let* $\|[\psi]\xi\|_\mathcal{F} := \|\psi\|_\mathcal{F} \to \|\xi\|_{\mathcal{F}^\psi}$.
7. *For group announcements, let* $\|[\psi]_\Delta\xi\|_\mathcal{F} := \|\psi\|_\mathcal{F} \to (\|\xi\|_{\mathcal{F}^\Delta_\psi})(\frac{p_\psi}{\top})$.

where $\mathcal{F}^\psi$ *and* $\mathcal{F}^\psi_\Delta$ *are as given by Definition 5.*

Given these definitions, a simple induction on φ gives us the following Theorem.

Theorem 1. *Definition 6 preserves and reflects truth. That is, for any formula* φ *and any scene* $(\mathcal{F}, s)$ *we have that* $(\mathcal{F}, s) \models \varphi$ *iff* $s \models \|\varphi\|_{\mathcal{F}}$.

We can now explain the public and group announcements. First observe that *public* announcements only modify the state law of the knowledge structure. Moreover, the new state law is always a conjunction containing the previous one. Hence the set of states is restricted, just like public announcements on Kripke models restrict the set of possible worlds. Second, note that a group announcement adds a single observational variable and can therefore at most double the number of states, just like in the Kripke semantics in Definition 3.

4 Example 1: Muddy Children

How does our new format do in practice? For this purpose, we consider some well-known benchmarks in the epistemic agency literature. We start with how their new representations looks like. After that, we go on to actual computational experiments. The famous Muddy Children example will illustrate how announcements, both of propositional and of epistemic facts, work on knowledge structures. An early version of the puzzle are the three ladies on a train in [23]. For a standard analysis with Kripke models, see [17, p. 24-30] or [11, p. 93-96].

Let p_i stand for "child i is muddy". We consider the case of three children $I = \{1, 2, 3\}$ who are all muddy, i.e. the actual state is $\{p_1, p_2, p_3\}$. At the beginning the children do not have any information, hence the initial knowledge structure $\mathcal{F}_0$ in Figure 1 has the state law $\theta_0 = \top$. All children can observe whether the others are muddy but do not see their own face. This is represented with observational variables: Agent 1 observes p_2 and p_3, etc. Now the father says: "At least one of you is muddy." This public announcement limits the set of states by adding this statement to the state law. Note that it already is a purely boolean statement, hence the formula is added as it is, leading to $\mathcal{F}_1$.

$$\mathcal{F}_0 = \left(\begin{array}{r} O_1 = \{p_2, p_3\} \\ V = \{p_1, p_2, p_3\}, \theta_0 = \top, \; O_2 = \{p_1, p_3\} \\ O_3 = \{p_1, p_2\} \end{array} \right)$$

$$\mathcal{F}_1 = \left(\begin{array}{r} O_1 = \{p_2, p_3\} \\ V = \{p_1, p_2, p_3\}, \theta_1 = (p_1 \vee p_2 \vee p_3), \; O_2 = \{p_1, p_3\} \\ O_3 = \{p_1, p_2\} \end{array} \right)$$

Fig. 1. Knowledge structures before and after the first announcement.

The father now asks "Do you know if you are muddy?" but none of the children does. As it is common in the literature, we understand this as a public announcement of "Nobody knows their own state.": $\bigwedge_{i \in I}(\neg(K_i p_i \vee K_i \neg p_i))$. This is not

a purely boolean formula, hence the public announcement is slightly more complicated: Using Definition 6 and Theorem 1 we find a boolean formula which on the current knowledge structure $\mathcal{F}_1$ is equivalent to the announced formula. Then this boolean equivalent is added to θ. We have

$$\begin{aligned}\|K_1 p_1\|_{\mathcal{F}_1} &= \forall(V \setminus O_1)(\theta_1 \to \|p_1\|_{\mathcal{F}_1}) = \forall p_1((p_1 \vee p_2 \vee p_3) \to p_1) \\ &= ((\top \vee p_2 \vee p_3) \to \top) \wedge ((\bot \vee p_2 \vee p_3) \to \bot) = \neg(p_2 \vee p_3)\end{aligned}$$

$$\begin{aligned}\|K_1 \neg p_1\|_{\mathcal{F}_1} &= \forall(V \setminus O_1)(\theta_1 \to \|\neg p_1\|_{\mathcal{F}_1}) = \forall p_1((p_1 \vee p_2 \vee p_3) \to \neg p_1) \\ &= ((\top \vee p_2 \vee p_3) \to \neg\top) \wedge ((\bot \vee p_2 \vee p_3) \to \neg\bot) = \bot\end{aligned}$$

and analogous for $K_2 p_2$, $K_2 \neg p_2$, $K_3 p_3$ and $K_3 \neg p_3$. These results make intuitive sense: In our situation where all children are muddy, a child knows it is muddy iff it sees that the other two children are clean. It can never know that it is clean itself. The announced formula becomes

$$\begin{aligned}\| \bigwedge_{i \in I} (\neg(K_i p_i \vee K_i \neg p_i))\|_{\mathcal{F}_1} &= \bigwedge_{i \in I} \|\neg(K_i p_i \vee K_i \neg p_i)\|_{\mathcal{F}_1} \\ &= \neg(\neg(p_2 \vee p_3)) \wedge \neg(\neg(p_1 \vee p_3)) \wedge \neg(\neg(p_1 \vee p_2)) \\ &= (p_2 \vee p_3) \wedge (p_1 \vee p_3) \wedge (p_1 \vee p_2)\end{aligned}$$

The announcement essentially says that at least two children are muddy. We get a knowledge structure $\mathcal{F}_2$ with the following more restrictive state law θ_2. Vocabulary and observational variables do not change, so we do not repeat them.

$$\theta_2 = (p_1 \vee p_2 \vee p_3) \wedge ((p_2 \vee p_3) \wedge (p_1 \vee p_3) \wedge (p_1 \vee p_2))$$

Now the same announcement ("Nobody knows their own state.") is made again. It is important that again we start with the epistemic formula $\bigwedge_{i \in I}(\neg(K_i p_i \vee K_i \neg p_i))$ and compute an equivalent formula with respect to $\mathcal{F}_2$. For reasons of space we skip tedious boolean reasoning and just note that

$$\|K_1 p_1\|_{\mathcal{F}_2} = \forall(V \setminus O_1)(\theta_2 \to \|p_1\|_{\mathcal{F}_2}) = \neg(p_3 \wedge p_2)$$

$$\|K_1 \neg p_1\|_{\mathcal{F}_2} = \forall(V \setminus O_1)(\theta_2 \to \|\neg p_1\|_{\mathcal{F}_2}) = \neg(p_2 \vee p_3)$$

which gives us $\|\neg(K_1 p_1 \vee K_1 \neg p_1)\|_{\mathcal{F}_2} = p_3 \wedge p_2$ and analogous formulas for children 2 and 3. Hence with respect to $\mathcal{F}_2$ we get the following boolean equivalent of the announcement, essentially saying that everyone is muddy.

$$\begin{aligned}\| \bigwedge_{i \in I} (\neg(K_i p_i \vee K_i \neg p_i))\|_{\mathcal{F}_2} &= (p_3 \wedge p_2) \wedge (p_3 \wedge p_1) \wedge (p_2 \wedge p_1) \\ &= p_1 \wedge p_2 \wedge p_3\end{aligned}$$

The resulting knowledge structure thus has the state law $\theta_3 = \theta_2 \wedge (p_1 \wedge p_2 \wedge p_3)$ which is in fact equivalent to $p_1 \wedge p_2 \wedge p_3$ and marks the end of the story: The only state left is the situation in which all three children are muddy.

5 Symbolic Model Checking: Implementation and Benchmarking

The previous section showed how epistemic operators get replaced by booleans when a new state law is computed. We could see that syntactically the state law becomes more and more complex, but semantically the same boolean function can be represented with a much shorter formula. This is where Binary Decision Diagrams (BDDs) come in extremely handy.

First presented in [5], BDDs provide an elegant data structure for boolean functions. In many cases they are less redundant and thus smaller than a corresponding truth table. Additionally, they can be manipulated efficiently: Given BDDs for φ and ψ we can compute the BDD for $\varphi \wedge \psi$, $\varphi \to \psi$ etc. Moreover, BDDs are canonical: Two formulas are equivalent iff their BDDs are identical. For an in-depth introduction, see [22, p. 202-280]. To see how BDDs can be used to describe knowledge structures, Figure 2 shows the BDDs for θ_0 to θ_3.

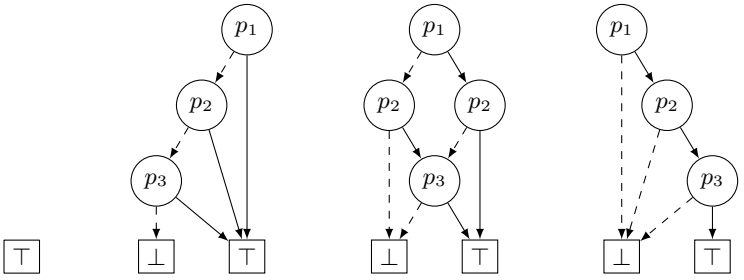

Fig. 2. Four BDDs representing the state laws θ_0 to θ_3.

Our new symbolic model checker SMCDEL works as follows: It takes two inputs, a scene $(\mathcal{F}, s)$ where the state law is given as a BDD, and a DEL formula φ. To check whether φ holds at state s we first compute the equivalent boolean formula $\|\varphi\|_{\mathcal{F}}$ according to Definition 6 and then check the boolean satisfaction $s \vDash \|\varphi\|_{\mathcal{F}}$. Alternatively, we can check whether a formula is valid on $\mathcal{F}$, i.e. true at *all* states, by checking whether $\theta \to \|\varphi\|_F$ is a tautology. The full set of states does not have to be generated and events are not executed explicitly.

We compared the performance of this method to DEMO-S5, an explicit model checker optimized for multi-agent S5 [15]. As a benchmark we used the question "For n muddy children, how many announcements of »Nobody knows their own state.« are needed until they do know their own state?". We measured how long each method takes to find and verify the correct answer, namely $n - 1$.

Figure 3 shows the results on a logarithmic scale: Explicit model checking with DEMO-S5 quickly becomes unfeasible whereas our symbolic model checker SMCDEL can deal with scenarios up to 40 agents in less than a second.

The model checker is implemented in Haskell and can be used similarly to DEMO-S5. To represent BDDs we use CacBDD [27] via the binding library

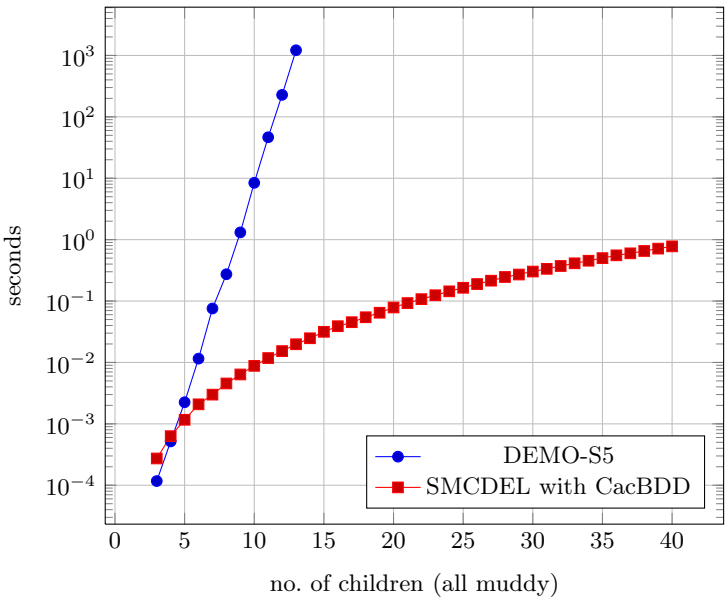

Fig. 3. Benchmark Results on a logarithmic scale.

HasCacBDD [19]. The program can also be used with CUDD [18,29] which provides very similar performance. All experiments were done using 64-bit Debian GNU/Linux 8.0 with kernel 3.16.0-4, GHC 7.8.3 and g++ 4.9 on an Intel Core i3-2120 3.30 GHz processor and 4 GB of memory.

Muddy Children has also been used to benchmark MCMAS [24] but the formula checked there concerns the correctness of behavior and not how many rounds are needed. Moreover, the interpreted system semantics of model checkers like MCMAS are very different from DEL. Still, connections between DEL and temporal logics have been studied and translations are available [3,13].

A scenario which fits nicely into both frameworks is the dining cryptographers protocol [7]. The statement "If cryptographer 1 did not pay the bill, then after the announcements are made, he knows that no cryptographers paid, or that someone paid, but in this case he does not know who did." is also checked in [24]. It can be formalized in DEL as follows where p_i says that agent i paid and ψ is the announcement: $\neg p_1 \to [\psi](K_1(\bigwedge_{i=1}^{n} \neg p_i) \vee (K_1(\bigvee_{i=2}^{n} p_i) \wedge \bigwedge_{i=2}^{n}(\neg K_1 p_i)))$. SMCDEL can check this for $n = 50$ in less than a second. Proper benchmarks and comparisons of all parameters will be done in the future.

6 Example 2: Russian Cards

As a second case study we applied our symbolic model checker to the Russian Cards Problem. One of its first logical analyses is [10] and the problem has since

gained notable attention as an intuitive example of information-theoretically (in contrast to computationally) secure cryptography [9,14].

The basic version of the problem is this: Seven cards, enumerated from 0 to 6, are distributed between Anne, Bob and Crow such that Anne and Bob both receive three cards and Crow one card. It is common knowledge which cards exist and how many cards each agent has. Everyone knows their own but not the others' cards. The goal of Anne and Bob now is to learn each others cards without Crow learning them. They can only communicate via public announcements.

Many different solutions exist but here we will focus on the so-called five-hands protocols (and their extensions with six or seven hands): First Anne makes an announcement of the form "My hand is one of these: ...". If her hand is 012 she could for example take the set $\{012, 034, 056, 135, 146, 236\}$. It can be checked that this announcement does not tell Crow anything, independent of which card it has. In contrast, Bob will be able to rule out all but one of the hands in the list depending on his own hand. Hence the second and last step of the protocol is an announcement by Bob about which card Crow has. For example, if Bob's hand is 345 he would finish the protocol with "Crow has card 6.".

Verifying this protocol for the fixed deal 012|345|6 with our symbolic model checker takes less than a second. Moreover, checking multiple protocols in a row does not take much longer because the BDD package caches results. Compared to that, a DEMO implementation [12] needs 4 seconds to check one protocol.

We can not just verify but also find all 5/6/7-hands protocols, using a combination of manual reasoning and brute-force. By Proposition 32 in [10] safe announcements from Anne never contain "crossing" hands, i.e. two hands with multiple card in common. If we also assume that the hands are lexicographically ordered, this leaves us with 1290 possible lists of five, six or seven hands of three cards. Only some of them are safe announcements which can be used by Anne. We can find them by checking all the corresponding 1290 formulas. Our model checker can filter out the 102 safe announcements within 1.6 seconds, generating and verifying the same list as in [10] where it was manually generated.

7 Equivalence of S5 Kripke Models and Knowledge Structures

Having shown the computational advantage of our new knowledge models, we now look more deeply into the foundations of what we have been doing. For a start, we show that knowledge structures and standard models for DEL are equivalent from a semantic point of view. Lemma 1 gives us a canonical way to show that a knowledge structure and an S5 Kripke model satisfy the same formulas. Theorems 2 and 3 say that such equivalent models and structures can always be found. These translations are also implemented in SMCDEL.

Lemma 1. *Suppose we have a knowledge structure* $\mathcal{F} = (V', \theta, O_1, \cdots, O_n)$ *and a finite S5 Kripke model* $M = (W, \pi, \mathcal{K}_1, \cdots, \mathcal{K}_n)$ *with a set of primitive propositions* $V \subseteq V'$*. Furthermore, suppose we have a function* $g : W \to \mathcal{P}(V')$ *such that*

C1 For all $w_1, w_2 \in W$*, and all* i *such that* $1 \leq i \leq n$*, we have that* $g(w_1) \cap O_i = g(w_2) \cap O_i$ *iff* $w_1 \mathcal{K}_i w_2$.
C2 For all $w \in W$ *and* $v \in V$*, we have that* $v \in g(w)$ *iff* $\pi(w)(v) = \mathbf{true}$.
C3 For every $s \subseteq V'$*,* s *is a state of* $\mathcal{F}$ *iff* $s = g(w)$ *for some* $w \in W$.

Then, for every formula φ *over* V *we have* $(\mathcal{F}, g(w)) \models \varphi$ *iff* $(M, w) \models \varphi$.

Proof. By induction on φ: Use C2 for atomic propositions, note that the boolean semantics are the same, use C1 and C3 for the knowledge operator and show that the conditions carry over to the results of announcements.

We do not give details here because the proof does not provide any new insights: Conditions C1 to C3 describe a special case of a p-morphism between M and the Kripke model encoded by $\mathcal{F}$, see Definition 7 below. Hence their equivalence with respect to the modal language already follows from general invariance results in modal logic [4, §2.1]. The following definition and theorem show that for every knowledge structure there is an equivalent Kripke model.

Definition 7. *For any* $\mathcal{F} = (V, \theta, O_1, \cdots, O_n)$*, we define the Kripke model* $M(\mathcal{F}) := (W, \pi, \mathcal{K}_1, \cdots, \mathcal{K}_n)$ *as follows*

1. W *is the set of all states of* $\mathcal{F}$,
2. *for each* $w \in W$*, let the assignment* $\pi(w)$ *be* w *itself and*
3. *for each agent* i *and all* $w, w' \in W$*, let* $w \mathcal{K}_i w'$ *iff* $w \cap O_i = w' \cap O_i$.

Theorem 2. *For any knowledge structure* $\mathcal{F}$*, any state* s *of* $\mathcal{F}$*, and any* φ *we have* $(\mathcal{F}, s) \models \varphi$ *iff* $(M(\mathcal{F}), s) \models \varphi$.

Proof. By Lemma 1 using the identity function as g.

Vice versa, for any S5 Kripke model we can find an equivalent knowledge structure. The essential idea is to add propositions as observational variables to encode the relations of each agent. To obtain a simple knowledge structure we should add as few propositions as possible. The method below adds $\sum_{i \in I} \mathsf{ceiling}(\log_2 k_i)$ propositions where k_i is the number of $\mathcal{K}_i$-equivalence classes and $\mathsf{ceiling}(\cdot)$ denotes the smallest integer not less than the argument. This could be further improved if one were to find a general way of using the propositions already present in the Kripke model as observational variables directly.

Definition 8. *For any S5 model* $M = (W, \pi, \mathcal{K}_1, \cdots, \mathcal{K}_n)$ *we define a knowledge structure* $\mathcal{F}(M)$ *as follows. For each* i*, write* $\gamma_1, \ldots, \gamma_{k_i}$ *for the equivalence classes given by* $\mathcal{K}_i$ *and let* $l_i := \mathsf{ceiling}(\log_2 k_i)$*. Let* O_i *be a set of* l_i *many fresh propositions. This yields the sets of observational variables* $O_1, \ldots, O_n$*, all disjoint to each other. If agent* i *has a total relation, i.e. only one equivalence class, then we have* $O_i = \varnothing$*. Enumerate* k_i *many subsets of* O_i *as* $O_{\gamma_1}, \ldots, O_{\gamma_{k_i}}$ *and define the function* $g_i : W \to \mathcal{P}(O_i)$ *by* $g_i(w) := O_{\gamma(w)}$ *where* $\gamma(w)$ *is the equivalence class of* w*. Let* $V' := V \cup \bigcup_{0<i\leq n} O_i$ *and define* $g : W \to \mathcal{P}(V')$ *by*

$$g(w) := \{v \in V \mid \pi(w)(v) = \top\} \cup \bigcup_{0<i\leq n} g_i(w)$$

Let $\overline{V'}$ *be the set of atomic propositions and their negations from* V'*. Finally, let* $\mathcal{F}(M) := (V', \theta_M, O_1, \ldots, O_n)$ *where*

$$\theta_M := \bigwedge \left\{ \bigvee Q \mid \ Q \subseteq \overline{V'} \text{ and } g(w) \models \bigvee Q \text{ for all } w \in W \right\}$$

Theorem 3. *For any finite S5 pointed Kripke model* (M, w) *and every formula* φ*, we have that* $(M, w) \models \varphi$ *iff* $(\mathcal{F}(M), g(w)) \models \varphi$*.*

Proof. By Definition 8, g_i is such that for all $w_1, w_2 \in W$, $g_i(w_1)$ and $g_i(w_2)$ are the same subset of O_i iff w_1 and w_2 are in the same equivalence class of $\mathcal{K}_i$. It is therefore easy to check the first two conditions of Lemma 1. For the "if" part of C3: If $s = g(w')$ for some $w' \in W$, then by the definition of θ_M, we have that $g(w') \models \theta_M$ and hence $g(w')$ is a state of $\mathcal{F}(M)$. For the "only if" part, assume that for every $w \in W$, $s \neq g(w)$. Then, for every $w \in W$, there is an atomic formula φ_w over V' such that $s \models \varphi_w$ but $g(w) \models \neg\varphi_w$. Therefore, $s \models \bigwedge_{w \in W} \varphi_w$. Moreover, we have for every $w' \in W$, $g(w') \models \bigvee_{w \in W} \neg\varphi_w$, and hence $\bigvee_{w \in W} \neg\varphi_w \in \Gamma_M$. Consequently, we have $s \not\models \Gamma_M$ and hence s is not a state of $\mathcal{F}(M)$. Now the theorem follows from Lemma 1.

What we have seen is how the two ways of modeling in this paper, though computationally different, are semantically equivalent. This leads us to consider how their interplay will work in more complex settings. The obvious direction to probe this is the area where DEL unleashes its full power: We now give an outlook how knowledge structures can be generalized to action models. They were first described in [1] and we do not repeat definitions here but refer to [11] for a textbook treatment. What action models are to Kripke frames, the following knowledge transformers are to knowledge structures.

Definition 9. *A* knowledge transformer *for a given vocabulary* V *is a tuple* $\mathcal{X} = (V^+, \theta^+, O_1, \ldots, O_n)$ *where* V^+ *is a set of atomic propositions such that* $V \cap V^+ = \varnothing$*,* θ^+ *is a possibly epistemic formula over* $V \cup V^+$ *and* $O_i \subseteq V^+$ *for all agents* i*. An* event *is a knowledge transformer together with a subset* $x \subseteq V^+$*, written as* $(\mathcal{X}, x)$*.*

The knowledge transformation *of a knowledge structure* $\mathcal{F} = (V, \theta, O_1, \ldots, O_n)$ *with a knowledge transformer* $\mathcal{X} = (V^+, \theta^+, O_1^+, \ldots, O_n^+)$ *for* V *is defined by:*

$$\mathcal{F}^{\mathcal{X}} := (V \cup V^+, \theta \wedge ||\theta^+||_{\mathcal{F}}, O_1 \cup O_1^+, \ldots, O_n \cup O_n^+)$$

Given a scene $(\mathcal{F}, s)$ *and an event* $(\mathcal{X}, x)$ *we define* $(\mathcal{F}, s)^{(\mathcal{X}, x)} := (\mathcal{F}^{\mathcal{X}}, s \cup x)$*.*

The two kinds of events discussed above fit well into this general definition: The public announcement of φ is the event $((\varnothing, \varphi, \varnothing, \ldots, \varnothing), \varnothing)$ and the announcement of φ to Δ is given by $((\{p_\varphi\}, p_\varphi \to \varphi, O_1^+, \ldots, O_n^+), \{p_\varphi\})$ where $O_i^+ = \{p_\varphi\}$ if $i \in \Delta$ and $O_i^+ = \varnothing$ otherwise.

Theorem 4. *For any S5 action model there is an equivalent knowledge transformer and vice versa.*

Proof. Define translations similar to Definitions 7 and 8. Then use Lemma 1.

Finally, Definition 6 can be extended to cover event operators: Let $\|[\mathcal{X}, x]\varphi\|_{\mathcal{F}} := \|\theta_x^+\|_{\mathcal{F}} \to \|\varphi'\|_{\mathcal{F}^{\mathcal{X}}}$ where $\theta_x^+ := \theta^+ \left(\frac{x}{\top}\right) \left(\frac{V^+ \setminus x}{\bot}\right)$ and $\varphi' := \varphi \left(\frac{x}{\top}\right) \left(\frac{V^+ \setminus x}{\bot}\right)$.

8 Conclusion and Future Work

We have achieved our goal of putting a new engine into DEL by a suitable semantic model transformation. This was shown to work well in various benchmarks, for example the Muddy Children and Russian cards. But there is obviously more to be explored now that we know this. In future work we aim to extend our theoretical framework and the implementation in different directions.

One line would be to use the same models with richer languages, and see whether the parallels that we found still persist. For example, action models with factual change [2] should also be representable as knowledge transformers. They also motivate a new notion of action equivalence which might help to solve a problem with action models where bisimulation had to be replaced with the more complicated notion of action emulation [16].

Another direction would be to extend the framework to other dynamic phenomena such as belief change or preference change which are usually non-S5. For this we can use the literature on abstraction for transition systems, starting with the seminal [8]. Moreover, BDDs have already been used to model belief change in [21]. Also abstraction ideas from the DEL literature could be implemented and their performance compared, for example the very compact modeling of Muddy Children in [20] and the mental programs from [6].

But perhaps the deepest issue that we see emerging in our approach is this. While standard logical approaches to information flow assume a sharp distinction between syntax and semantic models, our BDD-oriented approach suggests the existence of a third intermediate level of representation combining features of both that may be the right level to be at, also from a cognitive viewpoint. We leave the exploration of the latter grander program to another occasion.

Acknowledgements. This work was partially supported by NSFC grant 61472369 and carried out within the Tsinghua-UvA Joint Research Center in Logic. We thank our anonymous referees for useful comments and suggestions.

References

1. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: Bilboa, I. (ed.) TARK 1998, pp. 43–56 (1998)
2. van Benthem, J., van Eijck, J., Kooi, B.: Logics of communication and change. Information and Computation 204(11), 1620–1662 (2006)
3. van Benthem, J., Gerbrandy, J., Hoshi, T., Pacuit, E.: Merging frameworks for interaction. Journal of Philosophical Logic 38(5), 491–526 (2009)
4. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. In: Cambridge Tracts in Theoretical Computer Science, no. 53. CUP, Cambridge (2001)
5. Bryant, R.E.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Transaction on Computers C-35(8), 677–691 (1986)
6. Charrier, T., Schwarzentruber, F.: Arbitrary public announcement logic with mental programs. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pp. 1471–1479. IFAAMAS (2015)

7. Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. Journal of Cryptology 1(1), 65–75 (1988)
8. Clarke, E.M., Grumberg, O., Long, D.E.: Model checking and abstraction. ACM Transactions on Programming Languages and Systems 16(5), 1512–1542 (1994)
9. Cordón-Franco, A., van Ditmarsch, H., Fernández-Duque, D., Soler-Toscano, F.: A geometric protocol for cryptography with cards. Designs, Codes and Cryptography 74(1), 113–125 (2015), http://dx.doi.org/10.1007/s10623-013-9855-y
10. van Ditmarsch, H.: The russian cards problem. Studia Logica 75(1), 31–62 (2003)
11. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic epistemic logic, vol. 1. Springer, Heidelberg (2007)
12. van Ditmarsch, H., van der Hoek, W., van der Meyden, R., Ruan, J.: Model Checking Russian Cards. Electr. Notes Theor. Comput. Sci. 149(2), 105–123 (2006)
13. van Ditmarsch, H., van der Hoek, W., Ruan, J.: Connecting dynamic epistemic and temporal epistemic logics. Logic Journal of IGPL 21(3), 380–403 (2013)
14. Duque, D.F., Goranko, V.: Secure aggregation of distributed information. CoRR abs/1407.7582 (2014), http://arxiv.org/abs/1407.7582
15. van Eijck, J.: DEMO-S5. Tech. rep., CWI (2014)
16. van Eijck, J., Ruan, J., Sadzik, T.: Action emulation. Synthese 185(1), 131–151 (2012)
17. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about knowledge, vol. 4. MIT Press, Cambridge (1995)
18. Gammie, P.: hBDD. https://github.com/peteg/hBDD (2011, updated 2014)
19. Gattinger, M.: HasCacBDD (2015), https://github.com/m4lvin/HasCacBDD
20. Gierasimczuk, N., Szymanik, J.: A note on a generalization of the Muddy Children puzzle. In: Apt, K.R. (ed.) TARK 2011, pp. 257–264. ACM (2011)
21. Gorogiannis, N., Ryan, M.D.: Implementation of Belief Change Operators Using BDDs. Studia Logica 70(1), 131–156 (2002)
22. Knuth, D.E.: The Art of Computer Programming. Combinatorial Algorithms, Part 1, vol. 4A. Addison-Wesley Professional (2011)
23. Littlewood, J.: A Mathematician's Miscellany. Methuen, London (1953)
24. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: an open-source model checker for the verification of multi-agent systems. International Journal on Software Tools for Technology Transfer, 1–22 (2015)
25. Lomuscio, A.R., van der Meyden, R., Ryan, M.: Knowledge in Multiagent Systems: Initial Configurations and Broadcast. ACM Trans. Comp. L. 1(2), 247–284 (2000)
26. Luo, X., Su, K., Sattar, A., Chen, Y.: Solving Sum and Product Riddle via BDD-Based Model Checking. In: Web Intel./IAT Workshops, pp. 630–633. IEEE (2008)
27. Lv, G., Su, K., Xu, Y.: CacBDD: A BDD Package with Dynamic Cache Management. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 229–234. Springer, Heidelberg (2013)
28. van der Meyden, R., Su, K.: Symbolic Model Checking the Knowledge of the Dining Cryptographers. In: CSFW, pp. 280–291. IEEE Computer Society (2004)
29. Somenzi, F.: CUDD: CU Decision Diagram Package Release 2.5.0 (2012)
30. Su, K., Sattar, A., Luo, X.: Model Checking Temporal Logics of Knowledge Via OBDDs. The Computer Journal 50(4), 403–420 (2007)