

Parameterized Complexity Results for a Model of Theory of Mind based on Dynamic Epistemic Logic*

Iris van de Pol[†]
Institute for Logic, Language,
and Computation
University of Amsterdam
i.p.a.vandepol@uva.com

Iris van Rooij
Donders Institute for Brain,
Cognition, and Behaviour
Radboud University
i.vanrooij@donders.ru.nl

Jakub Szymanik[‡]
Institute for Logic, Language,
and Computation
University of Amsterdam
j.k.szymanik@uva.nl

ABSTRACT

In this paper we introduce a computational-level model of theory of mind (ToM) based on dynamic epistemic logic (DEL), and we analyze its computational complexity. The model is a special case of DEL model checking. We provide a parameterized complexity analysis, considering several aspects of DEL (e.g., number of agents, size of preconditions, etc.) as parameters. We show that model checking for DEL is PSPACE-hard, also when restricted to single-pointed models and S5 relations, thereby solving an open problem in the literature. Our approach is aimed at formalizing current intractability claims in the cognitive science literature regarding computational models of ToM.

Categories and Subject Descriptors

F.1.3 [Theory of Computation]: Complexity Measures & Classes—*Reducibility and Completeness*; F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic—*Modal Logic*; I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods—*Modal Logic*

General Terms

Theory

Keywords

Theory of mind; dynamic epistemic logic; computational complexity; parameterized complexity; computational-level model

*This research has been carried out in the context of the first author's master's thesis [36].

[†]Supported by Gravitation Grant 024.001.006 of the Language in Interaction Consortium from the Netherlands Organisation for Scientific Research.

[‡]Supported by the Netherlands Organisation for Scientific Research Veni Grant NWO-639-021-232.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

1. INTRODUCTION

Imagine that you are in love. You find yourself at your desk, but you cannot stop your mind from wandering off. What is she thinking about right now? And more importantly, is she thinking about you and does she know that you are thinking about her? Reasoning about other people's knowledge, belief and desires, we do it all the time. For instance, in trying to conquer the love of one's life, to stay one step ahead of one's enemies, or when we lose our friend in a crowded place and we find them by imagining where they would look for us. This capacity is known as theory of mind (ToM) and it is widely studied in various fields (see, e.g., [8, 10, 22, 33, 35, 37, 46, 47]).

We seem to use ToM on a daily basis and many cognitive scientists consider it to be ubiquitous in social interaction [1]. At the same time, however, it is also widely believed that computational cognitive models of ToM are intractable, i.e., that ToM involves solving problems that humans are not capable of solving (cf. [1, 26, 30, 49]). This seems to imply a contradiction between theory and practice: on the one hand we seem to be capable of ToM, while on the other hand, our theories tell us that this is impossible. Dissolving this paradox is a critical step in enhancing theoretical understanding of ToM.

The question arises what it means for a computational-level model¹ of cognition to be intractable. When looking more closely at these intractability claims regarding ToM, it is not clear what these researchers mean exactly, nor whether they mean the same thing. In theoretical computer science and logic there are a variety of tools to make precise claims about the level of complexity of a certain problem. In cognitive science, however, this is a different story. With the exception of a few researchers, cognitive scientists do not tend to specify formally what it means for a theory to be intractable. This makes it often very difficult to assess the validity of the various claims in the literature about which theories are tractable and which are not.

In this paper we adopt the *Tractable-cognition thesis* (see [41]) that states that people have limited resources

¹In cognitive science, often Marr's [32] tri-level distinction between computational-level ("what is the nature of the problem being solved?"), algorithmic-level ("what is the algorithm used for solving the problem?"), and implementational-level ("how is the algorithm physically realized?") is used to distinguish different levels of computational cognitive explanations. In this paper, we will focus on computational-level models of ToM and their computational complexity.

for cognitive processing and human cognitive capacities are confined to those that can be realized using a realistic amount of time.² More specifically we adopt the *FPT-cognition thesis* [41] that states that computationally plausible computational-level cognitive theories are limited to the class of input-output mappings that are fixed-parameter tractable for one or more input-parameters that can be assumed to be small in practice. To be able to make more precise claims about the (in)tractability of ToM we introduce a computational-level model of ToM based on dynamic epistemic logic (DEL), and we analyze its computational complexity. The model we present is a special case of DEL model checking. Here we include an informal description of the model.³ The kind of situation that we want to be able to model, is that of an observer that observes one or more agents in an initial situation. The observer then witnesses actions that change the situation and the observer updates their knowledge about the mental states of the agents in the new situation. Such a set up is often found in experimental tasks, where subjects are asked to reason about the mental states of agents in a situation that they are presented.

DBU (informal) – DYNAMIC BELIEF UPDATE

Instance: A representation of an initial situation, a sequence of actions – observed by an observer – and a (belief) statement φ of interest.

Question: Is the (belief) statement φ true in the situation resulting from the initial situation and the observed actions?

We prove that DBU is PSPACE-complete. PSPACE-completeness was already shown by Aucher and Schwarzen- truber [3] for DEL model checking in general. They considered unrestricted relations and multi-pointed event models. Since their proof does not hold for the special case of DEL model checking that we consider, we propose an alternative proof. Our proof solves positively the open question in [3] whether model checking for DEL restricted to S5 relations and single-pointed models is PSPACE-complete. Bolander, Jensen and Schwarzen- truber [?] independently considered an almost identical special case of DEL model checking (there called the plan verification problem). They also prove PSPACE-completeness for the case restricted to single-pointed models, but their proof does not settle whether hardness holds even when the problem is restricted to S5 models.

Furthermore, we investigate how the different aspects (or parameters, see Table 1) of our model influence its complexity. We prove that for most combinations of param-

²There is general consensus in the cognitive science community that computational intractability is a undesirable feature of cognitive computational models, putting the cognitive plausibility of such models into question [12, 23, 25, 41, 45]. There are diverging opinions about how cognitive science should deal with this issue (see, e.g., [11, 25, 40, 42]). It is beyond the scope of this paper to discuss this in detail. In this paper we adopt the parameterized complexity approach as described in [41].

³We pose the model in the form of a decision problem, as this is convenient for purposes of our complexity analysis. Even though ToM may be more intuitively modeled by a search problem, the complexity of the decision problem gives us lower bounds on the complexity of such a search problem, and therefore suffices for the purposes of our paper.

eters DBU is fp-intractable and for one case we prove fp-tractability. See Figure 2 for an overview of the results.

Besides the parameterized complexity results for DEL model checking that we present, the main conceptual contribution of this paper is that it bridges cognitive science and logic, by using DEL to model ToM (cf. [27, 46]). By doing so, the paper provides the means to make more precise statements about the (in)tractability of ToM.

The paper is structured as follows. In Section 2 we introduce basic definitions from dynamic epistemic logic and parameterized complexity theory. Then, in Section 3 we introduce a formal description of our computational-level model and we discuss the particular choices that we make. Next, in Section 4 we present our (parameterized) complexity results. Finally, in Section 5 we discuss the implications of our results for the understanding of ToM.

2. PRELIMINARIES

2.1 Dynamic Epistemic Logic

Dynamic epistemic logic is a particular kind of modal logic (see [15, 6]), where the modal operators are interpreted in terms of belief or knowledge. First, we define epistemic models, which are Kripke models with an accessibility relation for every agent $a \in \mathcal{A}$, instead of just one accessibility relation.

DEFINITION 2.1 (Epistemic model). *Given a finite set \mathcal{A} of agents and a finite set P of propositions, an epistemic model is a tuple (W, R, V) where*

- W is a non-empty set of worlds;
- R is a function that assigns to every agent $a \in \mathcal{A}$ a binary relation R_a on W ; and
- V is a valuation function from $W \times P$ into $\{0, 1\}$.

The accessibility relations R_a can be read as follows: for worlds $w, v \in W$, wR_av means “in world w , agent a considers world v possible.”

DEFINITION 2.2 ((Multi and single-)pointed epistemic model). *A pair (M, W_d) consisting of an epistemic model $M = (W, R, V)$ and a non-empty set of designated worlds $W_d \subseteq W$ is called a pointed epistemic model. A pair (M, W_d) is called a single-pointed model when W_d is a singleton, and a multi-pointed epistemic model when $|W_d| > 1$. By a slight abuse of notation, for $(M, \{w\})$, we also write (M, w) .*

We consider the usual restrictions on relations in epistemic models and event models, such as KD45 and S5 (see [15]). In KD45 models, all relations are transitive, Euclidean and serial, and in S5 models all relations are transitive, reflexive and symmetric.

We define the following language for epistemic models. We use the modal belief operator B , where for each agent $a \in \mathcal{A}$, $B_a\varphi$ is interpreted as “agent a believes (that) φ ”.

DEFINITION 2.3 (Epistemic language). *The language \mathcal{L}_B over \mathcal{A} and P is given by the following definition, where a ranges over \mathcal{A} and p over P :*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid B_a\varphi.$$

We will use the following standard abbreviations, $\top := p \vee \neg p$, $\perp := \neg \top$, $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \rightarrow \psi := \neg\varphi \vee \psi$, $\hat{B}_a := \neg B_a \neg\varphi$.

The semantics for this language is defined as follows.

DEFINITION 2.4 (Truth in a (single-pointed) epistemic model). *Let $M = (W, R, V)$ be an epistemic model, $w \in W$, $a \in \mathcal{A}$, and $\varphi, \psi \in \mathcal{L}_B$. We define $M, w \models \varphi$ inductively as follows:*

$$\begin{aligned} M, w \models p & \quad \text{iff} \quad V(w, p) = 1 \\ M, w \models \neg\varphi & \quad \text{iff} \quad \text{not } M, w \models \varphi \\ M, w \models (\varphi \wedge \psi) & \quad \text{iff} \quad M, w \models \varphi \text{ and } M, w \models \psi \\ M, w \models B_a\varphi & \quad \text{iff} \quad \text{for all } v \text{ with } wR_av: M, v \models \varphi \end{aligned}$$

When $M, w \models \varphi$, we say that φ is true in w or φ is satisfied in w .

DEFINITION 2.5 (Truth in a multi-pointed epistemic model). *Let (M, W_d) be a multi-pointed epistemic model, $a \in \mathcal{A}$, and $\varphi \in \mathcal{L}_B$. $M, W_d \models \varphi$ is defined as follows:*

$$M, W_d \models \varphi \quad \text{iff} \quad M, w \models \varphi \text{ for all } w \in W_d$$

Next we define event models.

DEFINITION 2.6 (Event model). *An event model is a tuple $\mathcal{E} = (E, Q, \text{pre}, \text{post})$, where E is a non-empty finite set of events; Q is a function that assigns to every agent $a \in \mathcal{A}$ a binary relation R_a on W ; pre is a function from E into \mathcal{L}_B that assigns to each event a precondition, which can be any formula in \mathcal{L}_B ; and post is a function from E into \mathcal{L}_B that assigns to each event a postcondition. Postconditions are conjunctions of propositions and their negations (including \top and \perp).*

DEFINITION 2.7 ((Multi and single-)pointed event model / action). *A pair (\mathcal{E}, E_d) consisting of an event model $\mathcal{E} = (E, Q, \text{pre}, \text{post})$ and a non-empty set of designated events $E_d \subseteq E$ is called a pointed event model. A pair (\mathcal{E}, E_d) is called a single-pointed event model when E_d is a singleton, and a multi-pointed event model when $|E_d| > 1$. We will also refer to (\mathcal{E}, E_d) as an action.*

We define the notion of a product update, that is used to update epistemic models with actions [4].

DEFINITION 2.8 (Product update). *The product update of the state (M, W_d) with the action (\mathcal{E}, E_d) is defined as the state $(M, W_d) \otimes (\mathcal{E}, E_d) = ((W', R', V'), W'_d)$ where*

- $W' = \{(w, e) \in W \times E; M, w \models \text{pre}(e)\}$;
- $R'_a = \{((w, e), (v, f)) \in W' \times W'; wR_av \text{ and } eQ_af\}$;
- $V'(p) = 1$ iff either $(M, w \models p \text{ and } \text{post}(e) \neq \neg p)$ or $\text{post}(e) \models p$; and
- $W'_d = \{(w, e) \in W'; w \in W_d \text{ and } e \in E_d\}$.

Finally, we define when actions are applicable in a state.

DEFINITION 2.9 (Applicability). *An action (\mathcal{E}, E_d) is applicable in state (M, W_d) if there is some $e \in E_d$ and some $w \in W_d$ such that $M, w \models \text{pre}(e)$. We define applicability for a sequence of actions inductively. The empty sequence, consisting of no actions, is always applicable. A sequence a_1, \dots, a_k of actions is applicable in a state (M, W_d) if (1) the sequence a_1, \dots, a_{k-1} is applicable in (M, W_d) and (2) the action a_k is applicable in the state $(M, W_d) \otimes a_1 \otimes \dots \otimes a_{k-1}$.*

2.2 Parameterized Complexity Theory

We introduce some basic concepts of parameterized complexity theory. For a more detailed introduction we refer to textbooks on the topic [16, 17, 21, 34].

DEFINITION 2.10 (Parameterized problem). *Let Σ be a finite alphabet. A parameterized problem L (over Σ) is a subset of $\Sigma^* \times \mathbb{N}$. For an instance (x, k) , we call x the main part and k the parameter.*

The complexity class FPT, which stands for fixed-parameter tractable, is the direct analogue of the class P in classical complexity. Problems in this class are considered efficiently solvable, because the non-polynomial-time complexity inherent in the problem is confined to the parameter and in effect the problem is efficiently solvable even for large input sizes, provided that the value of the parameter is relatively small.

DEFINITION 2.11 (Fixed-parameter tractable / the class FPT). *Let Σ be a finite alphabet.*

1. *An algorithm A with input $(x, k) \in \Sigma \times \mathbb{N}$ runs in fpt-time if there exists a computable function f and a polynomial p such that for all $(x, k) \in \Sigma \times \mathbb{N}$, the running time of A on (x, k) is at most*

$$f(k) \cdot p(|x|).$$

Algorithms that run in fpt-time are called fpt-algorithms.

2. *A parameterized problem L is fixed-parameter tractable if there is an fpt-algorithm that decides L . FPT denotes the class of all fixed-parameter tractable problems.*

Similarly to classical complexity, parameterized complexity also offers a hardness framework to give evidence that (parameterized) problems are not fixed-parameter tractable. The following notion of reductions plays an important role in this framework.

DEFINITION 2.12 (Fpt-reduction). *Let $L \subseteq \Sigma \times \mathbb{N}$ and $L' \subseteq \Sigma' \times \mathbb{N}$ be two parameterized problems. An fpt-reduction from L to L' is a mapping $R : \Sigma \times \mathbb{N} \rightarrow \Sigma' \times \mathbb{N}$ from instances of L to instances of L' such that there is a computable function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $(x, k) \in \Sigma \times \mathbb{N}$:*

1. *$(x', k') = R(x, k)$ is a yes-instance of L' if and only if (x, k) is a yes-instance of L ;*
2. *R is computable in fpt-time; and*
3. *$k' \leq g(k)$.*

Another important part of the hardness framework is the parameterized intractability class W[1]. To characterize this class, we consider the following parameterized problem.

{k}-WSAT[2CNF]

Instance: A 2CNF propositional formula φ and an integer k .

Parameter: k .

Question: Is there an assignment $\alpha : \text{var}(\varphi) \rightarrow \{0, 1\}$, that sets k variables in $\text{var}(\varphi)$ to true, that satisfies φ ?

The class $W[1]$ consists of all parameterized problems that can be fpt-reduced to $\{k\}$ -WSAT[2CNF]. A parameterized problem is hard for $W[1]$ if all problems in $W[1]$ can be fpt-reduced to it. It is widely believed that $W[1]$ -hard problems are not fixed-parameter tractable [17]. Another parameterized intractability class, that can be used in a similar way, is the class para-NP. The class para-NP consists of all parameterized problems that can be solved by a non-deterministic fpt-algorithm. To show para-NP-hardness, it suffices to show that DBU is NP-hard for a constant value of the parameters [20]. Problems that are para-NP-hard are not fixed-parameter tractable, unless $P = NP$ [21, Theorem 2.14].

3. COMPUTATIONAL-LEVEL MODEL OF THEORY OF MIND

Next we present a formal description of our computational-level model. Our aim is to capture, in a qualitative way, the kind of reasoning that is necessary to be able to engage in ToM. Arguably, the essence of ToM is the attribution of mental states to another person, based on observed behavior, and to predict and explain this behavior in terms of those mental states. The aspect of ToM that we aim to formalize with our model is the attribution of mental states. There is a wide range of different kinds of mental states such as epistemic, emotional and motivational states. In our model we focus on epistemic states, in particular on belief.

To be cognitively plausible, our model needs to be able to capture a wide range of (dynamic) situations, where all kinds of actions can occur, not just actions that change beliefs (epistemic actions), but also actions that change the state of the world (ontic actions). This is why, following Bolander and Andersen [9], we use postconditions in the product update of DEL (in addition to preconditions).

Furthermore, we want to model the (internal) perspective of the observer (on the situation). Therefore, the god perspective, also called the perfect external approach by Aucher [2] – that is inherent to single-pointed epistemic models – will not suffice for all cases that we want to be able to model. This perfect external approach supposes that the modeler is an omniscient observer that is perfectly aware of the actual state of the world and the epistemic situation (what is going on in the minds of the agents). The cognitively plausible observers that we are interested in here will not have infallible knowledge in many situations. They are often not able to distinguish the actual world from other possible worlds, because they are uncertain about the facts in the world and the mental states of the agent(s) that they observe. That is why, again following Bolander and Andersen [9], we allow for multi-pointed epistemic models (in addition to single-pointed models), which can model the uncertainty of an observer, by representing their perspective as a set of worlds. How to represent the internal or fallible perspective of an agent in epistemic models is a conceptual problem that has not been settled yet in the DEL-literature. There have been several proposals to deal with this (see, e.g., [2, 14, 24]).

Also, since we do not assume that agents are perfectly knowledgeable, we allow the possibility of modeling false beliefs of the observers and agents, by using KD45 models (rather than S5 models). Even though KD45 models present an idealized form of belief (with perfect introspec-

tion and logical omniscience), we argue that at least to some extent they are cognitively plausible, and that therefore, for the purpose of this paper, it suffices to focus on KD45 models. Our complexity results (which we present in the next section) do not depend on this choice; they hold for DBU restricted to KD45 models and restricted to S5 models, and also for the unrestricted case.

We define our computational-level model of ToM as follows.

DBU (formal) – DYNAMIC BELIEF UPDATE
Instance: A set of propositions P , and set of Agents \mathcal{A} . An initial state s_o , where $s_o = ((W, V, R), W_d)$ is a pointed epistemic model. An applicable sequence of actions a_1, \dots, a_k , where $a_j = ((E, Q, pre, post), E_d)$ is a pointed event model. A formula $\varphi \in \mathcal{L}_B$.
Question: Does $s_o \otimes a_1 \otimes \dots \otimes a_k \models \varphi$?

The model can be naturally used to formalize ToM tasks that are employed in psychological experiments. The classical ToM task that is used by (developmental) psychologists is the false belief task [5, 48]. The DEL-based formalization of the false belief task by Bolander [8] can be seen as an instance of DBU. For more details on how DBU can be used to model ToM tasks, we refer to [36].

4. COMPLEXITY RESULTS

4.1 PSPACE-completeness

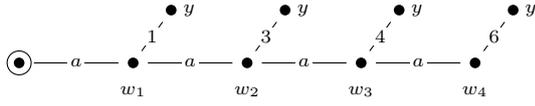
We show that DBU is PSPACE-complete. For this, we consider the decision problem TQBF. This problem is PSPACE-complete [44].

TQBF
Instance: A quantified Boolean formula $\varphi = Q_1x_1Q_2x_2 \dots Q_mx_m.\psi$.
Question: Is φ true?

THEOREM 1. DBU is PSPACE-hard.

PROOF. To show PSPACE-hardness we specify a polynomial-time reduction R from TQBF to DBU. Let ψ be a Boolean formula. First, we sketch the general idea behind the reduction. We use the reduction to list all possible assignments to $var(\psi)$. To do this we use groups of worlds (which are R_a -equivalence classes) to represent particular truth assignments. Each group consists of a string of worlds that are fully connected by equivalence relation R_a . Except for the first world in the string, all worlds represent a true variable x_i (under a particular assignment).

We give an example of such a group of worlds that represents assignment $\alpha = \{x_1 \mapsto T, x_2 \mapsto F, x_3 \mapsto T, x_4 \mapsto T, x_5 \mapsto F, x_6 \mapsto T\}$. Each world has a reflexive loop for every agent, which we leave out for the sake of presentation. More generally, in all our drawings we replace each relation R_a with a minimal R'_a whose transitive reflexive closure is equal to R_a . \odot marks the designated world. Since all relations are reflexive, we draw relations as lines (leaving out arrows at the end).



We refer to worlds w_1, \dots, w_4 as the *bottom worlds* of this group. If a bottom world has an R_i relation to a world that makes proposition y true, we say that it represents variable x_i .

The reduction makes sure that in the final updated model (the model that results from updating the initial state with the actions – which are specified by the reduction) each possible truth assignment to the variables in ψ will be represented by a group of worlds. Between the different groups, there are no R_a -relations (only R_i -relations for $1 \leq i \leq m$). By ‘jumping’ from one group (representing a particular truth assignment) to another group with relation R_i , the truth value of variable x_i can be set to true or false. We can now translate a quantified Boolean formula into a corresponding formula of \mathcal{L}_B by mapping every universal quantifier Q_i to \hat{B}_i and every existential quantifier Q_j to \hat{B}_j .

To illustrate how this reduction works, we give an example. Figure 1 shows the final updated model for a quantified Boolean formula with variables x_1 and x_2 . In this model there are four groups of worlds: $\{w_1, w_2, w_3\}$, $\{w_4, w_5\}$, $\{w_6, w_7\}$ and $\{w_8\}$. Worlds w_1, \dots, w_8 are what we refer to as the bottom worlds. The gray worlds and edges can be considered a byproduct of the reduction; they have no particular function.

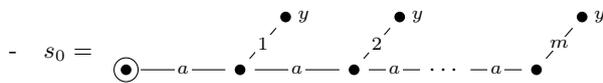
We represent variable x_1 by $\hat{B}_1 y$ and variable x_2 by $\hat{B}_2 y$. Then, in the model above, checking whether $\exists x_1 \forall x_2. x_1 \vee x_2$ is true can be done by checking whether formula $\hat{B}_1 B_2 (\hat{B}_a \hat{B}_1 y \vee \hat{B}_a \hat{B}_2 y)$ is true, which is indeed the case. Also, checking whether $\forall x_1 \forall x_2. x_1 \vee x_2$ is true can be done by checking whether $B_1 B_2 (\hat{B}_a \hat{B}_1 y \vee \hat{B}_a \hat{B}_2 y)$ is true, which is not the case.

Now, we continue with the formal details. Let $\varphi = Q_1 x_1 \dots Q_m x_m. \psi$ be a quantified Boolean formula with quantifiers Q_1, \dots, Q_m and $\text{var}(\psi) = \{x_1, \dots, x_m\}$. We define the following polynomial-time computable mappings. For $1 \leq i \leq m$, let $[x_i] = \hat{B}_i y$, and

$$[Q_i] = \begin{cases} B_i & \text{if } Q_i = \forall \\ \hat{B}_i & \text{if } Q_i = \exists \end{cases}$$

Formula $[\psi]$ is the adaptation of formula ψ where every occurrence of x_i in ψ is replaced by $\hat{B}_a [x_i]$. Then $[\varphi] = [Q_1] \dots [Q_m] [\psi]$. We formally specify the reduction R . We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_m, [\varphi])$, where:

$$- P = \{y\}, \mathcal{A} = \{a, 1, \dots, m\}$$



All relations in s_0, a_1, \dots, a_m are equivalence relations. Note that all worlds in s_0, a_1, \dots, a_m have reflexive loops for all agents. We omit all reflexive loops for the sake of readability.

$$- a_1 = \begin{matrix} \text{---} 1 \text{---} \\ e_1 : \langle \top, \top \rangle \quad e_2 : \langle \neg \hat{B}_1 y \vee y, \top \rangle \end{matrix}$$

$$\vdots$$

$$- a_m = \begin{matrix} \text{---} m \text{---} \\ e_1 : \langle \top, \top \rangle \quad e_2 : \langle \neg \hat{B}_m y \vee y, \top \rangle \end{matrix}$$

We show that $\varphi \in \text{TQBF}$ if and only if $R(\varphi) \in \text{DBU}$. We prove that for all $1 \leq i \leq m+1$ the following claim holds. For any assignment α to the variables x_1, \dots, x_{i-1} and any bottom world w of a group that agrees with α , the formula $Q_i x_i \dots Q_m x_m. \psi$ is true under α if and only if $[Q_i] \dots [Q_m] [\psi]$ is true in world w . In the case for $i = m+1$, this refers to the formula $[\psi]$.

We start with the case for $i = m+1$. We show that the claim holds. Let α be any assignment to the variables x_1, \dots, x_m , and let w be any bottom world of a group γ that represents α . Then, by construction of $[\psi]$, we know that ψ is true under α if and only if $[\psi]$ is true in w .

Assume that the claim holds for $i = j+1$. We show that then the claim also holds for $i = j$. Let α be any assignment to the variables x_1, \dots, x_{j-1} and let w be a bottom world of a group that agrees with α . We show that the formula $Q_j \dots Q_m. \psi$ is true under α if and only if $[Q_j] \dots [Q_m] [\psi]$ is true in w .

First, assume that $Q_j \dots Q_m. \psi$ is true under α . Consider the case where $Q_j = \forall$. Then for both assignments $\alpha' \supseteq \alpha$ to the variables x_1, \dots, x_j , formula $Q_{j+1} \dots Q_m. \psi$ is true under α' . Now, by assumption, we know that for any bottom world w' of a group that agrees with α – so in particular for all bottom worlds w' that are R_j -reachable from w – formula $[Q_{j+1}] \dots [Q_m] [\psi]$ is true in w' . Since $[Q_j] = B_j$, this means that $[Q_j] \dots [Q_m] [\psi]$ is true in w . The case where $Q_j = \exists$ is analogous.

Next, assume that $Q_j \dots Q_m. \psi$ is not true under α . Consider the case where $Q_j = \forall$. Then there is some assignment $\alpha' \supseteq \alpha$ to the variables x_1, \dots, x_j , such that $Q_{j+1} \dots Q_m. \psi$ is not true under α' . Now, by assumption, we know that for any bottom world w' of a group that agrees with α – so in particular for some bottom world w' that is R_j -reachable from w – formula $[Q_{j+1}] \dots [Q_m] [\psi]$ is not true in w' . Since $[Q_j] = B_j$, this means that $[Q_j] \dots [Q_m] [\psi]$ is not true in w . The case where $Q_j = \exists$ is analogous.

Hence, the claim holds for the case that $i = j$. Now, by induction, the claim holds for the case that $i = 1$, and hence it follows that $\varphi \in \text{TQBF}$ if and only if $R(\varphi) \in \text{DBU}$. Since this reduction runs in polynomial time, we can conclude that DBU is PSPACE-hard. \square

THEOREM 2. DBU is PSPACE-complete.

PROOF. In order to show PSPACE-membership for the problem DBU, we can modify the polynomial-space algorithm given by Aucher and Schwarzentruber [3]. Their algorithm works for the problem of checking whether a given (single-pointed) epistemic model makes a given DEL-formula true, where the formula contains event models that can be multi-pointed, but that have no postconditions. In order to make the algorithm work for multi-pointed epistemic models, we can simply call the algorithm several times, once for each of the designated worlds. Also, a modification is needed to deal with postconditions. The algorithm checks the truth of a formula by inductively calling itself for subformulas. In order to deal with postconditions, only the case where the formula is a propositional variable needs to be modified. This modification is rather straightforward. For more details, we refer to [36]. \square

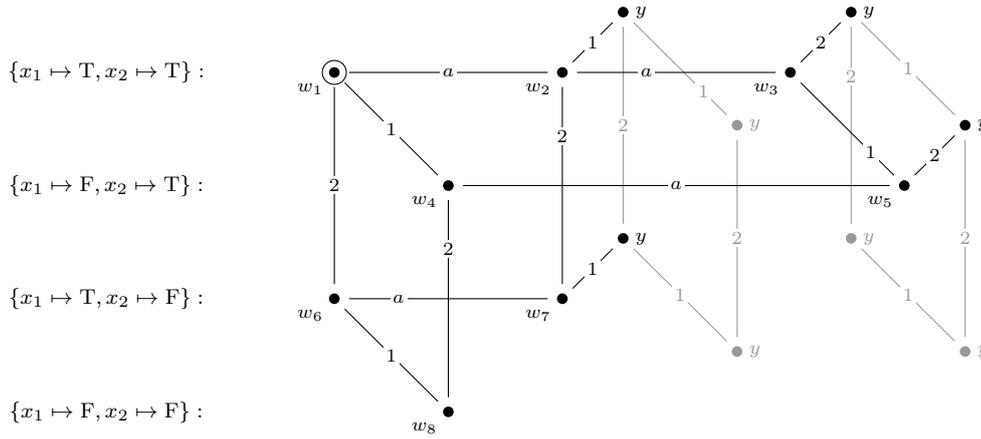


Figure 1: Example for the reduction in the proof of Theorem 1; a final updated model for a quantified Boolean formula with variables x_1 and x_2 .

4.2 Parameterized Complexity Results

Next, we provide a parameterized complexity analysis of DBU.

4.2.1 Parameters for DBU

We consider the following parameters for DBU. For each subset $\kappa \subseteq \{a, c, e, f, o, p, u\}$ we consider the parameterized variant κ -DBU of DBU, where the parameter is the sum of the values for the elements of κ as specified in Table 1. For instance, the problem $\{a\}$ -DBU is parameterized by the number of agents. Even though technically speaking there is only one parameter, we will refer to each of the elements of κ as parameters.

For the modal depth of a formula we count the maximum number of nested occurrences of operators B_a . Formally, we define the modal depth $d(\varphi)$ of a formula φ (in \mathcal{L}_B) recursively as follows.

$$d(\varphi) = \begin{cases} 0 & \text{if } \varphi = p \in P \text{ is a proposition;} \\ \max\{d(\varphi_1), d(\varphi_2)\} & \text{if } \varphi = \varphi_1 \wedge \varphi_2; \\ d(\varphi_1) & \text{if } \varphi = \neg\varphi_1; \\ 1 + d(\varphi_1) & \text{if } \varphi = B_a\varphi_1. \end{cases}$$

For the size of a formula we count the number of occurrences of propositions and logical connectives. Formally, we define the size $s(\varphi)$ of a formula φ (in \mathcal{L}_B) recursively as follows.

$$s(\varphi) = \begin{cases} 1 & \text{if } \varphi = p \in P \text{ is a proposition;} \\ 1 + s(\varphi_1) + s(\varphi_2) & \text{if } \varphi = \varphi_1 \wedge \varphi_2; \\ 1 + s(\varphi_1) & \text{if } \varphi = \neg\varphi_1; \\ 1 + s(\varphi_1) & \text{if } \varphi = B_a\varphi_1. \end{cases}$$

4.2.2 Intractability Results

In the following, we show fixed-parameter intractability for several parameterized versions of DBU. We will mainly use the parameterized complexity classes $W[1]$ and para-NP to show intractability, i.e., we will show hardness for these classes. Note that we could additionally use the class para-PSPACE [20] to give stronger intractability results. For instance, the proof of Theorem 1 already shows that $\{p\}$ -DBU

Param. Description

Param.	Description
a	number of agents
c	maximum size of the preconditions
e	maximum number of events in the event models
f	size of the formula
o	modal depth of the formula, i.e., the order parameter
p	number of propositions in P
u	number of actions, i.e., the number of updates

Table 1: Overview of the different parameters for DBU.

is para-PSPACE hard, since the reduction in this proof uses a constant number of propositions. However, since in this paper we are mainly interested in the border between fixed-parameter tractability and intractability, we will not focus on the subtle differences in the degree of intractability, and restrict ourselves to showing $W[1]$ -hardness and para-NP-hardness. This is also the reason why we will not show membership for any of the (parameterized) intractability classes; showing hardness suffices to indicate intractability. For the following proofs we use the well-known satisfiability problem SAT for propositional formulas. The problem SAT is NP-complete [13, 29]. Moreover, hardness for SAT holds even when restricted to propositional formulas that are in 3CNF.

PROPOSITION 3. $\{a, c, e, f, o\}$ -DBU is para-NP-hard.

PROOF. To show para-NP-hardness, we specify a polynomial-time reduction R from SAT to DBU, where parameters a, c, e, f , and o have constant values. Let φ be a propositional formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. Without loss of generality we assume that φ is a 3CNF formula with clauses c_1 to c_l .

The general idea behind this reduction is that we use the worlds in the final updated model (that results from updating the initial state with the actions – which are spec-

ified by the reduction) to list all possible assignments to $\text{var}(\varphi)$, by setting the propositions (corresponding to the variables in $\text{var}(\varphi)$) to true and false accordingly. Then checking whether formula φ is satisfiable can be done by checking whether φ is true in any of the worlds. Actions a_1 to a_m are used to create a corresponding world for each possible assignment to $\text{var}(\varphi)$. Furthermore, to keep the formula that we check in the final updated model of constant size, we sequentially check the truth of each clause c_i and encode whether the clauses are true with an additional variable x_{m+1} . This is done by actions a_{m+1} to a_{m+i} . In the final updated model, variable x_{m+1} will only be true in a world, if it makes clauses c_1 to c_i true, i.e., if it makes formula φ true.

For more details, we refer to [36]. \square

PROPOSITION 4. $\{c, e, f, o, p\}$ -DBU is para-NP-hard.

PROOF. To show para-NP-hardness, we specify a polynomial-time reduction R from SAT to DBU, where parameters c, e, f, o , and p have constant values. Let φ be a propositional formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. The general idea behind this reduction is similar to the reduction in the proof of Theorem 1. Again we use groups of worlds to represent particular assignments to the variables in φ . Here, there is only relation R_b between the different groups. Furthermore, to keep the formula that we check in the final updated model of constant size, we sequentially check the truth of each clause c_i and encode whether the clauses are true with an additional variable z . This is done by actions a_{m+1} to a_{m+i} . Action a_{m+j} (corresponding to clause j) marks each group of worlds (which represents a particular assignment to the variables in φ) that ‘satisfies’ clauses 1 to j . (This marking happens by means of an R_c -accessible world where z is true.) Then, in the final updated model, there will only be such a marked group if all clauses, and hence the whole formula, is satisfiable.

For more details, we refer to [36]. \square

PROPOSITION 5. $\{a, e, f, o, p\}$ -DBU is para-NP-hard.

PROOF. To show para-NP-hardness, we specify a polynomial-time reduction R from SAT to DBU, where parameters a, e, f, o and p have constant values. Let φ be a propositional formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. The reduction is based on the same principle as the one used in the proof of Proposition 4. To keep the number of agents constant, we use a different construction to represent the variables in $\text{var}(\varphi)$. We encode the variables by a string of worlds that are connected by alternating relations R_a and R_b .

Furthermore, we keep the size of the formula (and consequently the modal depth of the formula) constant by encoding the satisfiability of the formula with a single proposition. We do this by adding an extra action a_{m+1} . Action a_{m+1} makes sure that each group of worlds that represents a satisfying assignment for the given formula, will have an R_c relation from a world that is R_b -reachable from the designated world to a world where proposition z^* is true.

For more details, we refer to [36]. \square

We consider the following parameterized problem, that we will use in our proof of Proposition 6. This problem is W[1]-complete [18].

$\{k\}$ -MULTICOLORED CLIQUE

Instance: A graph G , and a vertex-coloring $c : V(G) \rightarrow \{1, 2, \dots, k\}$ for G .

Parameter: k .

Question: Does G have a clique of size k including vertices of all k colors? That is, are there $v_1, \dots, v_k \in V(G)$ such that for all $1 \leq i < j \leq k : \{v_i, v_j\} \in E(G)$ and $c(v_i) \neq c(v_j)$?

PROPOSITION 6. $\{a, c, f, o, u\}$ -DBU is W[1]-hard.

PROOF. We specify an fpt-reduction R from $\{k\}$ -MULTICOLORED CLIQUE to $\{a, c, f, o, u\}$ -DBU. Let (G, c) be an instance of $\{k\}$ -MULTICOLORED CLIQUE, where $G = (N, E)$. The general idea behind this reduction is that we use the worlds in the model to list all k -sized subsets of the vertices in the graph with k different colors, where each individual world represents a particular k -subset of vertices in the graph (with k different colors). Then we encode (in the model) the existing edges between these nodes (with particular color endings), and in the final updated model we check whether there is a world corresponding to a k -subset of vertices that is pairwise fully connected with edges. This is only the case when G has a k -clique with k different colors.

For more details, we refer to [36]. \square

PROPOSITION 7. $\{c, o, p, u\}$ -DBU is W[1]-hard.

PROOF. We specify the following fpt-reduction R from $\{k\}$ -WSAT[2CNF] to $\{c, o, p, u\}$ -DBU. We sketch the general idea behind the reduction. Let φ be a propositional formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. Then let φ' be the formula obtained from φ , by replacing each occurrence of x_i with $\neg x_i$. We note that φ is satisfiable by some assignment α that sets k variables to true if and only if φ' is satisfiable by some assignment α' that sets $m - k$ variables to true, i.e., that sets k variables to false. We use the reduction to list all possible assignments to $\text{var}(\varphi') = \text{var}(\varphi)$ that set $m - k$ variables to true. We represent each possible assignment to $\text{var}(\varphi)$ that sets $m - k$ variables to true as a group of worlds, like in the proof of Theorem 1. (In fact, due to the details of the reduction, in the final updated model, there will be several identical groups of worlds for each of these assignments).

For more details, we refer to [36]. \square

PROPOSITION 8. $\{a, f, o, p, u\}$ -DBU is W[1]-hard.

PROOF. We specify the following fpt-reduction R from $\{k\}$ -WSAT[2CNF] to $\{a, f, o, p, u\}$ -DBU. We modify the reduction in the proof of Proposition 7 to keep the values of parameters a and f constant. After these modifications, the value of parameter c will no longer be constant. To keep the number of agents constant, we use the same strategy as in the reduction in the proof of Proposition 5, where variables x_i, \dots, x_m are represented by strings of worlds with alternating relations R_b and R_a . Just like in the proof of Proposition 5, the size of the formula (and consequently the modal depth of the formula) is kept constant by encoding the satisfiability of the formula with a single proposition. Then each group of worlds that represents a satisfying assignment for the given formula, will have an R_c relation from a world

that is R_b -reachable from the designated world to a world where proposition z^* is true.

For more details, we refer to [36]. \square

4.2.3 Tractability Results

Next, we turn to a case that is fixed-parameter tractable.

THEOREM 9. $\{e, u\}$ -DBU is fixed-parameter tractable.

PROOF. We present the following fpt-algorithm that runs in time $e^u \cdot p(|x|)$, for some polynomial p , where e is the maximum number of events in the actions and u is the number of updates, i.e., the number of actions.

As a subroutine, the algorithm checks whether a given basic epistemic formula φ holds in a given epistemic model M , i.e., whether $M \models \varphi$. It is well-known that model checking for basic epistemic logic can be done in time polynomial in the size of M plus the size of φ (see e.g. [7]).

Let $x = (P, \mathcal{A}, i, s_0, a_1, \dots, a_f, \varphi)$ be an instance of DBU. First the algorithm computes the final updated model $s_f = s_0 \otimes a_1 \otimes \dots \otimes a_f$ by sequentially performing the updates. For each i , s_i is defined as $s_{i-1} \otimes a_i$. The size of each s_i is upper bounded by $O(|s_0| \cdot e^u)$, so for each update checking the preconditions can be done in time polynomial in $e^u \cdot |x|$. This means that computing s_f can be done in fpt-time.

Then, the algorithm decides whether φ is true in s_f . This can be done in time polynomial in the size of s_f plus the size of φ . We know that $|s_f| + |\varphi|$ is upper bounded by $O(|s_0| \cdot e^u) + |\varphi|$, thus upper bounded by $e^u \cdot p(|x|)$, for some polynomial p . Therefore, deciding whether φ is true in s_f is fixed-parameter tractable. Hence, the algorithm decides whether $x \in \text{DBU}$ and runs in fpt-time. \square

4.2.4 Overview of the Results

We showed that DBU is PSPACE-complete, we presented several parameterized intractability results (W[1]-hardness and para-NP-hardness) and we presented one fixed-parameter tractable result, namely for $\{e, u\}$ -DBU. In Figure 2, we present a graphical overview of our results and the consequent border between fpt-tractability and fpt-intractability for the problem DBU. We leave $\{a, c, p\}$ -DBU and $\{c, f, p, u\}$ -DBU as open problems for future research.

5. DISCUSSION & CONCLUSIONS

We presented the DYNAMIC BELIEF UPDATE model as a computational-level model of ToM and analyzed its complexity. The aim of our model was to provide a formal approach that can be used to interpret and evaluate the meaning and veridicality of various complexity claims in the cognitive science and philosophy literature concerning ToM. In this way, we hope to contribute to disentangling debates in cognitive science and philosophy regarding the complexity of ToM.

In Section 4.1, we proved that DBU is PSPACE-complete. This means that (without additional constraints), there is no algorithm that computes DBU in a reasonable amount of time. In other words, without restrictions on its input domain, the model is computationally too hard to serve as a plausible explanation for human cognition. This may not be surprising, but it is a first formal proof backing up this claim, whereas so far claims of intractability in the literature remained informal.

Informal claims about what constitutes sources of intractability abound in cognitive science. For instance, it

seems to be folklore that the ‘order’ of ToM reasoning (i.e., that I think that you think that I think ...) is a potential source of intractability. The fact that people have difficulty understanding higher-order theory of mind [19, 28, 31, 43] is not explained by the complexity results for parameter o – the modal depth of the formula that is being considered, in other words, the order parameter. Already for a formula with modal depth one, DBU is NP-hard; so $\{o\}$ -DBU is not fixed-parameter tractable. On the basis of our results we can only conclude that DBU is fixed-parameter tractable for the order parameter in combination with parameters e and u . But since DBU is fp-tractable for the smaller parameter set $\{e, u\}$, this does not indicate that the order parameter is a source of complexity. This does not mean it may not be a source of difficulty for human ToM performance. After all, tractable problems can be too resource-demanding for humans for other reasons than computational complexity (e.g., due to stringent working-memory limitations).

Surprisingly, we only found one (parameterized) tractability result for DBU. We proved that for parameter set $\{e, u\}$ – the maximum number of events in an event model and the number of updates, i.e., the number of event models – DBU is fixed-parameter tractable. Given a certain instance x of DBU, the values of parameters e and u (together with the size of initial state s_0) determine the size of the final updated model (that results from applying the event models to the initial state). Small values of e and u thus make sure that the final updated model does not blow up too much in relation to the size of the initial model. The result that $\{e, u\}$ -DBU is fp-tractable indicates that the size of the final updated model can be a source of intractability (cf. [38, 39]).

The question arises how we can interpret parameters e and u in terms of their cognitive counterparts. To what aspect of ToM do they correspond, and moreover, can we assume that they have small values in (many) real-life situations? If this is indeed the case, then restricting the input domain of the model to those inputs that have sufficiently small values for parameters e and u will render our model tractable, and we can then argue that (at least in terms of its computational complexity) it is a cognitively plausible model.

In his formalizations of the false belief task Bolander [8] indeed used a limited amount of actions with a limited amount of events in each action (he used a maximum of 4). This could, however, be a consequence of the over-simplification (of real-life situations) used in experimental tasks. Whether these parameters in fact have sufficiently small values in real life, is an empirical hypothesis that can (in principle) be tested experimentally. However, it is not straightforward how to interpret these formal aspects of the model in terms of their cognitive counterparts. The associations that the words *event* and *action* trigger with how we often use these words in daily life, might adequately apply to some degree, but could also be misleading. A structural way of interpreting these parameters is called for. We think this is an interesting topic for future research.

Besides the role that our results play in the investigation of (the complexity) of ToM our results are also of interest in and of themselves. The results in Theorems 1 and 2 resolve an open question in the literature about the computational complexity of DEL. Aucher and Schwarzenrüber [3] already showed that the model checking problem for DEL, in general, is PSPACE-complete. However, their proof for

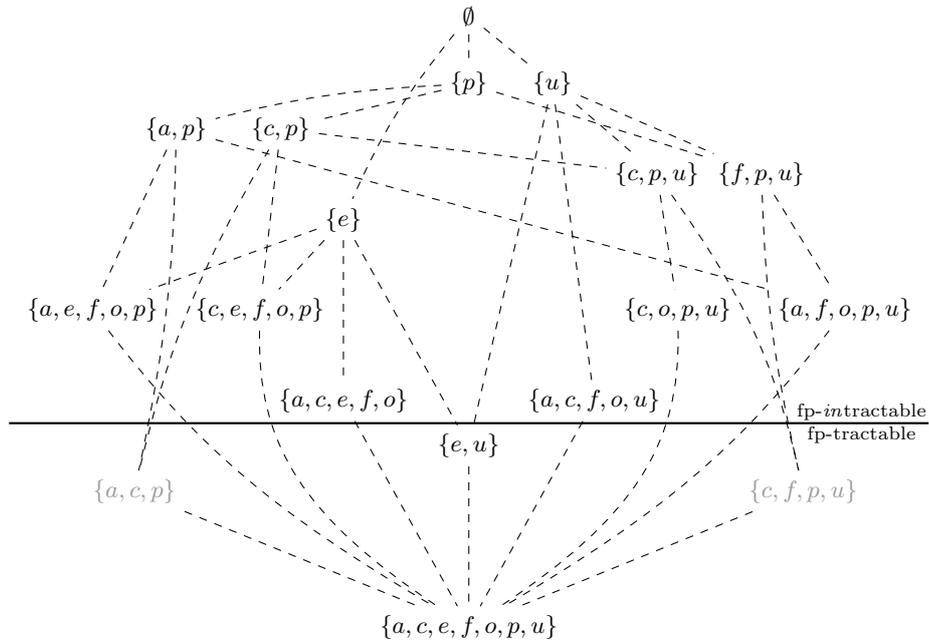


Figure 2: Overview of the parameterized complexity results for the different parameterizations of DBU, and the line between fp-tractability and fp-intractability (under the assumption that the cases for $\{a, c, p\}$ and $\{c, f, p, u\}$ are fp-tractable).

PSPACE-hardness does not work when the input domain is restricted to S5 (or KD45) models and their hardness proof also relies on the use of multi-pointed models (which in their notation is captured by means of a union operator). With our proof of Theorem 1, we show that DEL model checking is PSPACE-hard even when restricted to single-pointed S5 models. Furthermore, the novelty of our approach lies in the fact that we apply parameterized complexity analysis to dynamic epistemic logic, which is still a rather unexplored area.

Acknowledgements

We thank the reviewers for their comments. We thank Thomas Bolander, Nina Gierasimczuk, Ronald de Haan and Martin Holm Jensen and the members of the Computational Cognitive Science group at the Donders Centre for Cognition for discussions and feedback.

6. REFERENCES

- [1] I. Apperly. *Mindreaders: the cognitive basis of “Theory of Mind”*. Psychology Press, 2011.
- [2] G. Aucher. An internal version of epistemic logic. *Studia Logica*, 94(1):1–22, 2010.
- [3] G. Aucher and F. Schwarzentruber. On the complexity of dynamic epistemic logic. In *Proceedings of the Fourteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2013.
- [4] A. Baltag, L. S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 1998.
- [5] S. Baron-Cohen, A. M. Leslie, and U. Frith. Does the autistic child have a “theory of mind”? *Cognition*, 21(1):37–46, 1985.
- [6] J. van Benthem. *Logical Dynamics of Information and Interaction*. Cambridge University Press, Cambridge, 2011.
- [7] P. Blackburn, J. van Benthem, et al. Modal logic: A semantic perspective. *Handbook of modal logic*, 3:1–84, 2006.
- [8] T. Bolander. Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In *Proceedings of European Conference on Social Intelligence (ECSI 2014)*, pages 87–107, 2014.
- [9] T. Bolander and M. B. Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.
- [10] T. Braüner. Hybrid-logical reasoning in false-belief tasks. In B. Schipper, editor, *Proceedings of the Fourteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2013.
- [11] N. Chater, J. B. Tenenbaum, and A. Yuille. Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, 10(7):287–291, 2006.
- [12] C. Cherniak. *Minimal rationality*. MIT Press, 1990.
- [13] S. A. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 151–158. ACM, 1971.
- [14] C. Dégremont, L. Kurzen, and J. Szymanik. Exploring the tractability border in epistemic tasks. *Synthese*, 191(3):371–408, 2014.

- [15] H. van Ditmarsch, W. van der Hoek, and B. P. Kooi. *Dynamic Epistemic Logic*. Springer, 2008.
- [16] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, New York, 1999.
- [17] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- [18] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.
- [19] L. Flobbe, R. Verbrugge, P. Hendriks, and I. Krämer. Children’s application of theory of mind in reasoning and language. *Journal of Logic, Language and Information*, 17(4):417–442, 2008.
- [20] J. Flum and M. Grohe. Describing parameterized complexity classes. *Information and Computation*, 187(2):291–319, 2003.
- [21] J. Flum and M. Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer, Berlin, 2006.
- [22] U. Frith. Mind blindness and the brain in autism. *Neuron*, 32(6):969–979, 2001.
- [23] M. Frixione. Tractable competence. *Minds and Machines*, 11(3):379–397, 2001.
- [24] N. Gierasimczuk and J. Szymanik. A note on a generalization of the muddy children puzzle. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2011.
- [25] G. Gigerenzer. Why heuristics work. *Perspectives on psychological science*, 3(1):20–29, 2008.
- [26] W. F. G. Haselager. *Cognitive Science and Folk Psychology: The Right Frame of Mind*. Sage Publications, 1997.
- [27] A. M. Isaac, J. Szymanik, and R. Verbrugge. Logic and complexity in cognitive science. In A. Baltag and S. Smets, editors, *Johan van Benthem on Logic and Information Dynamics*, volume 5 of *Outstanding Contributions to Logic*, pages 787–824. Springer International Publishing, 2014.
- [28] P. Kinderman, R. Dunbar, and R. P. Bentall. Theory-of-mind deficits and causal attributions. *British Journal of Psychology*, 89(2):191–204, 1998.
- [29] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.
- [30] S. C. Levinson. On the human ‘interaction engine’. In N. J. Enfield and S. C. Levinson, editors, *Roots of human sociality: Culture, cognition and interaction*, pages 39–69. Oxford: Berg, 2006.
- [31] M. Lyons, T. Caldwell, and S. Shultz. Mind-reading and manipulation – is machiavellianism related to theory of mind? *Journal of Evolutionary Psychology*, 8(3):261–274, 2010.
- [32] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: WH Freeman, 1982.
- [33] S. Nichols and S. P. Stich. *Mindreading: An integrated account of pretence, self-awareness, and understanding other minds*. Clarendon Press/Oxford University Press, 2003.
- [34] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, 2006.
- [35] A. Perea. *Epistemic Game Theory: reasoning and choice*. Cambridge University Press, 2012.
- [36] I. van de Pol. How Difficult is it to Think that you Think that I Think that ...? *A DEL-based Computational-level Model of Theory of Mind and its Complexity*. Master’s thesis, University of Amsterdam, the Netherlands, 2015.
- [37] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(04):515–526, 1978.
- [38] I. van Rooij, P. Evans, M. Müller, J. Gedge, and T. Wareham. Identifying sources of intractability in cognitive models: An illustration using analogical structure mapping. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 915–920, 2008.
- [39] I. van Rooij and T. Wareham. Parameterized complexity in cognitive modeling: Foundations, applications and opportunities. *The Computer Journal*, 51(3):385–404, 2008.
- [40] I. van Rooij, C. D. Wright, J. Kwisthout, and T. Wareham. Rational analysis, intractability, and the prospects of ‘as if’-explanations. *Synthese*, pages 1–20, 2014.
- [41] I. van Rooij. The tractable cognition thesis. *Cognitive Science*, 32(6):939–984, 2008.
- [42] I. van Rooij, C. D. Wright, and T. Wareham. Intractability and the use of heuristics in psychological explanations. *Synthese*, 187(2):471–487, 2012.
- [43] J. Stiller and R. I. Dunbar. Perspective-taking and memory capacity predict social network size. *Social Networks*, 29(1):93–104, 2007.
- [44] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the 5th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 1–9. ACM, 1973.
- [45] J. K. Tsotsos. Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13(03):423–445, 1990.
- [46] R. Verbrugge. Logic and social cognition: the facts matter, and so do computational models. *Journal of Philosophical Logic*, 38(6):649–680, 2009.
- [47] H. M. Wellman, D. Cross, and J. Watson. Meta-analysis of theory-of-mind development: The truth about false belief. *Child Development*, 72(3):655–684, 2001.
- [48] H. Wimmer and J. Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. *Cognition*, 13(1):103–128, 1983.
- [49] T. W. Zawidzki. *Mindshaping: A New framework for understanding human social cognition*. MIT Press, 2013.