

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

MSc Thesis (*Afstudeerscriptie*)

written by

Iris van de Pol

(born May 24, 1985 in Eindhoven, the Netherlands)

under the supervision of **Jakub Szymanik (ILLC)** and **Iris van Rooij (RU)**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense: **Members of the Thesis Committee:**

March 9, 2015

Prof. Johan van Benthem

Dr. Iris van Rooij

Dr. Jakub Szymanik

Prof. Rineke Verbrugge

Prof. Ronald de Wolf



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

Theory of Mind (ToM) is an important cognitive capacity, that is by many held to be ubiquitous in social interaction. However, at the same time, ToM seems to involve solving problems that are intractable and thus cannot be performed by humans in a (cognitively) plausible amount of time. Several cognitive scientists and philosophers have made claims about the intractability of ToM, and they argue that their particular theories of social cognition circumvent this problem of intractability. We argue that it is not clear how these claims regarding the intractability of ToM can be interpreted and/or evaluated and that a formal framework is needed to make such claims more precise. In this thesis we propose such a framework by means of a model of ToM that is based on dynamic epistemic logic. We show how the model captures an essential part of ToM and we use it to model several ToM tasks. We analyze the complexity of this model with tools from (parameterized) complexity theory: we prove that the model is PSPACE-complete and fixed-parameter tractable for certain parameters. We discuss the meaning of our results for the understanding of ToM.

Acknowledgements

First of all, I would like to thank Jakub and Iris for their support and guidance through this interesting and challenging journey. In the courses about cognition and complexity that you taught with endless enthusiasm and energy, I happily found an area of study where I could connect several of the many areas that I am interested in, namely philosophy, cognition and computer science. I very much enjoyed our conversations about the plentyful fundamental questions in the interdisciplinary area of logic, cognition and philosophy, and I am grateful for your help with bringing this thesis to a good end. I am looking forward to the continuation of our collaboration the coming years.

I was happy to become part of Iris's Computational Cognitive Science (CCS) group at the Donders Institute for Brain, Cognition and Behavior. I want to thank all of the members of the CCS group for the interesting presentations and discussions at our weekly seminar. It is wonderful to be among people from a variety of (scientific) backgrounds, all interested in computational cognitive modeling and (the philosophy of) cognitive science. Thank you for the intellectually stimulating environment and for the feedback on my work in progress. I would also like to thank my thesis committee members Johan, Rineke and Ronald, for showing their interest in my thesis.

These past two and a half years at the Master of Logic have been full of hard work and long hours, but have also been lots of fun. I want to thank my fellow MoL students for making it such an enjoyable time. Thank you Philip, for bringing me wonderful homemade lunches, and for having the tendency to pull me away from my thesis work and convincing me to join for a match of table tennis. Thank you John, for the many hours, days, and months we spent in the MoL room together, both working on our thesis, and for taking me on exploration tours through the NIKHEF building during those desperately needed breaks. Thank you Mel, Donna, Maartje and Cecilia, for being such patient flatmates and for all the lovely meals without which I might not have survived :-). A golden medal goes to my partner Ronald. Your support means the world to me. You have been the most patient and helpful friend that I could ever imagine.

Last but not least, I want to thank my family for their love and care. Thank you Alma, Jaap, Eva, Carlos and Inés. Whether geographically nearby or distant, you are always close. Special thanks go to my niece and nephews Fenna, Simon, Joris and Milo (who is not with us anymore, but who is in our hearts). Your smiles, play and cries have kept me grounded these past few years. Many thanks go to my parents, Wim and Thera. You always encouraged me to follow

my own interests, and you have always been there when I needed you. You taught me the value of challenging myself and working hard in order to grow and learn, and at the same time you supported me unconditionally and had faith in me, every step along the way.

Contents

Preface	ix
1 Introduction	1
2 Background	3
2.1 Computational-level Theories	3
2.2 The Tractable Cognition Thesis	4
2.3 ToM in Cognitive Science and Philosophy	6
2.4 Intractability Claims	9
2.5 ToM, Logic and Computational Modeling	12
3 Modeling	15
3.1 Preliminaries: Dynamic Epistemic Logic	15
3.1.1 Informal Description of Dynamic Epistemic Logic	15
3.1.2 Formal Description of Dynamic Epistemic Logic	19
3.2 Computational-level Model	22
3.3 Tasks	25
3.3.1 Sally-Anne	25
3.3.2 Chocolate Task	28
3.3.3 Food Truck	30
3.4 Strengths and Weaknesses of the Model	34
4 Complexity Results	37
4.1 Preliminaries: Computational Complexity Theory	37
4.1.1 Classical Complexity Theory	37
4.1.2 Parameterized Complexity Theory	40
4.2 General Complexity Results	43
4.3 Parameterized Complexity Results	55
4.3.1 Intractability Results	56
4.3.2 Tractability Results	70

4.4	Overview of the Complexity Results	71
5	Discussion	73
6	Conclusion	79
	Appendix	91

Preface

This thesis is written with different audiences in mind, and it can be read in several ways. On the one hand, it is aimed at (computational) cognitive scientists that are interested in the complexity of (social) cognition, and in particular in ToM. The model we present in this thesis is based on dynamic epistemic logic (DEL). For those that are unfamiliar with DEL, we include an informal treatment of its main concepts in Section 3.1.1. Those who are mainly interested in the model and the philosophical argument that we present, may choose to skip the details of the complexity-theoretic results in Chapter 4, as these are not essential to understand the argument that we present.

On the other hand, the thesis is also aimed at logicians that are interested in the application of logic in the field of cognitive science, especially those that are interested in (the complexity of) dynamic epistemic logic and cognitive modeling. The reader familiar with DEL might want to skip the informal (and formal description) of DEL in Section 3.1. The reader familiar with (parameterized) complexity theory might want to skip the preliminaries in Section 4.1.

Furthermore, the (parameterized) complexity results of DEL that we present in Chapter 4 are of independent interest. Logicians mainly interested in complexity-theoretic aspects of dynamic epistemic logic can restrict their attention to Chapter 4.

Chapter 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 our life, or to stay one step ahead of our enemies, or just when we lose our friends 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., Frith, 2001, Nichols & Stich, 2003, Premack & Woodruff, 1978, Verbrugge, 2009, Wellman et al., 2001).

We seem to use ToM on a daily basis and many cognitive scientists consider it to be ubiquitous in social interaction (see Apperly, 2011). At the same time, however, it is also widely believed that models of ToM are computationally intractable, i.e., that ToM involves solving problems that humans are not capable of solving with their limited cognitive capacities (see, e.g., Apperly, 2011; Haselager, 1997; Levinson, 2006; Zawidzki, 2013). 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 model 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 computer science there are a variety of tools to make precise claims about the level of complexity of a certain problem.¹ 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.

¹Many of these claims, though, are based on certain widely believed conjectures, such as for instance that $P \neq NP$.

In this thesis we propose a formal framework in which intractability claims regarding models of ToM can be interpreted and evaluated. We use tools from dynamic epistemic logic and (parameterized) complexity theory to provide such a framework. We investigate how the different aspects (or parameters) of our model influence its complexity.

One of these aspects has our particular attention. A defining feature of ToM is the fact that it can vary in order. ‘Dan believes that I will pick him up at eight’, is an example of first-order belief attribution. ‘Trish thinks that Fernando knows that we will throw him a surprise party’ is an instance of second-order belief attribution. In principle this order could increase indefinitely, but in practice most people lose track rather quickly (Kinderman et al., 1998; Lyons et al., 2010; Stiller & Dunbar, 2007). In analyzing the complexity of ToM, we are particularly interested in how this “order parameter” influences the (parameterized) complexity of ToM.

The thesis is structured as follows. In Chapter 2, we begin by introducing a notion of tractability for models of cognition, and we discuss existing research related to ToM in the areas of philosophy, cognitive science, and logic and computational modeling. Then, in Chapter 3, we present a model of ToM based on dynamic epistemic logic. Next, in Chapter 4, we analyze the complexity of this model. Finally, in Chapter 5, we discuss the interpretation of our complexity results and open theoretical questions.

Chapter 2

Background

In this chapter we introduce the notion of computational-level theories and we discuss how one can investigate, by formal means, whether such theories can be computed in a cognitively plausible amount of time. Then, we discuss existing research in the areas of philosophy, cognitive science, and logic and computational modeling, relating to theory of mind.

2.1 Computational-level Theories

There are different levels of explanation that can be employed when studying a cognitive capacity. Marr's (1982) three levels of analysis provide an insightful way to think about this. Marr viewed cognitive systems as information processing systems and he distinguished the following three levels for their analysis: (1) the computational, (2) the algorithmic, and (3) the implementational level. A computational-level analysis is a theory about *what* (information-processing) problem a system is solving. A computational-level theory is specified in terms of a function, i.e., an input-output mapping. An algorithmic-level analysis is a theory of *how* a system computes that function. Such a theory specifies how information is encoded and identifies an algorithm that transforms inputs into the required output. An implementational-level analysis is a theory of how a system physically realizes the algorithm.

The relation between the different levels is both one of dependence and one of independence. A choice at the computational level puts constraints on the collection of algorithmic-level theories that are consistent with it, which are those algorithms that are capable of solving the problem that is proposed at the computational level. In this way the algorithmic level is dependent on the computational level, but at the same time there is also a level of independence or underdeterminism. In principle, there are (infinitely) many algorithms that compute the same function. In the same way, the implementational level is both constrained and underdetermined by the algorithmic level. Only those physical structures that actually implement a given algorithmic-level theory are consistent with it and there are in principle (infinitely) many physical realizations that implement a given

algorithm. This underdetermination of the lower levels is also known as multiple realizability (see Putnam, 1967).

Marr argued that, to offer a full explanation, a theory should cover all three levels. Marr proposed a top-down approach, starting at the computational-level and working down to the physical implementation (Marr, 1982). Because each level has many consistent theories below it, it is very informative to exclude inadequate computational-level theories; this eliminates many inadequate theories at the level below (Blokpoel, 2015).

In this thesis we will focus on the computational level. We will be concerned with capturing, in a qualitative way, what kind of reasoning ToM involves. We will not be concerned with what cognitive mechanisms and implementational structures the mind uses to realize this.

2.2 The Tractable Cognition Thesis

Unfortunately so for cognitive scientists, theories of cognition are highly underdetermined by the available empirical data (cf. Anderson, 1978, 1990). In particular, the information-processing that is described at the computational level happens inside the minds of people and is thus not directly observable. Any tool that can help limit the hypothesis space of computational-level theories is therefore extremely useful and can contribute to faster convergence on better theories. We will explain why computational complexity analysis is such a tool (see also Isaac et al., 2014).

Considerations of computational complexity apply at the computational level, but in fact, its applicability originates from limitations at the physical level. Since humans are finite systems, they have bounded resources for cognitive processing. For instance, the number of cells in our brains, bodies and environment are limited. To illustrate, one may assume a generous upper bound on the number of basic computational steps in the brain of about 10^{12} (neurons) \times 10^3 (connections per neuron) \times 10^3 (firing rate) = 10^{18} steps per second (see, e.g., van Rooij, 2008; Tsotsos, 1990). To sufficiently explain a cognitive capacity, a computational theory of cognition should take into account the processing limitations of humans. If a theory proposes that people can solve a problem that cannot reasonably be assumed to be solvable in a credible amount of time (given the processing limitations of people), then it is not a psychologically plausible explanation.

Computational complexity theory offers a formal framework in which distinctions can be made between the level of complexity of different problems. Several researchers have proposed to use these computational-complexity distinctions to develop a notion of tractable computational-level theories of cognition (cf. Cherniak, 1990; Frixione, 2001; Levesque, 1988; Mostowski & Wojtyniak, 2004; Tsotsos, 1990). They suggested to count those theories as tractable, that correspond to problems that are computable in polynomial time (see also Edmonds (1965)), and they call those theories intractable that correspond with problems that are NP-hard. This view is also known as the *P-Cognition thesis* (van Rooij, 2008). Intuitively, problems that are NP-hard are difficult to solve because the running time of any algorithm that solves it increases exponentially in the

size of the input. This means that for all but trivial cases, solving an NP-hard problem takes an unreasonable amount of time to serve as psychological explanation.

In *The Tractable Cognition Thesis* (2008), van Rooij argues that the *P-Cognition thesis* risks being overly restrictive. Some NP-hard problems actually allow for feasible algorithms when restricted to inputs with certain structural properties – even though in general (i.e., for an unrestricted domain) these algorithms run in super-polynomial time. Because the *P-Cognition thesis* would exclude such theories, it is too constraining and risks the rejection of valid theories. Building on the relatively young theory of parameterized complexity (pioneered by Downey and Fellows, 1999), instead of polynomial-time computability van Rooij proposes fixed-parameter tractability (FPT) as a more appropriate notion of tractability for theories of cognition. A problem is fixed-parameter tractable if it has an algorithm that runs polynomially in the input size and (possibly) non-polynomially only in an additional measure that captures a structural property of the input: the input parameter (see Section 4.1 for a formal definition of fixed-parameter tractability). The *FPT-Cognition thesis* states that computational-level theories of cognition that belong to the hypothesis space of possible theories of cognition are those that are fixed-parameter tractable for one or more input parameters that can be assumed to have small values in real life.

A possible objection to both the *P-Cognition thesis* and the *FPT-Cognition thesis* is that results in computational complexity are built on a certain formalization of computation that might not be applicable to human cognition, namely the Turing machine formalization. (Readers that are unfamiliar with this formalization are referred to the appendix for an informal description and a formal definition of Turing machines.) Turing (1936) proposed his machine model to capture formally what it means for a problem to be computable by an algorithm. Turing claimed that everything that can be calculated by a machine (working on finite data and a finite program of instructions) is Turing machine computable. This is also known as the *Church-Turing thesis*¹ (Copeland, 2008; Kleene, 1967). Most mathematicians and computer scientists accept the Church-Turing thesis and the same seems to hold for many cognitive scientists and psychologists (van Rooij, 2008, but see also Kugel, 1986; Lucas, 1961; Penrose, 1989, 1994).

In computational complexity theory, time is measured in terms of the number of computational steps that are used by a Turing machine (and space in terms of the number of tape cells that the machine uses). Although defined in terms of Turing machines, this measure does not depend on the particular details of the Turing machine formalism, according to the (widely accepted) *Invariance thesis*. The *Invariance thesis* (van Emde Boas, 1990) states that “reasonable machines simulate each other with polynomially bounded overhead in time and constant factor overhead in space”, which means that given two reasonable machine models, the amount of time and space that these

¹Around the same time, Church (1936a) formulated a similar claim based on recursive functions (or lambda-definable functions). All three notions of computability – Turing-computable, recursiveness and lambda-definable – have been proven to be equivalent, i.e., they cover the same collection of functions (Church, 1936b; Kleene, 1936; Turing, 1936).

machines will use to compute the same problem will differ only by a polynomial amount (in terms of the input size). What the *Church-Turing thesis* and the *Invariance thesis* give us (assuming that they are correct), is that computational complexity is not dependent on the underlying machine model. Consequently, also the *P-Cognition thesis* and the *FPT-Cognition thesis* are not dependent on the particular details of the Turing machine, since the measure of complexity abstracts away from machine details.

Like most mathematicians and computer scientists, and many cognitive scientists, we accept both the *Church-Turing thesis* and the *Invariance thesis*, which allows us to abstract away from machine details. In this thesis we will assume that cognitive processing is some form of computation, at least in the broad sense: the transition of a (finite) system from one state into another state. Furthermore, following van Rooij (2008), we will adopt the *FPT-Cognition thesis*, taking fixed-parameter tractability as our notion of tractability for computational-level theories.

Next, we will look more closely at the cognitive capacity ‘theory of mind’ and how it is perceived in cognitive science and philosophy. We do not claim to give a full overview of the many positions; we will merely highlight some of the main practices and debates in experimental psychology and the philosophy of mind.

2.3 ToM in Cognitive Science and Philosophy

In its most general formulation, theory of mind (also called mindreading or folk psychology, or ToM for short) refers to the cognitive capacity to attribute mental states to people and to predict and explain behavior in terms of those mental states, like “*purpose* or *intention*, as well as *knowledge*, *believe*, *thinking*, *doubt*, *guessing*, *pretending*, *liking* and so forth” (Premack & Woodruff, 1978). The recognition of this capacity builds on research in social psychology in the 1950s on how people think about and describe human behavior (Ravenscroft, 2010). In particular, it builds on Fritz Heider’s (1958) important distinction between intentional and unintentional behavior, and his emphasis on the difference that this makes in everyday explanations of behavior. Heider noted that in explanations of others’ behavior, people go far beyond observable data; they make use of causal understanding in terms of mental states such as beliefs, desires and intentions.

Many cognitive scientists consider ToM to be ubiquitous in social interaction (see Apperly, 2011). However, in the past decade there has been an emerging debate in the philosophy of mind about whether ToM is indeed as ubiquitous as often claimed. Many philosophers of the phenomenologist or enactivist type believe that in real life there are only very few cases in which we actually use ToM. Most of the time it might seem that we are engaging in ToM, but really we are using much more basic mechanisms that make use of sociolinguistic narratives and the direct perception of goal-directedness and intentionality (see, e.g., Slors, 2012). In this thesis our main commitment with respect to ToM – consistent with the view by Slors (2012) – is that, at least in some cases, people explain and predict behavior by means of reasoning about mental states, and

that therefore ToM is a cognitive capacity worth investigating (cf. Blokpoel et al., 2012).

In a less recent debate in the philosophy of mind there is the question regarding the realism of mental states and consequently whether mental states can be investigated scientifically. On the one hand, there are *realists* like Jerry Fodor (1987), who claim that the success of everyday explanations of behavior in terms of mental states, also called “folk psychology,” indicates the existence of mental states. On the other hand, there are *eliminativists* like Paul Churchland (1981), who claim that folk psychology is a false theory and that the mental states that it involves are not real. Referring to mental states for psychological explanation is non-scientific and should not be incorporated into scientific theories. Finally, there are the *moderate realists*, or *instrumentalists*, like Daniel Dennett (1987), who agree that common-sense psychology is highly successful, but deny that this implies the independent existence of the mental states involved. Mental state attributions are only true and real in so far as they help us to successfully explain behavior that cannot be explained otherwise (Pitt, 2013). We believe that the fact that we (seem to) use ToM successfully (at least in some cases) is enough justification for scientific investigation, regardless of the (independent) existence of mental states.

From the beginning of ToM research until present day there has been close collaboration between philosophers and psychologists (see Apperly, 2011). The term *theory of mind* was first coined by Premack & Woodruff (1978) in their famous paper *Does the chimpanzee have a theory of mind?* This paper inspired a lively debate in the philosophy of mind (cf. Bennett, 1978; Dennett, 1978; Pylyshyn, 1978), which in turn has led to new paradigms in experimental psychology which investigate perspective change, of which the false-belief task is a notable example (see Wimmer & Perner, 1983).

Two well-known theories in the philosophy of mind that have highly influenced research in experimental psychology are the *Theory theory* and the *Simulation theory*. According to the *Theory theory*, people perform theory of mind (in philosophy referred to as folk psychology) by means of an abstract theory about the relation between mental states and behavior, which is represented in their minds (brains) (Ravenscroft, 2010). According to this view, performing ToM boils down to theoretical reasoning using abstract mental state concepts and principles that describe how they interact.²

According to the *Simulation theory* (see Goldman, 1989; Goldman, 2006; Gordon, 1986), on the other hand, ToM does not entail representing a fully specified theory about the relation between mental states and behavior. ToM does not involve conceptual understanding and reasoning, instead it involves perspective taking by means of simulating the other’s mind with your own: “putting ourselves in the other’s mental shoes” (Slors, 2012). In present day, many cognitive scientists agree that these theories have proven useful to disentangle different aspects that might be involved in ToM. However, they also think that such a sharp distinction between *theory* and *simulation* is not

²See Gopnik & Wellman (1994), Gopnik (1997), and Gopnik, Meltzoff & Kuhl (1999) for examples of the kind of research that this view has inspired in experimental psychology.

productive and they believe that ToM in fact involves a bit of both (cf. Apperly, 2011, Nichols & Stich, 2003).

We believe that both the *Theory theory* and the *Simulation theory* are situated at what Marr (1982) calls the algorithmic level, and that in fact they are equivalent at the computational level. They are not concerned with explaining ToM in terms of *what* the nature of this cognitive capacity is (namely, explaining and predicting behavior in terms of mental states), but in terms of *how* people perform this capacity. They are hypotheses about what cognitive mechanisms enable people to perform ToM. Since we will focus on the computational level, our investigation is independent from assumptions on the cognitive mechanisms that underlie ToM. We are not committed to the *Theory theory* or the *Simulation theory*, nor to any other algorithmic-level theory.

In present-day cognitive science literature on ToM, there is an extensive focus on task-oriented empirical research, particularly on the false-belief task.³ While we underline the importance of experimental research, we think that this fixation on tasks might lead to confusion about the general explanatory goal of cognitive psychology. The overall purpose of psychology is not to understand and explain human performance on tasks. Rather, it is to explain human capacities (Cummins, 2000). Tasks are a tool to tap into cognitive capacities and processes, a tool to test hypotheses and explanations. The risk of focusing mainly on tasks is that they start to lead a life of their own; explaining task performance is then confused with the original target of explaining the capacity that they tap into (in this case ToM).

Another worry concerns the focus on just one particular task (and different flavors of this task), namely the false-belief task. Passing the false-belief task has become a synonym for having ToM, but it is not certain to what extent the false-belief task captures all relevant aspects of ToM. Furthermore, the false-belief task involves much more than just ToM. Aspects such as linguistic performance, dealing with negation, knowledge about the world, executive functioning, and social competence play an important role in the false-belief task, and it is not clear how measurements of the task can single out ToM performance (cf. Apperly, 2011).

Central to cognitive science research on ToM are questions regarding development. Around the age of four children start to pass the benchmark task for ToM performance: the false-belief task (Wellman et al., 2001). However, performance on a non-verbal version of the false-belief task, developed by Onishi & Baillargeon (2005) – that uses the violation-of-expectation-paradigm together with looking-time measures – indicates that infants as young as fifteen months are capable of some form of implicit belief representation. To explain these results, many cognitive scientists adopt two-systems theories (Apperly, 2011) or two-staged theories (Leslie, 2005) of ToM. Although they are very interesting, here we will not be concerned with developmental questions; we will focus

³The false-belief task was first introduced by Wimmer & Perner (1983). In this task children are told a story about Maxi, who is in the kitchen with his mother. They put some chocolate in the fridge, and then Maxi goes outside to play with his friend. While Maxi is away, mother puts the chocolate in a cupboard. Maxi returns, and the child is asked where Maxi thinks the chocolate is. In Section 3.3.1 we will present and formalize a well-known version of this task called the Sally-Anne task.

on what is often called full-blown ToM performance. We believe that explaining the jump in the development of ToM and explaining the possibility of ToM *at all* in the light of intractability claims are two different questions. In this thesis we will focus on the latter question.

Lastly, there is the interesting phenomenon that people seem to have more difficulty with higher-order ToM compared to first-order ToM. Sentences like “I think that you believe in unicorns” are called first-order belief attributions, whereas statements like “I think that you think that I believe in unicorns” are called second-order belief attributions. Second-order belief attribution already seems to be more difficult than first-order belief attribution. For instance, children pass the first-order false-belief task around four years of age (Wellman et al., 2001), while success on second-order versions of this task emerges at about age five or six (Miller, 2009). Both adults and children have been found to make more mistakes on a turn-taking game when it involves second-order reasoning than when it involves first-order reasoning (Flobbe et al., 2008; Hedden & Zhang, 2002). Furthermore, several studies that investigated perspective taking at even higher levels found a prominent drop in performance from the fourth level (Kinderman et al., 1998; Lyons et al., 2010; Stiller & Dunbar, 2007; but see also O’Grady et al., 2015). A commonly held (but debated) view is that higher-order ToM (i.e., beyond first or second level) is cognitively more demanding (see, e.g., Miller, 2009; O’Grady et al., 2015). Therefore, the question arises how the order of ToM contributes to the computational complexity of ToM. This is one of the questions that we investigate in this thesis.

2.4 Intractability Claims

In present-day literature on ToM, intractability is an issue that many researchers are concerned with. Cognitive psychologists and philosophers who try to provide an account of what ToM entails remark that at the sight of it, the way we understand ToM seems to imply that it involves solving an intractable problem (cf. Apperly, 2011; Levinson, 2006; Haselager, 1997; Zawidzki, 2013). Each of these researchers begins by explaining why at first sight ToM seems to be an impossible skill for people to possess. Most of them then continue by building their accounts of ToM in such a way as to circumvent these issues and they claim that their theories are tractable.

We applaud these researchers’ effort to take into account computational complexity constraints in their theorizing about ToM, but we are not sure exactly how to evaluate their claims. This concerns both the seeming intractability of ToM and their solutions for it: the tractability of their own theories. We focus on claims by Ian Apperly (2011). We will argue that without a formal specification it is not clear how to interpret and evaluate these claims.

In *Mindreaders: The Cognitive Basis of “Theory of Mind”* Apperly (2011) tries to solve the seeming intractability of ToM (which he refers to as mindreading – for the sake of convenience, in discussing his argument, we will do so too) by proposing his (two-systems) account of ToM. There are two related issues that Apperly points out as the cause of this intractability. First, he

argues that mindreading entails *abductive inference to the best explanation*. “That’s to say, other beliefs will always be possible, but on the balance of the evidence, we should identify the one that seems most plausible” (Apperly, 2011, p. 118). Furthermore, with Fodor (1983) he argues that “a notorious feature of abductive inferences [is] that there is no way of being certain what information may or may not be relevant” (Apperly, 2011, p. 118,119). Here, he links abductive inference to the *problem of relevance* or *frame problem* (see, e.g., Dennett, 1984; Pylyshyn, 1987; Wheeler, 2008, but see also Rietveld, 2012). He does not distinguish between intractability problems arising from the nature of abductive inference on the one hand and the problem of relevance on the other hand. Apperly argues that they are closely related and that together they are responsible for the seeming intractability of mindreading:

[I]f it is really impossible to limit what information might be relevant for a particular inference or decision, then for anything other than the simplest system there is essentially no limit on the processing that is necessary to search exhaustively through the information that is available. Unlimited search is a computationally intractable problem, with the unpleasant result that reaching a decision or making the inference is impossible. Viewed this way, we should never manage to mindread, not even for the simple case of Sally’s false belief. (Apperly, 2011, p. 119)

We interpret Apperly’s argument as follows. (1) Mindreading requires abductive inference. (2) Abductive inference suffers from the relevance problem. (3) (Because of the relevance problem) abductive inference involves exhaustive search. (4) Problems that involve exhaustive search are computationally intractable (5) Hence, mindreading is intractable.

This argument seems to build on the implicit assumption that the search space for abductive inference is very large (otherwise, exhaustive search would not be such a problem) and that there is no smart algorithm that can find the solution without searching through the entire space. This is indeed what is commonly assumed to be the case for NP-hard problems (Garey & Johnson, 1979), but it is not the case that all problems with large search spaces are dependent on naïve exhaustive search algorithms. Take for instance the problem of finding the shortest path between two nodes in a graph. The search space for this problem (the amount of possible paths between the 2 nodes in the graph) is very large; in the worst case it is $\sum_{i=2}^n (n-i)!$, where n is the number of nodes. Despite its large search space, the shortest path problem can be solved efficiently (in polynomial time) by Dijkstra’s (1959) algorithm. The property of having a large search space is by itself not a necessary cause of intractability. This shows that intuitions about intractability can sometimes be misleading. That is exactly why it is important to specify more precisely what is meant by computational intractability.

We argue that there are two factors in Apperly’s argument that should be distinguished from each other. On the one hand, there is the fact that mindreading involves some form of abductive inference and that this form of reasoning is assumed to be intractable. The well-known problem

of propositional abduction in computer science has indeed been proven to be intractable (Σ_2^P -complete, Eiter & Gottlob, 1995), and also other formalizations of abduction are notorious for their intractability (see Blokpoel et al., 2010). It is not immediately clear, however, whether propositional abduction (or other formalisms of abduction) corresponds to the kind of inference that Apperly is referring to. To evaluate Apperly’s claim about the intractability of abductive inference to the best explanation, it is necessary to specify more formally what this form of reasoning entails.

The second factor is that of the relevance problem. We agree that the relevance problem is a serious issue for the entire field of cognitive science. However, we claim that, instead of contributing to the complexity of a particular theory, the relevance problem arises before specifying a particular theory. For decision (and search) problems – and in a similar way for other computational models of cognition – the input is always assumed to be given. Part of the relevance problem is exactly this assumption of a given input (cf. Blokpoel et al., 2015; Kwisthout, 2012). Even if a theory by itself would be tractable, it is not clear how people can select the right input, and it would therefore be only half of the explanatory story. Although we agree that the relevance problem is a serious challenge for cognitive science, we will not discuss it in further detail in this thesis.

Apperly’s solution to the (seeming) intractability of mindreading lies in his two-systems theory. Apperly (2011, p. 143) argues “that human adults have two kinds of cognitive processes for mindreading – ‘low-level’ processes that are cognitively efficient but inflexible, and ‘high-level’ processes that are highly flexible but cognitively demanding.” Apperly proposes that his two-systems theory explains how mindreading can be tractable:

On my account we should be extremely worried by the potentially intractable computational problems posed by mindreading. Facing up to these problems leads to the expectation that people use two general kinds of cognitive process for high-level and low-level mindreading that make mindreading tractable in quite different ways. (Apperly, 2011, p. 179)

It goes beyond the scope of this thesis to give a thorough explanation and evaluation of the theory that Apperly proposes. What is important to note is that although Apperly’s worries about intractability stem from his computational-level theory of mindreading (namely from the kind of reasoning that it involves), the solution that he proposes works at the algorithmic level. His dual-systems theory, is a theory about *how* people perform this capacity. It is about the cognitive mechanisms that underlie our mindreading capacities. If, however, Apperly is right and the nature of mindreading entails a computationally intractable problem, then this cannot be solved at the algorithmic level. If a problem is intractable (NP-hard) then (under the conjecture that $P \neq NP$) there can be no algorithm that solves it in a reasonable amount of time (polynomial time). At most, an algorithmic-level theory could tractably approximate a computational-level theory – which is often not the case (cf. van Rooij & Wareham, 2012).

2.5 ToM, Logic and Computational Modeling

Here, we will discuss a few approaches to the study of ToM in the area of logic and computational modeling. In the past decade there has been growing interest in the use of logic for formal models of human reasoning and agency; or as van Benthem (2008) phrased it “[m]odern logic is undergoing a cognitive turn” (see also van Benthem et al., 2007; Isaac et al., 2014; van Lambalgen & Coughlan, 2008; Leitgeb, 2008; Verbrugge, 2009).

When using logic as a tool to study human cognition there is an apparent tension between the normative and descriptive perspective on logic. The normative perspective on logic has led to the rejection of logic as a plausible formalism to represent human reasoning, since there are many cases where human judgment does not abide by the rules of (classical) logic. A famous example of this is the Wason selection task (Wason, 1968), where many subjects give answers that are not consistent with the answer prescribed by classical logic. Among psychologists this led to a widespread rejection of logic as a plausible tool to represent human reasoning and cognition.

However, this discrepancy between logic and human inference is not necessarily inherent to logic as a whole, but stems from the normative view on logic, which is not fruitful in the area of cognition. As Michiel van Lambalgen likes to say in his lectures: “there is no such thing as *logic*, there are only *logics*”. The gap between classical logic and human reasoning does not indicate that human reasoning cannot be described by any logic; the challenge lies in choosing the right logic. Stenning & Van Lambalgen (2008) propose that human reasoning is a form of defeasible reasoning, which they model with a non-monotonic logic based on closed-world reasoning. They use this logic to formalize the first-order false-belief task (Wimmer & Perner, 1983) and other reasoning tasks.

Closely related to logic and behavioral economics are approaches based on game theory. Camerer (2010) uses game theory to study the behavior of subjects in a wide range of strategic games. Hedden & Zhang (2002) use a turn-taking game to study first and second-order ToM in adults. They find that subjects perform well on their game when it requires first-order ToM, while they have much more difficulty applying second-order ToM. Flobbe et al. (2008) also found such a difference between first-order and second-order performance on an adaptation of Hedden and Zhang’s (2002) strategic game, both for children and adults (where the adults outperformed the children). Meijering et al. (2012) studied what kind of algorithm people might use when playing a different presentation of this strategic game, which they call the Marble Drop Game. They use eye-tracking to investigate whether people use backward induction or forward reasoning with backtracking. Bergwerff et al. (2014) and Szymanik et al. (2013) study the same question using computational complexity analysis (see also Szymanik, 2013).

There are several approaches to computational modeling of ToM. One of them is the use of ACT-R models (see, e.g., Hiatt & Trafton, 2010; Triona et al., 2002), which is a (computational) cognitive architecture, (mainly) developed by John Anderson (1993), based on his theory of rational analysis (Anderson, 1990). Arslan et al. (2013) modeled a second-order false-belief task with a

hybrid ACT-R model to study developmental transitions between zeroth, first and second-order reasoning, and they used this to make predictions about children’s performance on first and second-order false belief questions. A virtue of their model is that it is not dependent on the details of the false-belief task, but can be used to model a wide range of situations. Furthermore, it can be used to model arbitrary levels of higher-order ToM. Since the model is based on particular assumptions about the cognitive architecture of the mind (brain), it can be seen as (partially) situated at Marr’s algorithmic level. Because we want our complexity analysis to be independent from any particular assumptions on the cognitive architecture of the mind (brain), we aim to formulate our model at the computational-level.

A popular approach in recent research on human behavior and inference is that of probabilistic modeling, particularly approaches involving Bayesian models. Baker et al. (2011) use a Bayesian inverse planning model⁴ to model the attribution of desire and belief on the basis of observed actions. The strength of Bayesian approaches is that they are good at capturing the role of uncertainty. However, the order parameter (of higher-order ToM) has not yet been formalized by Bayesian models and it is not clear how Bayesian models can deal with higher-order ToM without hard-coding a particular limit on the order. In Section 3.3.3, we will formalize the task that Baker et al. (2011) model (the food truck task) with a different set of formal tools.

The approach that we use here is based on dynamic epistemic logic (DEL) (see van Ditmarsch et al., 2008), which we will discuss in more detail in the next section. DEL is a (modal) logic that can be used to model knowledge and belief. Different kinds of modal logic have been used before to model ToM in particular contexts (e.g., Belkaid & Sabouret, 2014; Bolander, 2014; Braüner, 2013; van Ditmarsch & Labuschagne, 2007; Flax, 2006). To analyze the computational complexity of ToM, we propose a computational model that can capture, in a qualitative way, the kind of reasoning that ToM involves (in a wide range of situations). We model ToM at the computational level as an input-output mapping. Therefore the computational complexity of our model will be independent from the particular algorithms that compute it. Our primary interest is the contribution of the order of ToM on the complexity. Since DEL is based on relational structures (Kripke structures), it is well-suited to represent various degrees of belief attribution (up to any order).

⁴See Blokpoel et al. (2010) for a complexity analysis of this model, and see also van Rooij et al. (2011) for an alternative version of the model that uses recipient design.

Chapter 3

Modeling

In this chapter we present our computational-level model of ToM, based on dynamic epistemic logic (DEL). First, we will – both formally and informally – discuss the basic concepts and definitions of DEL. Then, we will present our computational-level model. Finally, we will use this model to capture several ToM-tasks and we will discuss both the strengths and weaknesses of the model.

3.1 Preliminaries: Dynamic Epistemic Logic

The reader that is familiar with the details of DEL may choose to skip Sections 3.1.1 and 3.1.2. The point to take away is that we use the same framework as van Ditmarsch et al. (2008), with two modifications. Following Bolander & Andersen (2011) we allow both single and multi-pointed (rather than just single-pointed) models and we include postconditions (in addition to preconditions) in our event models (which are mappings to propositional literals). The postconditions will allow us to model ontic change, in addition to epistemic change, which we believe is needed for a general applicability of the model. Furthermore the use of multi-pointed models allows us to represent the internal perspective of an observer (cf. Aucher, 2010; Dégremonet et al., 2014; Gierasimczuk & Szymanik, 2011), instead of the omniscient god perspective (or perfect external view).

For the purpose of this thesis we are mainly interested in epistemic models and event models as semantic objects and not so much in the corresponding language.

3.1.1 Informal Description of Dynamic Epistemic Logic

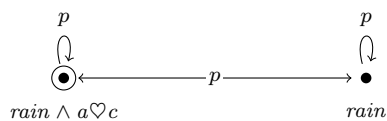
This section is aimed at cognitive scientists who are not familiar with modal logic and dynamic epistemic logic. We will try to explain the main concepts and workings of DEL in an intuitive way. We refer the reader that is familiar with DEL but would like a reminder of the main definitions to Section 3.1.2.

Dynamic epistemic logic is based on a type of relational structures called Kripke frames. A Kripke frame is a collection of possible worlds (points) and an accessibility relation (arrows) between

them. This structure can be used to represent the knowledge and beliefs of one or more agents. To do so, a set of propositions is considered, which are statements about the world that can be true or false. An example of such a proposition is that it is raining in Amsterdam or that Aldo loves Carlos, which (at some point in time) could either be the case or not the case (to keep things simple, we will assume that there is nothing in between true and false).

Let us assume for now that we are some omniscient god and we know which of these propositions are true or false in the actual world (the real world that we live in now). We can represent this knowledge by taking a possible world (which we mark as the actual world) and setting these propositions to true or false accordingly (by defining a valuation function V). Now consider a person, say Philip, who is just an ordinary earthling and does not have perfect knowledge about the actual world. Let us assume that Philip does not know whether Aldo loves Carlos, but he is certain that it is raining in Amsterdam (since he is right in the middle of it, getting soaked). The actual state of affairs (represented in the actual world) is that Aldo indeed loves Carlos (Aldo truthfully told Carlos that this morning) and that it is raining in Amsterdam (Philip is not hallucinating).

In a picture, our example looks as follows. For technical reasons we have a reflexive arrow for each agent in each possible world.¹ The actual world is marked with a circle around it. We use the symbol p for Philip and we label his arrows with p . We use the following labels for the propositions: $rain$ = “it is raining in Amsterdam”, and $a\heartsuit c$ = “Aldo loves Carlos”.



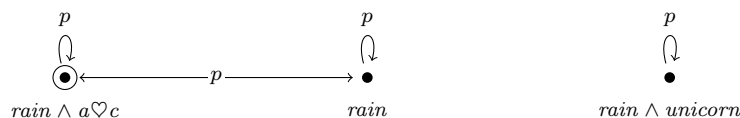
An epistemic model that represents (1) that Philip knows that it is raining, and (2) that Philip does not know whether Aldo loves Carlos.

We represent Philip’s uncertainty about whether Aldo loves Carlos by having another possible world, in which we set the proposition “Aldo loves Carlos” to false and we have a bidirectional arrow between the two worlds. The meaning of an arrow from world 1 to world 2 can be understood as follows: if world 1 would be the actual world, then Philip considers world 2 possible. (Vice versa for an arrow from world 2 to world 1.) Furthermore, we represent the fact that Philip knows that it is raining in Amsterdam by setting “it is raining” to true in both worlds in the model. Philip might not be sure about everything that is going on in the world, but in all different worlds that he considers possible, he knows one thing for sure, namely that it is raining.

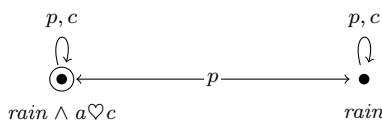
¹The reflexive arrows in epistemic models that represent knowledge (using a specific kind of models, namely S5 models, in which the accessibility relations (which are the arrows between the worlds) have certain properties) come from the (highly) debated assumption (axiom) that knowledge is infallible, i.e., that a model can only express that an agent knows that p , if p is indeed true in the model. There are also other kinds of models – for instance the KD45 models that we will use in our computational-level model – that do not require reflexive arrows for every agent in every world. These kinds of models are often used to model belief. The rationale behind this is that beliefs can be false, whereas knowledge is per definition true. See Section 3.2 for more explanation.

To put it a bit more formally, given a certain world that we have marked as the actual world (w_o), agent a knows that proposition p is true if in all worlds that are accessible (i.e., reachable by an arrow) from the actual world (for agent a), p is true. In the language of epistemic logic, this is expressed with a modal operator, the knowledge operator K . The statement $K_a\varphi$ expresses that agent a knows φ . Similarly, one can express belief with the belief operator B , where $B_a\varphi$ expresses that agent a believes φ . A Kripke frame with possible worlds and (a valuation over) propositions that is used to represent knowledge², is called an epistemic model.

One thing to keep in mind is that these Kripke frames are abstract notions that we use as a tool. The notion of a “possible world” should not be taken literally. The use of Kripke frames to represent knowledge and belief does not commit us to ontological claims about the existence of these worlds (for instance, in some parallel universe). Another thing that can be confusing about the term “possible world” is that we do not necessarily consider all possible worlds to be possible in the everyday sense of the word; not all possible worlds need to be considered as a possible state of affairs. We could for instance add another possible world to our example, one in which the proposition *unicorn* (denoting “unicorns exist”) is true. If Philip, like most people, does not believe that this proposition could ever be true, then he does not consider this “possible world” to be possible. The following picture represents that Philip knows that it is raining in Amsterdam, knows that unicorns do not exist and does not know whether Aldo loves Carlos.



People do not only have knowledge or beliefs about basic propositions but also about other people’s knowledge and beliefs. This can also be represented by epistemic models. In the same model, there can be accessibility relations for different people. Imagine the following situation. Philip has arrived at work where he meets his colleague Carlos (he still knows that it is raining, because he can see it through the window). Carlos knows that Philip does not know whether Aldo loves Carlos (they talked about this yesterday). This situation can be represented by the following epistemic model (in which Carlos also knows that it rains).



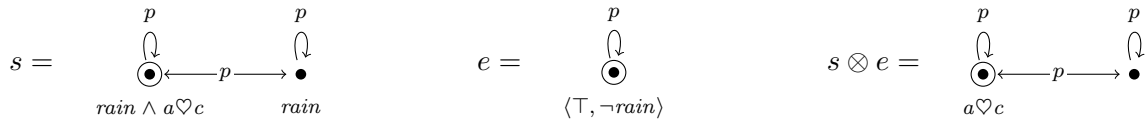
There is no arrow for Carlos between the two possible worlds, since he knows that Aldo loves him and therefore does not consider the world in which Aldo does not love him as a possible

²Kripke frames are also used in various other ways. For instance to represent temporal statements or as mathematical objects in certain mathematical theories.

representation of the world right now. Also, he knows that Philip does not have this knowledge, because in all the worlds that Aldo can reach from the actual world, i.e., all the worlds that he considers possible (which is only the actual world) there is both an arrow for Philip to a world in which $a \heartsuit c$ is true and an arrow to a world in which $a \heartsuit c$ is not true. Remember that an agent a knows φ if in all the worlds that agent a can reach from the actual world, φ is true. That Philip does not know whether Aldo loves Carlos, can be expressed as “Philip does not know that $a \heartsuit c$ is the case and Philip does not know that $a \heartsuit c$ is not the case”, or formally as $\neg K_p a \heartsuit c \wedge \neg K_p \neg a \heartsuit c$. This formula is indeed true in the actual world, which is the only world that Carlos considers possible. Therefore $\neg K_p a \heartsuit c \wedge \neg K_p \neg a \heartsuit c$ is true in all the worlds that are reachable for Carlos from the actual world, and thus the model expresses that Carlos knows that $\neg K_p a \heartsuit c \wedge \neg K_p \neg a \heartsuit c$.

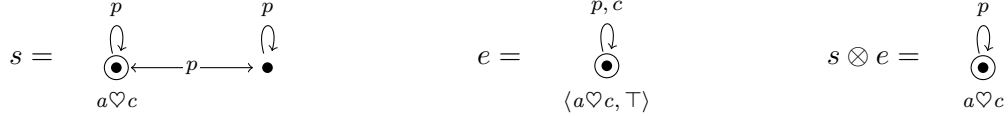
Of course, situations – and also our knowledge and beliefs about them – do not always stay the same. To capture change – either in the world or in epistemic aspects of the situation (the knowledge or beliefs in the minds of people) – we can use event models. An event model is essentially the same kind of model as an epistemic model. An event model can be “applied” to an epistemic model (by means of what is formally called a product update) to yield a representation of the (posterior) situation that results from the occurrence of the event in an initial situation (the epistemic model). Like an epistemic model, an event model consists of worlds (now called events), and an accessibility relation between them. Furthermore, each event in an event model is labeled with preconditions (propositions and possibly epistemic statements) and postconditions (propositions).

Postconditions can change the truth values of the propositions in the possible worlds in the initial situation (the original epistemic model). This is also called ontic change. Intuitively, they represent actual changes in the world. For instance, in the case that it stops raining while Philip is cycling to work. An event model with the postcondition “it is not raining in Amsterdam” will set the proposition “it is raining in Amsterdam” to false in all possible worlds (of the initial epistemic model). Let s be the epistemic model that represents the situation in which it is raining and Philip knows it is raining (and he does not know whether $a \heartsuit c$), and let e be the event model that represent the change of the situation, namely that it stops raining. Then the model that results from applying the event model to the initial epistemic model is denoted by $s \otimes e$. Graphically we can show this as follows. Event models are labeled with a tuple (in this case $\langle \top, \neg rain \rangle$). The first element of the tuple is the precondition and the second element is the postcondition of the event. When an element of the tuple is \top , this simply means that the event has no precondition (or postcondition).



Preconditions define the applicability of an event to a world. Intuitively, preconditions specify

a change in the epistemic situation, by eliminating possible worlds. This is also called epistemic change. Let us go back to the scenario where Philip has arrived at work and meets his colleague Carlos. Now, Carlos tells Philip that he is very happy, because Aldo finally told him that he loves him. Let us assume that both Aldo and Carlos never lie about such things and that Philip knows this. Then, Philip is no longer uncertain about whether Aldo loves Carlos, he now knows that this is indeed the case. We can represent this change in Philip’s knowledge as a result of Aldo’s statement by the following initial model s , event model e , and updated model $s \otimes e$.



In real-life situations, we often do not have such an omniscient take on what is going on (since most of us are not omniscient gods). However, to model certain simple tasks, like the false-belief task, this perfect external modeling perspective is actually quite useful. Since such tasks work under the assumption that all relevant facts are given to the subject, the subject can model the situation as if they are some perfect external spectator. However, many cognitively relevant situations are more complex than the toy examples we find in experimental tasks. To be able to model the beliefs and knowledge in such (uncertain) situations, and to model the internal perspective of agents, in this thesis we allow a *set of possible* worlds in a model to be designated (instead of selecting one world as the actual world), that together constitute the perspective of a particular agent.

3.1.2 Formal Description of Dynamic Epistemic Logic

We introduce some basic definitions from dynamic epistemic logic (DEL), which is an active research field, initiated among others by Baltag et al. (1998), van Benthem (1989), Gerbrandy & Groeneveld (1997); Plaza (1989), but see also van Benthem (2011) for a modern treatment. We base our definitions on the framework provided by van Ditmarsch et al. (2008). Following Bolander & Andersen (2011), we add two modifications. We allow both single and multi-pointed (rather than just single-pointed) models and we include postconditions³ (in addition to preconditions) in our event models (which are mappings to propositional literals).

Dynamic epistemic logic is a particular kind of modal logic, where the modal operators are interpreted in terms of belief or knowledge. Firstly, 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 3.1.1 (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;

³See van Ditmarsch & Kooi (2006) for a different definition of postconditions than the one we use, which is equivalent.

- 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 relations R_a in an epistemic model are accessibility relations, i.e., for worlds $w, v \in W$, wR_av means “in world w , agent a considers world v possible.”

Definition 3.1.2 ((Multi and single-)pointed (perspectival) epistemic model / state). 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) . If W_d is closed under R_a , where $a \in \mathcal{A}$, it is called a perspectival epistemic model for agent a . Given a single-pointed model $(M, \{w\})$, the associated perspectival epistemic model for agent a is $(M, \{v; wR_av\})$. We will also refer to (M, W_d) as a state.

We call a relation R reflexive if for all $w \in W$, Rww ; transitive if for all $w, v, u \in W$, if both Rwv and Rvu then Rwu ; symmetric if for all $w, v \in W$, if Rwv then Rvw ; serial if for all $w \in W$ there exists $v \in W$ such that Rwv ; and Euclidean if for all $w, v, u \in W$, if $(Rwv$ and $Rwu)$ then Rvu . We call a relation KD45 if it is transitive, serial and Euclidean. We call a relation S5 if it is an equivalence relation, i.e., if it is reflexive, transitive and symmetric. If all the relations R_a in a model are KD45 relations, we call the model a KD45 model. Similarly, if all the relations R_a in a model are S5 relations, we call the model an S5 model. In this thesis we focus on KD45 models. However, as we will explain later, all our results also hold for S5 models.

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 beliefs (that) φ ”.

Definition 3.1.3 (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, i.e., truth definitions, for this language are defined as follows.

Definition 3.1.4 (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:

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

When $M, w \models \varphi$, we say that φ is true in w or φ is satisfied in w . We write $M \models \varphi$, when $M, w \models \varphi$ for all $w \in M$. We write $\models \varphi$, when $M, w \models \varphi$ for all epistemic models $M = (W, R, V)$ and all $w \in W$.

Definition 3.1.5 (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 3.1.6 (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 cannot be any formula in \mathcal{L}_B ; they are conjunctions of propositional literals, that is, conjunctions of propositions and their negations (including \top and \perp).

Definition 3.1.7 ((Multi and single-)pointed (perspectival) 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 when E_d is a singleton, and a multi-pointed event model when $|E_d| > 1$. If E_d is closed under Q_a , where $a \in \mathcal{A}$, it is called a perspectival event model for agent a . Given a single-pointed action $(\mathcal{E}, \{e\})$, the associated perspectival event model of agent a is $(\mathcal{E}, \{f; eQ_af\})$. 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 (cf. Baltag et al., 1998).

Definition 3.1.8 (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

- $W' = \{(w, e) \in W \times E; M, w \models \text{pre}(e)\}$;

- $R'_a = \{(w, e), (v, f) \in W' \times W'; wR_av \text{ and } eQ_af\}$;
- $V'((w, e), p) = 1$ iff either $(M, w \models p \text{ and } \text{post}(e) \not\models \neg p)$ 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 3.1.9 (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 $(M, W_d) \otimes a_1 \otimes \dots \otimes a_{k-1}$.*

3.2 Computational-level Model

Next we present our 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. So a necessary part of ToM is the attribution of mental states by an observer to an agent, based on observed actions performed by this agent (in a particular situation). This is what we aim to formalize with our model. There is a wide range of different kinds of mental states such as epistemic, emotional and motivational states. For the purpose of this thesis we focus on a subset of these. In our model we focus on epistemic states, in particular on belief attribution.

The formalism that we will use to build our model is dynamic epistemic logic (DEL). We choose this formalism because it is a well-developed and well-studied framework (see, e.g., van Ditmarsch et al., 2008; van Benthem, 2011) in which belief statements (both for single and multi-agent situations) can be expressed nicely. Furthermore, the relational structures that DEL is based on, are well suited to deal with higher-order belief statements, with no limitation on the level of belief attribution that can be represented. In practice, people cannot deal with arbitrary levels of belief attribution. Many people have difficulty dealing with higher-order ToM (cf. Flobbe et al., 2008; Kinderman et al., 1998; Lyons et al., 2010; Stiller & Dunbar, 2007). However, there is not one specific boundary that limits the level of belief attribution that humans are capable of understanding. Therefore we want our formalism to be able to represent arbitrary levels of belief statements, which DEL can indeed do. Also, our model needs to be able to deal with dynamic situations, in which changes (actions) occur that might influence the beliefs of the agents involved. Whereas (basic) epistemic logic (without event models) can only represent beliefs in static situations⁴, dynamic

⁴Technically, public announcements (see van Ditmarsch et al., 2008) can express dynamics without event models, but one can in fact see public announcements as a particular type of event models.

epistemic logic can deal with a wide range of dynamic situations (by updating epistemic models with event models).

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, so not just actions that change beliefs (epistemic actions), but also actions that change the state of the world. This is why, following Bolander & Andersen (2011), we use postconditions in the product update of DEL (in addition to preconditions) so that we can model also ontic actions in addition to epistemic actions.

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 (2010) – 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 observer 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 real status of certain facts in the world and certain mental states of the agent(s) that they observe. That is why we allow for multi-pointed epistemic models⁵ (in addition to single-pointed models) that can model the uncertainty of an observer, by representing their perspective as a set of worlds. How to represent the internal and/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., Aucher, 2010; Dégrement et al., 2014; Gierasimczuk & Szymanik, 2011). Our proposal is technically similar to Aucher’s definition of internal models, although formulated somewhat differently, and we use it in a different way.

Since we want our model to be cognitively plausible, we do not assume that agents are perfectly knowledgeable. To allow the observers and agents in our representations to have false beliefs about the world, we use KD45 models rather than S5 models. Both KD45 models and S5 models (are based on frames that) satisfy axiom 4 ($B_a\varphi \rightarrow B_aB_a\varphi$); and axioms 5 ($\neg B_a\varphi \rightarrow B_a\neg B_a\varphi$), which specify that agents have positive (4) and negative (5) introspection. In other words, in models that satisfy these axioms, when an agent believes φ , they will also believe that they believe φ ; and when they do not believe φ , they will believe that they do not believe φ . Whether these axioms are cognitively plausible in all situations, can be debated, but at least in some situations these assumptions do not seem problematic.

In the axiom system S5, in addition to the introspection axioms 4 and 5, also axiom T ($B_a\varphi \rightarrow \varphi$) is used, which expresses that belief or knowledge is veridical. This axiom is usually formulated in terms of the knowledge operator K . It then specifies that when a model expresses that an agent

⁵For many multi-pointed models there exists a single-pointed model that is equivalent to it, i.e., that makes the same formulas true, but in general this is not the case. See the appendix for a proof that multi-pointed models are not equivalent to single-pointed models.

knows some statement, this statement must indeed be true in the model. In such a system we cannot express that “Rayan knows that Dewi wants a cookie, while she in fact does not”. Since in real life we often talk about having knowledge without having the means to be completely sure that our ‘knowledge’ is indeed true, axiom T might not be ideal for modeling human agents. Especially when expressed in terms of beliefs, this axiom becomes highly implausible, since it would then specify that all beliefs are veridical, while it is in fact possible to believe something that is false. Therefore, to model beliefs, often the KD45 system is used, where, in addition to axioms 4 and 5, instead of axiom T, axiom D ($\neg B_a \perp$) is used, which expresses that beliefs have to be consistent. In other words, an agent can not believe φ and not φ at the same time. Again, it can be questioned whether the D axiom applies to all situations, but it seems uncontroversial to assume that at least in some cases people have consistent beliefs.

Furthermore, both KD45 and S5 models have the property that all logical tautologies hold in all possible worlds, and therefore every agent knows all logical tautologies. This is also known as the property of logical omniscience. Clearly, not every person (in fact, not even any person) can be assumed to know all logical truths. However, this is not a problem for the purpose of this thesis, since this property does not influence the complexity results that we present. Note that, as we will explain later, our complexity results also do not depend on our choice for KD45 models over S5 models; they hold both for KD45 models and for S5 models (in fact, our complexity results hold even for the unrestricted case, where no additional axioms are used).

Even though KD45 models present an idealized form of belief (with perfect introspection and logical omniscience), we argue that at least to some extent they are cognitively plausible, and that therefore, for the purpose of this thesis, it suffices to focus on KD45 models. The fact that these models are well-studied in the literature, contributes to the relevance of the complexity results that we will present in Chapter 4.

We define the model as follows. For completeness and clarity, we include both an informal and a formal description.

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?

DBU (formal) – DYNAMIC BELIEF UPDATE

Instance: A set of propositions \mathcal{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 used in different ways. First of all, there is the possibility to place the observer either inside or outside the model itself. Depending on the situation to be modeled, the observer can be represented by an accessibility relation in the model (like in our formalization of the food-truck task in Section 3.3.3). One could also choose to leave the observer outside the model, and not represent them⁶ by an accessibility relation in the model. Moreover, one could either use single-pointed epistemic models to represent the perfect external point of view or one could use multi-pointed models to represent an uncertain and/or fallible point of view. The perfect external point of view can be used to model certain (simple) tasks, where all relevant facts are assumed to be given (like in the formalization of the Sally-Anne task⁷ in Section 3.3.1). In other cases, where the observer does not have all relevant information at all stages of a situation, the (multi-pointed) uncertain point of view is more appropriate (like in our formalization of the food-truck task in Section 3.3.3).

For the reader that is familiar with DEL it is worth noting that, when restricted to single-pointed epistemic models and event models without postconditions (i.e., where the postconditions are \top), our definition of DBU is a particular case of the model checking problem of DEL (cf. Aucher & Schwarzentruher, 2013). Therefore, several of our complexity results in Chapter 4 also hold for the model checking problem of DEL.

3.3 Tasks

The model that we presented is highly abstract. In this section we will validate our model by showing that you can naturally use it to model tasks that have been used in psychological experiments. These tasks are considered by psychologists to tap into our capacity of interest: ToM.

3.3.1 Sally-Anne

We present a model of the Sally-Anne task, based on Bolander’s (2014) DEL-based formalization (with some minor adjustments)⁸. The Sally-Anne task (Baron-Cohen et al., 1985; Wimmer & Perner, 1983) is the classic experiment used to test (first-order) understanding of false belief in young children. In this task, children are told or shown a story about Sally and Anne. It goes as

⁶To avoid gender-biases and sexism we choose to use the gender-neutral ‘singular *they*’ instead of the pronouns *he* or *she* in cases where the gender identity is undetermined by the context.

⁷Note that the classically proposed “correct” answer to the Sally-Anne task hinges on the assumption that all relevant facts are given. For instance if Sally would be peaking through a hole in the door when Anne moves the marble or that Sally expects Anne to move the marble to box, because that is what Anne always does, then the answer that Sally thinks that the marble is in the basket would no longer be appropriate.

⁸Later in his paper Bolander actually presents an extended version of his formalization using edge-conditioned event models to capture the relation between seeing and believing. Although we think this is interesting, the formal details that are needed to present this extension take up much space and for the purpose of this thesis the more basic formalization that we present here suffices. Note that the edge-conditioned events are a generalization of the events we use and the hardness results that we present later on also hold when using edge-conditioned events.

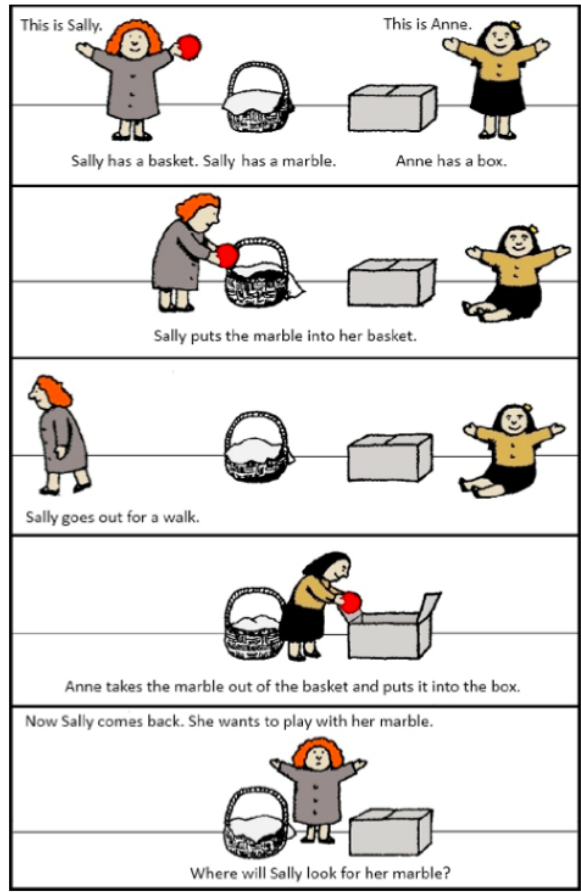


Figure 3.1: An illustration of the Sally-Anne task by Axel Scheffler, from Baron-Cohen et al. (1985), with permission from Elsevier.

follows. (0) There are two children, Sally and Anne, in a room with a box and a basket. Sally has a marble, and (1) she puts it in the basket. (2) Then Sally leaves the room and (3) Anne moves the marble from the basket to the box, after which (4) Sally returns to the room. (See Figure 3.1 for an illustration of the story.)

After being presented the story, children are asked where Sally thinks the marble is. The answer that is considered correct⁹ is that Sally thinks the marble is in the basket (since that is where she left it when she left the room).

The formalization of the story goes as follows. Step (0) is modeled as an epistemic state, and step (1) to (4) as actions. We present the following propositions, epistemic models, and actions. We use agent symbols s and a for Sally and Anne, respectively. We designate the actual world with a circle, and we label event models with with a tuple. The first element of the tuple is the

⁹See footnote 7.

precondition and the second element is the postcondition of the event. When an element of the tuple is \top , this simply means that the event has no precondition (or postcondition).

Propositions

- *basket*: “The marble is in the basket.”
- *box*: “The marble is in the box.”
- *Sally*: “Sally is in the room.”

Initial state and actions

State (0): Sally and Anne are in a room with a box and a basket. Sally has a marble in her hand.

$$s_0 = \begin{array}{c} a, s \\ \downarrow \\ \bullet \\ \text{Sally} \end{array}$$

Action (1): Sally puts the marble into the basket.

$$a_1 = \begin{array}{c} a, s \\ \downarrow \\ \bullet \\ \langle \neg \text{basket}, \text{basket} \rangle \end{array} \quad s_0 \otimes a_1 = \begin{array}{c} a, s \\ \downarrow \\ \bullet \\ \text{Sally, basket} \end{array}$$

Action (2): Sally leaves the room.

$$a_2 = \begin{array}{c} a, s \\ \downarrow \\ \bullet \\ \langle \text{Sally}, \neg \text{Sally} \rangle \end{array} \quad s_0 \otimes a_1 \otimes a_2 = \begin{array}{c} a, s \\ \downarrow \\ \bullet \\ \text{basket} \end{array}$$

Action (3): While Sally is away, Anne puts the marble in the box.

$$a_3 = \begin{array}{ccc} a & & a, s \\ \downarrow & & \downarrow \\ \bullet & \xrightarrow{s} & \bullet \\ \langle \neg \text{Sally} \wedge \text{basket}, & & \langle \neg \text{Sally}, \top \rangle \\ \text{box} \wedge \neg \text{basket} & & \end{array} \quad s_0 \otimes a_1 \otimes a_2 \otimes a_3 = \begin{array}{ccc} a & & a, s \\ \downarrow & & \downarrow \\ \bullet & \xrightarrow{s} & \bullet \\ \text{box} & & \text{basket} \end{array}$$

Action (4): Sally returns to the room.

$$a_4 = \begin{array}{c} a, s \\ \downarrow \\ \bullet \\ \langle \neg \text{Sally}, \text{Sally} \rangle \end{array} \quad s_0 \otimes a_1 \otimes a_2 \otimes a_3 \otimes a_4 = \begin{array}{ccc} a & & a, s \\ \downarrow & & \downarrow \\ \bullet & \xrightarrow{s} & \bullet \\ \text{Sally, box} & & \text{Sally, basket} \end{array}$$

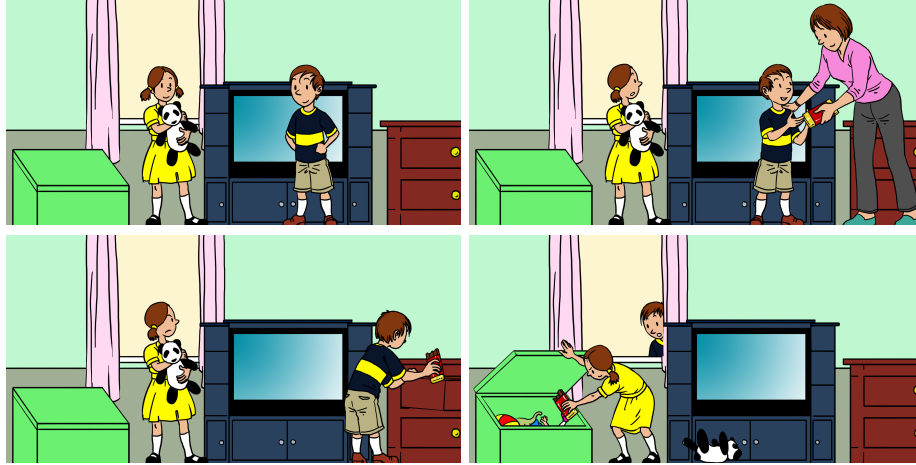


Figure 3.2: An illustration of the chocolate task, by ©Avik Kumar Maitra. Used in a study by Arslan et al. (2015).

In the final model, that results from updating the initial state with the actions, Sally (falsely) believes that the marble is in the basket, where she left it. To put it more formally: $s_o \otimes a_1 \otimes a_2 \otimes a_3 \otimes a_4 \models B_s basket$. The task can now be modeled as the following instance of DBU: $(\{basket, box, Sally\}, \{a, s\}, s_o, a_1, \dots, a_4, B_s basket)$.

3.3.2 Chocolate Task

In a similar fashion, we present a model of a compact version of the second-order chocolate task (Arslan et al., 2015), based on Bolander’s (2014) DEL-based formalization (with some adjustments). In this task, children are told a story about Murat and Ayla. Following Bolander, we model a compact version of the story that goes as follows. (0) Murat and Ayla are in a room. Murat has a chocolate bar and (1) he puts it into a drawer. (2) Then Murat leaves the room and (3) he peaks into the room through a window. (4) Ayla moves the chocolate from the drawer into the toy box. (See Figure 3.2 for an illustration of the story.)

After presenting the story, the experimenter asks the second-order false belief question: “Where does Ayla think that Murat thinks that the chocolate is?” The answer that is considered correct is that Ayla thinks that Murat thinks that the chocolate is in the drawer. (Since Ayla did not see Murat peak through the window, she thinks that Murat expects the chocolate to be where he left it: in the drawer.)

The formalization of the story goes as follows. Step (0) is modeled as an epistemic state, and step (1) to (4) as actions. We present the following propositions, epistemic models, and actions. We use agent symbols m and a for Murat and Ayla, respectively.

Propositions

- *drawer*: “The chocolate bar is in the drawer.”
- *box*: “The chocolate bar is in the toy box.”
- *Murat*: “Murat is in the room.”
- *window*: “Murat is peaking through the window.”

Initial state and actions

State (0): Murat and Ayla are in a room. Murat has a chocolate bar in his hand.

$$s_0 = \begin{array}{c} a, m \\ \downarrow \\ \bullet \\ \text{Murat} \end{array}$$

Action (1): Murat puts the chocolate bar into a drawer.

$$a_1 = \begin{array}{c} a, m \\ \downarrow \\ \bullet \\ \langle \neg \text{drawer}, \text{drawer} \rangle \end{array} \quad s_0 \otimes a_1 = \begin{array}{c} a, m \\ \downarrow \\ \bullet \\ \text{Murat}, \text{drawer} \end{array}$$

Action (2): Murat leaves the room.

$$a_2 = \begin{array}{c} a, s \\ \downarrow \\ \bullet \\ \langle \text{Murat}, \neg \text{Murat} \rangle \end{array} \quad s_0 \otimes a_1 \otimes a_2 = \begin{array}{c} a, m \\ \downarrow \\ \bullet \\ \text{drawer} \end{array}$$

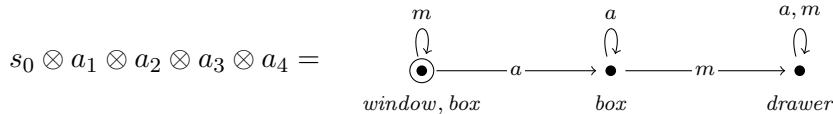
Action (3): Murat peaks into the room through a window.

$$a_3 = \begin{array}{c} m \\ \downarrow \\ \bullet \\ \langle \neg \text{Murat} \wedge \neg \text{window}, \text{window} \rangle \end{array} \xrightarrow{a} \begin{array}{c} a, m \\ \downarrow \\ \bullet \\ \langle \neg \text{Murat} \wedge \neg \text{window}, \top \rangle \end{array}$$

$$s_0 \otimes a_1 \otimes a_2 \otimes a_3 = \begin{array}{c} m \\ \downarrow \\ \bullet \\ \text{drawer} \wedge \text{window} \end{array} \xrightarrow{a} \begin{array}{c} a, m \\ \downarrow \\ \bullet \\ \text{drawer} \end{array}$$

Action (4): Ayla moves the chocolate from the drawer into the toy box.

$$a_4 = \begin{array}{c} m \\ \downarrow \\ \bullet \\ \langle \text{drawer} \wedge \neg \text{Murat} \wedge \text{window}, \\ \neg \text{drawer} \wedge \text{box} \rangle \end{array} \xrightarrow{a} \begin{array}{c} a \\ \downarrow \\ \bullet \\ \langle \text{drawer} \wedge \neg \text{Murat} \wedge \neg \text{window}, \\ \neg \text{drawer} \wedge \text{box} \rangle \end{array} \xrightarrow{m} \begin{array}{c} a, m \\ \downarrow \\ \bullet \\ \langle \text{drawer} \wedge \neg \text{Murat} \wedge \neg \text{window}, \\ \top \rangle \end{array}$$



(We leave out those worlds from the picture that are not accessible from the designated world.)

In the final model, that results from updating the initial state with the actions, Ayla (falsely) believes that Murat believes that the chocolate is in the drawer, where he left it. To put it more formally: $s_o \otimes a_1 \otimes a_2 \otimes a_3 \otimes a_4 \models B_a B_m \text{drawer}$. The task can now be modeled as the following instance of DBU: $(\{\text{drawer}, \text{box}, \text{Murat}, \text{window}\}, \{a, m\}, s_0, a_1, \dots, a_4, B_a B_m \text{drawer})$.

3.3.3 Food Truck

We model the food truck task, which we adapt from Baker et al. (2011). First we describe the situation of the task and then we present our model.

The subject is presented a 2D animation on a computer screen (See Figure 3.3 for an illustration). The subject is told that the animation represents the walking trajectory of a hungry graduate student that left their office and is walking around campus in search of satisfying lunch food. The student knows that there are three different food trucks that regularly offer lunch on campus – Korean (K), Lebanese (L) and Mexican (M). Furthermore, the student knows that there are only two parking spots where food trucks are allowed to park and that consequently, each day there are at most two of those three trucks offering lunch. Since there is no fixed schedule for this, the student is not certain which one they will find.

Figure 3.3 shows a screen shot of one of the scenarios that is presented to the subject. The blue dot represents the graduate student and the yellow squares mark the spots where food trucks are allowed to park. The shaded (gray) region represents the part of the environment that the student cannot see from his current position and the unshaded (white) region represents the students current field of view. The black dots and the lines between them, represent the starting point of the student and their walking trajectory so far.

In the scenario presented in Figure 3.3, the student can initially only see the first parking spot, where truck K is parked. The second parking spot is out of sight. By frame 10, the student has walked past truck K, indicating that they prefer truck M or L (or both) over K. After seeing truck L on parking spot 2 (in frame 10), the student walks back to truck K to have lunch there. Under the assumption that the student’s only goal is to eat at the truck that they prefer most (among the available trucks) and that they act efficiently towards this goal¹⁰, this implies that the student prefers K over L, and moreover, that they prefer M over K.

¹⁰Baker et al. (2011) assume that the graduate student in their task operates under the principle of rational action. As the concept of rationality is rather convoluted, we propose to talk about efficient and (single) goal directed action instead. We think that the main assumptions that are needed to interpret this task in a straightforward manner are

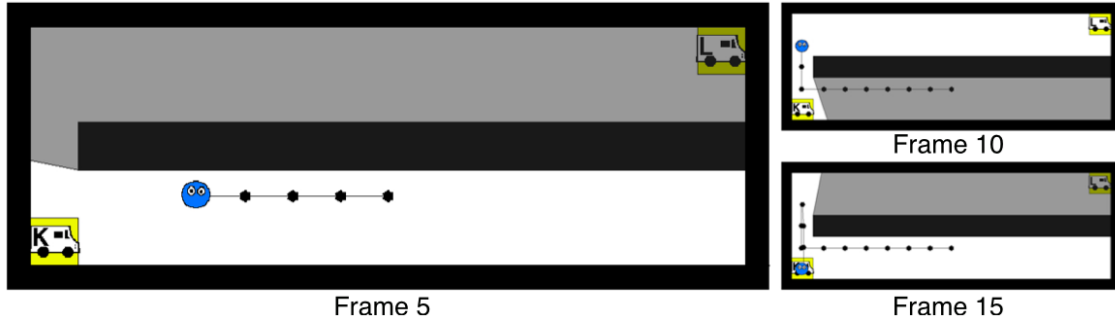


Figure 3.3: Screenshots of an animation in the food truck task, from Baker et al. (2011).

In the original task, subjects are asked (at the end of the animation¹¹) to give a rating of the agent’s desires (for trucks K, L and M, on a scale from 1 to 7) and (retrospectively) of the student’s beliefs about the occupation of the occluded parking spot, before they went on their path (for trucks L, M or no truck). Here, we model the agent’s desires for trucks K, L and M as a preference order. For convenience, we assume that the agent’s preferences are based on a strict order, i.e., for each two trucks he prefers one over the other instead of desiring eating at either one of them equally. Furthermore, we do not model the belief attributions, since they build heavily on the Bayesian assumption that the subject has prior probability distributions on the beliefs of the student, which we do not assume in our framework. Particularly in some of the other scenarios of the task (with a different distribution of the trucks over the parking spots, different environments – allowing for different paths leading to the trucks – and different starting points of the agent) there does not seem to be enough information in the scenario to judge about the belief of the agent (without having prior beliefs). Since there is no additional information given in the task about the previous experiences of the agent (for instance that most of the time they find a particular truck at a particular parking spot), we think that an inference without prior knowledge or assumptions cannot lead to a judgment on the belief of the student about which truck will be at the occluded parking spot.

The experimental data collected by Baker et al. (2011) shows that most subjects indeed inferred that in this scenario, the student prefers M the most, then K and then L. We formalize the reasoning that leads to this answer with the DBU model in the following way. We model the scenario depicted in Figure 3.3. The other scenarios that differ on certain aspect from this one, can be modeled in a similar way.

(1) that getting food is the student’s only goal; for instance that the student will not walk past a certain spot just out of habit, or because they want to say hi to someone there; and (2) that the student will choose the least amount of effort to reach his goal, which is eating at the truck that they prefer most among the available trucks.

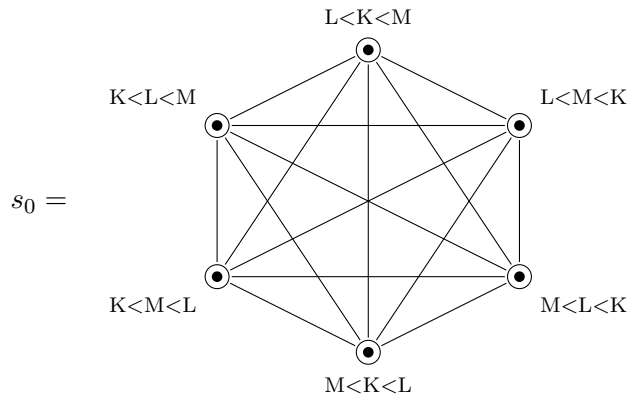
¹¹There were also trials on which subjects were asked to give a rating in the middle of the animation, at the moment that the occluded parking spot became visible for the student. These trials were excluded from the analysis, because subject reports indicated that many were confused by these “Middle” judgment point trials (Baker et al., 2011).

We introduce the following propositions, epistemic models, and actions.

Propositions

- *spot2*: “The agent has walked to a point from where he can see the second parking spot.”
- *eat*: “The agent has eaten.”
- $K < L < M$: “The agent prefers M the most, then L , and then K .”
- $K < M < L$: (similarly)
- $L < K < M$: (similarly)
- $L < M < K$: (similarly)
- $M < L < K$: (similarly)
- $M < K < L$: (similarly)

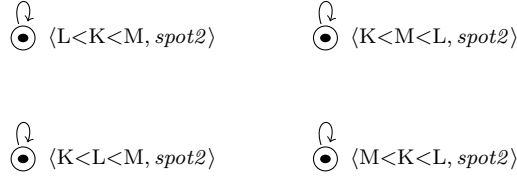
We model the following line of reasoning. Before the agent starts their walk, all options (with regards to his preferences) are still being held possible by the subject. This is represented by the six (connected) possible worlds that are all in the perspectival model for the subject; the subject considers all six possible preference orderings over K , L and M as possible preferences of the student. Notice that in this model it would not work if we were to only select one world, because this would mean that the subject in fact knows the preference of the student. Therefore we use a multi-pointed model, i.e., perspectival model for an agent, which we introduced in Section 3.1.2. We let s be the agent symbol for the subject, but since all the relations in the picture are R_s relations, we omit the label s .



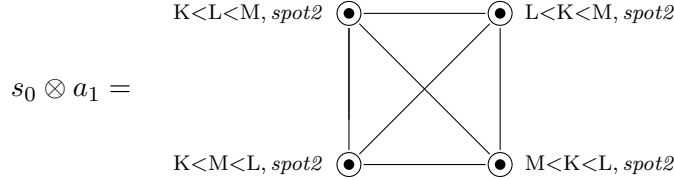
When the student has passed the first parking spot, where K is parked, and moves on to a spot from where he can see the second parking spot, the subject infers that K is not the student’s first choice (otherwise he would have stopped at K immediately to eat, assuming that the student will

choose the least amount of effort that results in eating at the truck that they prefer most among the available trucks.) So the subject does no longer hold preferences $M < L < K$ and $L < M < K$ as possible preferences for the student. This is represented by action 1, that eliminates the worlds where $M < L < K$ and $L < M < K$ are true from the initial model and makes *spot2* true in all worlds.

$a_1 =$ “The student walks to where he can see the second parking spot.”

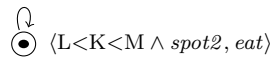


The updated model – after applying action 1 to the initial state – looks as follows.



When the student returns to eat at truck K after having spotted truck L at the second parking spot, the subject infers that the student prefers K over L, and moreover (since going around the corner to where he could see spot 2 indicated that K was not the student’s first choice) that the student prefers M the most, then L and then K. This is represented by action 2, that eliminates all worlds except for the world where $L < K < M$ is true and sets *eat* to true.

$a_2 =$ “The student sees L on spot 2, returns to K and eats.”



The final model (that results from applying action 1 and 2 to the initial situation) is a model with only one possible world, where $L < K < M$, *spot2* and *eat* are true.



The final model represents that the subject believes $L < K < M$ to be the preference of the student, i.e., $s_0 \otimes a_1 \otimes a_2 \models L < K < M$. We can now model the task as the following instance of DBU: $(\{\textit{spot2}, \textit{eat}, K < L < M, \dots, M < K < L\}, \{s\}, s_o, a_1, a_2, L < K < M)$.

3.4 Strengths and Weaknesses of the Model

We introduced the DYNAMIC BELIEF UPDATE model based on dynamic epistemic logic, and we used it to model a first-order false belief task, a second-order false belief task and a task about inferring preferences on the basis of a walking trajectory. Here, we will discuss both the strengths and weaknesses of our model.

The main strength of the model is that it captures an essential part of ToM in a qualitative way. Namely, the attribution of beliefs to an agent on the basis of observed actions performed by that agent in an initial situation (or on the basis of other factors of change). Furthermore, it can be used to model a wide range of situations that include both epistemic (informational) and ontic (factual) change. We showed this for a selection of ToM tasks.

Also, the model allows for different kinds of modeling perspectives, by using either single or multi-pointed (perspectival) models to model respectively a certain or an uncertain point of view. We believe that much can still be gained by creating more flexibility for expressing different (internal and external, certain and uncertain) perspectives in DEL. Especially when dealing with cognitively plausible situations, one needs to be able to describe fallible and uncertain perspectives on a situation, preferably for all agents involved. With the perspectival models that we proposed here, only the uncertain perspective of one of the agents can be modeled. Also, the choice between single and multi-pointed models can only be used to make a crude distinction between the case where there is certainty about what the actual world is and the case where there is no such certainty. However, it cannot be used to distinguish between different levels of belief, i.e., different levels of how certain a person is with respect to certain propositions being true or false. It is quite common in real-life situations to hold such different levels of beliefs. For instance, one can be pretty sure that the supermarket around the corner is open until ten p.m. while being a little less sure that this supermarket sells a particular brand of chocolate. In recent work on the topic of DEL, there have been proposals for how to model degrees of belief (see, e.g., Baltag & Smets, 2006; van Ditmarsch & Labuschagne, 2007). We believe that finding appropriate ways to model uncertain and fallible perspectives and to distinguish between different levels of uncertainty and certainty is an important challenge for DEL. This is especially the case when we want to use DEL to model cognitively plausible agents and situations. Though it is (in principle) not impossible to describe different perspectives in DEL, it is not yet clear what is the best way of dealing with this. These are interesting questions for future research.

The model deals with an idealized notion of belief and knowledge (with perfect introspection and logical omniscience). We formulated our dynamic epistemic language and consequently our DBU model in terms of beliefs, but the model can easily be used to talk about knowledge as well, by replacing the belief operator B with the knowledge operator K . To some extent, the KD45 structures that we used as input for our model can be seen as cognitively plausible, but the notion of belief and knowledge that they can express is idealized. This could perhaps be dealt with by

choosing different (weakened) axioms and corresponding properties of the accessibility relations of the epistemic models. Note that these idealized properties do not influence the complexity results that we present in this thesis.

Moreover, the model can only express epistemic mental states and not the full range of interesting mental states that people can hold. It is not clear how DEL could be used to express this wide range of existing mental states (like emotional and motivational states) in a systematic and flexible way in one model (but see, e.g., Lorini & Schwarzenrüber, 2011). In our formalization of the food truck task, we talked about preferences by representing them with propositions. This can be seen as an ad hoc solution. It would be interesting to see if representing other kinds of mental states can be done in a more systematic, flexible and elegant way.

Finally, we raise the question of whether our model is perhaps overly expressive. We want the model to be expressive enough to model a wide range of situations. At the same time, if the model is too powerful it might not be cognitively plausible, as it can then express much more than humans are capable of. We will prove in Section 4.2 that DBU is PSPACE-complete, which means that without additional restrictions, it takes a (psychologically) unrealistic amount of resources to actually compute it. In Chapter 5 we will propose restriction on the input domain of the model that can render the model tractable, on the basis of our fixed-parameter tractability result in Section 4.

A last strength of the model is that it can deal with higher-order ToM without having to hard-code a particular limit on the level of belief attribution that it can express. Because DEL can deal with arbitrary levels of belief attribution, our model does not have any strict limitation on the maximum level of higher-order ToM. Although there are limits to the level of higher-order theory of mind that people can understand and deal with, there is not one particular limit that bounds this. Therefore, we think it would not be plausible for a model of ToM to hard-code any particular limit on this in the model.

Chapter 4

Complexity Results

In this chapter we analyze the computational complexity of the DYNAMIC BELIEF UPDATE model that we presented in Section 3.2. As we explained earlier, our aim is to investigate informal claims about the complexity of ToM by formal means. We are particularly interested in how the order parameter (the level of belief attribution) influences the complexity of the model. We also consider several other parameters of the model, as they might play a crucial role. For our analysis, we use tools from both classical complexity theory and parameterized complexity theory.

4.1 Preliminaries: Computational Complexity Theory

Here, we introduce the basic concepts of computational complexity theory, which we use in the proofs in Sections 4.2 and 4.3. The reader that is familiar with the details of classical complexity theory may choose to skip Section 4.1.1, and the reader that is familiar with the details of parameterized complexity theory may choose to skip Section 4.1.2.

4.1.1 Classical Complexity Theory

We introduce some basic concepts of classical complexity theory. For a more detailed treatment we refer to textbooks on the topic (e.g., Arora & Barak, 2009).

In complexity theory, computational problems are often studied in the form of decision problems. Decision problems represent yes-no questions that are asked about a given input.

Definition 4.1.1 (Decision problem). *Let Σ be a finite alphabet. A decision problem L (over Σ) is a subset of Σ^* , where Σ^* is the set of all strings over the alphabet Σ , i.e., $\Sigma^* = \bigcup\{\Sigma^m; m \in \mathbb{N}\}$. We call $x \in \Sigma^*$ a yes-instance of L if and only if $x \in L$.*

With the size $|x|$ of an instance $x \in \Sigma^*$ we denote the size of the string x , i.e., the number of symbol occurrences in x . We represent a decision problem L in the following form. To simplify notation, we do not mention the underlying alphabet explicitly.

PROBLEMNAME (L)

Instance: $x \in \Sigma^*$

Question: Is $x \in L$?

Another important notion in complexity theory is the concept of asymptotic running time, which measures how fast the running time of an algorithm increases with the input size. In order to capture this formally, the following notion of Big-Oh is used.

Definition 4.1.2 (Big-Oh). *Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be a function. Then,*

$$O(f) = \{g : \mathbb{N} \rightarrow \mathbb{N}; \exists c, n_o \in \mathbb{N} \forall n > n_o : g(n) \leq c \cdot f(n)\}.$$

As is common, we will write $f(n)$ is $O(g(n))$ instead of $f(n) \in O(g(n))$.

The concept of efficiently solvable problems is captured by the complexity class P, which is defined as follows.

Definition 4.1.3 (The class P). *Let Σ be a finite alphabet.*

- 1. An algorithm A with input $x \in \Sigma^*$ runs in polynomial time if there exists a polynomial p such that for all $x \in \Sigma^*$, the running time of A on x is at most $p(|x|)$. We call algorithms with this property polynomial-time algorithms.*
- 2. P denotes the class of all decision problems that can be decided by a polynomial-time algorithm.*

In order to give evidence that certain problems are intractable, i.e., are not in P, complexity theory offers a theoretical tool that is based on the following complexity class, NP.

Definition 4.1.4 (The class NP). *Let Σ be a finite alphabet and $L \subseteq \Sigma^*$ a decision problem. Then L is in complexity class NP if there exists a polynomial p and a polynomial-time computable function $f : \Sigma^* \rightarrow \mathbb{N}$ (called the verifier for L) such that for all $x \in \Sigma^*$,*

$$x \in L \iff \exists u \in \Sigma^{p(|x|)} \text{ s.t. } f(x, u) = 1.$$

If $x \in L$ and $u \in \Sigma^{p(|x|)}$ satisfy $f(x, u) = 1$, then we call u a certificate for x .

Another crucial part of this intractability tool is the notion of polynomial-time reductions.

Definition 4.1.5 (Polynomial-time reduction). *Let $L \subseteq \Sigma^*$ and $L' \subseteq (\Sigma')^*$ be two decision problems. A polynomial-time reduction from L to L' is a mapping $R : \Sigma^* \rightarrow (\Sigma')^*$ from instances of L to instances of L' such that for all $x \in \Sigma^*$:*

- 1. $x' = R(x)$ is a yes-instance of L' if and only if x is a yes-instance of L , and*

2. R is computable in polynomial time.

We can now describe the final notions that we need for the theoretical tool to show intractability: the notions of hardness and completeness for a certain complexity class.

Definition 4.1.6 (Completeness and Hardness). *Let L be a decision problem and K a complexity class. Then,*

1. L is K -hard if each problem L' in K is polynomial-time reducible to L , and
2. L is K -complete if L is K -hard and in K .

It follows from the definitions that $P \subseteq NP$. It is widely believed that $P \neq NP$ (see, e.g., Fortnow, 2009, Gasarch, 2012). This conjecture implies that NP-hard problems are not polynomial-time decidable. Therefore, showing that a problem is NP-hard gives evidence that this problem is intractable.

The following two complexity classes can be used in a similar way to show intractability: a problem that is hard for any of these classes is not polynomial-time solvable, unless $P = NP$.

Definition 4.1.7 (The class co-NP). *Let Σ be a finite alphabet and $L \subseteq \Sigma^*$ a decision problem. Then L is in complexity class co-NP if there exists a polynomial p and a polynomial-time computable function $f : \Sigma^* \rightarrow \mathbb{N}$ (called the verifier for L) such that for all $x \in \Sigma^*$,*

$$x \in L \iff \forall u \in \Sigma^{p(|x|)} \text{ it holds that } f(x, u) = 1.$$

If $x \in L$ and $u \in \Sigma^{p(|x|)}$ satisfy $f(x, u) = 1$, then we call u a certificate for x .

Definition 4.1.8 (The class PSPACE). *Let $s : \mathbb{N} \rightarrow \mathbb{N}$ be a function. We say that a Turing machine \mathbb{M} runs in space s if for every $x \in \Sigma^*$ every run of \mathbb{M} with input x only consists of configurations of size at most $s(|x|)$ (see the appendix for a definition of Turing machines). The class PSPACE consists of all problems Q for which there exists a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ and a Turing machine \mathbb{M} such that (1) \mathbb{M} runs in space p and (2) \mathbb{M} decides Q .*

Intuitively, the class PSPACE consists of all problems that can be solved by an algorithm that uses a polynomial amount of memory.

As mentioned before, the class P is a subset of NP . Similarly, P is also a subset of co-NP. Whether P is a strict subset of co-NP (or NP) is not known, but it is widely believed that this is the case. It is also widely believed that NP and co-NP do not coincide, but this is also not known. Finally, all classes P , NP and co-NP are contained in the class PSPACE. Again, whether these inclusions are strict is not known. For an overview of these complexity classes, see Figure 4.1.

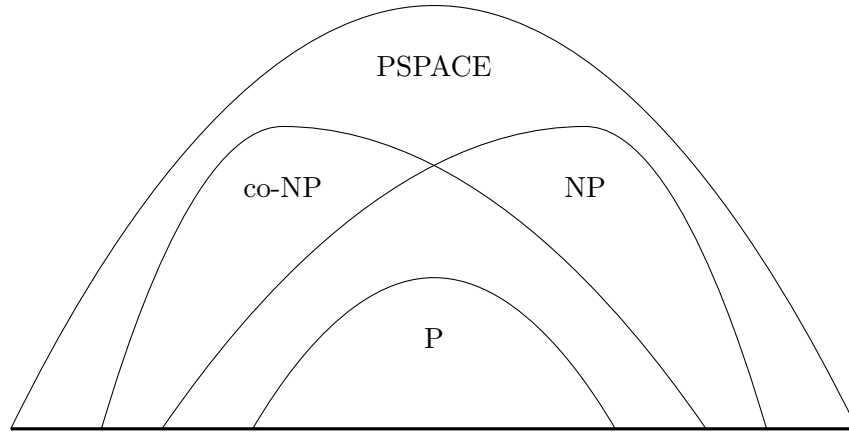


Figure 4.1: Overview of the complexity classes P, NP, co-NP, and PSPACE.

4.1.2 Parameterized Complexity Theory

Next, we introduce some basic concepts of parameterized complexity theory. For a more detailed introduction we refer to textbooks on the topic (Downey & Fellows, 1999, 2013; Flum & Grohe, 2006; Niedermeier, 2006). Readers with a background in cognitive science (rather than in computer science) may find treatments by van Rooij (2003), van Rooij (2008), and Wareham (1998) particularly illustrative.

Definition 4.1.9 (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.*

Often, the assumption is made that the parameter value k is computable from the main part x of the input in polynomial time (see, e.g., Flum & Grohe, 2006). All the parameters that we consider in this thesis satisfy this assumption.

We represent a parameterized problem L in the following form. Again, to simplify notation, we do not mention the underlying alphabet explicitly.

$\{k\}$ -PROBLEMNAME (L)
Instance: $(x, k) \in \Sigma^* \times \mathbb{N}$
Parameter: k
Question: Is $x \in L$?

Even though, technically speaking, the parameter in a parameterized problem is a single value, for the sake of convenience, we will often use several parameters. For instance, in a parameterized problem $\{k_1, \dots, k_n\}$ -PROBLEMNAME, we will say that each of the values k_i is a parameter, while strictly speaking the single parameter value of the problem is $k_1 + \dots + k_n$.

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 4.1.10 (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 4.1.11 (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 Boolean formula φ , and an integer k .

Parameter: k .

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

Definition 4.1.12 (The class W[1]). *The class W[1] consists of all parameterized problems that can be fpt-reduced to the problem $\{k\}$ -WSAT[2CNF].*

The concepts of hardness and completeness (for a parameterized complexity class) are now defined similarly to the corresponding notions in classical complexity.

Definition 4.1.13 (Hardness and completeness). *Let L be a parameterized problem and K a parameterized complexity class. Then,*

1. L is K -hard if each problem L' in K is fpt-reducible to L , and
2. L is K -complete if L is K -hard and in K .

It is widely believed that $\text{FPT} \neq \text{W}[1]$ (see Downey & Fellows, 2013). This conjecture implies that $\text{W}[1]$ -hard problems are not fixed-parameter tractable. Therefore, showing that a problem is $\text{W}[1]$ -hard gives evidence that this problem is not fixed-parameter tractable.

Another parameterized intractability class, that can be used in a similar way, is the class para-NP .

Definition 4.1.14 (The class para-NP). *The class para-NP consists of all parameterized problems that can be solved by a nondeterministic algorithm that runs in fpt-time. Intuitively, a nondeterministic algorithm is an algorithm that can make guesses during the computation. The algorithm solves the problem if there is at least one sequence of guesses that leads the algorithm to accept. The algorithm runs in fpt-time if for all possible sequences of guesses the algorithm terminates in fpt-time. For more details about nondeterministic algorithms, we refer to textbooks on complexity theory (e.g., Arora & Barak 2009).*

$\text{W}[1]$ is a subset of para-NP . This implies that para-NP -hard problems are not fixed-parameter tractable, unless $\text{W}[1] = \text{FPT}$. In fact, the conjecture $\text{P} \neq \text{NP}$ already implies that para-NP -hard problems are not fixed-parameter tractable (cf. Flum & Grohe, 2006, Theorem 2.14).

The parameterized complexity class para-PSPACE can be used similarly for the same purpose of giving evidence against fixed-parameter tractability.

Definition 4.1.15 (The class para-PSPACE). *The class para-PSPACE consists of all problems Q for which there exists a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$, a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ and a Turing machine \mathbb{M} such that for any input (x, k) (1) \mathbb{M} runs in space $f(k) \cdot p(|x|)$ and (2) \mathbb{M} decides Q (see the appendix for a definition of Turing machines).*

For a more formal (and slightly differently formulated) definition of para-PSPACE , we refer to the work of Flum & Grohe (2003).

The following inclusions hold for the parameterized complexity classes discussed above: $\text{FPT} \subseteq \text{W}[1] \subseteq \text{para-NP} \subseteq \text{para-PSPACE}$. These inclusions are believed to be strict, but this is not known. An interesting difference between $\text{W}[1]$ and para-NP is that problems in $\text{W}[1]$ can be solved in time $O(n^{f(k)})$ – where n is the size of the input, k is the parameter value, and f is some computable function – whereas this is not possible for problems that are para-NP -hard, unless

$P = NP$ (Flum & Grohe, 2006). In other words, problems in $W[1]$ can be solved in polynomial time for a fixed parameter value k , where the order of the polynomial depends on k . For an overview of these parameterized complexity classes, see Figure 4.2. The reason why classical complexity classes (such as P and NP) are not depicted in this overview is that one cannot directly compare classical complexity classes and parameterized complexity classes, in terms of inclusion. This is because classical complexity classes are subsets of Σ^* , whereas parameterized complexity classes are subsets of $\Sigma^* \times \mathbb{N}$.

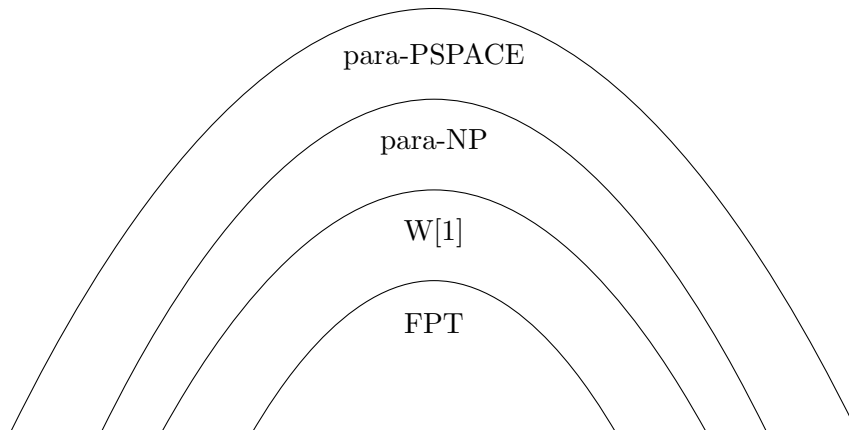


Figure 4.2: Overview of the parameterized complexity classes FPT , $W[1]$, $para-NP$, and $para-PSPACE$.

4.2 General Complexity Results

In this section we will show that our model is $PSPACE$ -complete. We will first provide two different proofs of NP -hardness. Giving these proofs might seem unnecessary at first sight, since $PSPACE$ -hardness implies NP -hardness. However, we will use these proofs in the next section to establish some parameterized complexity results. Moreover, the proofs allow us to introduce the principles that we will use in the (more involved) $PSPACE$ -hardness proof.

We note that the results in this section (and in Section 4.3) hold for epistemic models with arbitrary relations. In fact, they hold even when restricted to $KD45$ models or to $S5$ models. Because every $S5$ model is also a $KD45$ model, we know that showing hardness for any problem restricted to $S5$ models also implies hardness for the problem restricted to $KD45$ models. Namely, if you could solve the problem efficiently for $KD45$ models, you would also be able to solve it efficiently for $S5$ models. Therefore, since our hardness proofs only use $S5$ models, they also hold for $KD45$ models. Similarly, if you show membership to a particular complexity class for any problem restricted to $KD45$ models, then you immediately show membership to this class for the restriction to $S5$ models. Therefore, since our proof of $PSPACE$ -membership in the proof of Theorem 5 holds

for KD45 models, we also showed PSPACE-membership for S5 models.

We consider the following well-known problems, SAT and UNSAT, that we will use in the reductions in this section. The problem SAT is NP-complete, and the problem UNSAT is co-NP-complete (cf. Cook, 1971; Levin, 1973). Moreover, hardness for SAT holds even when restricted to Boolean formulas that are in 3CNF (that is, formulas that consist of conjunctions of disjunctions of three literals; these disjunctions are called clauses).

SAT

Instance: A Boolean formula φ .

Question: Is φ satisfiable? I.e., does there exist an assignment $V : \text{var}(\varphi) \rightarrow \{0, 1\}$, that satisfies φ ?

UNSAT

Instance: A Boolean formula φ .

Question: Is φ unsatisfiable? I.e., is it the case that for all assignments $V : \text{var}(\varphi) \rightarrow \{0, 1\}$, φ is not satisfied?

Proposition 1. DYNAMIC BELIEF UPDATE (DBU) is NP-hard.

Proof. To show NP-hardness, we specify a polynomial-time reduction R from SAT to DBU. Let φ be a Boolean 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 idea behind this reduction is that we use the worlds in the model that results from updating the initial state with the actions – which are specified 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. Action a_1 to a_m are used to enumerate 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 (which we will use to derive para-NP-hardness results in the next section), we sequentially check the truth of each clause c_i and encode whether the clauses are true with an additional variable: variable x_{m+1} . This is done by actions a_{m+1} to a_{m+l} . In the final updated model, variable x_{m+1} will only be true in a world, if it makes clauses c_1 to c_l true, i.e., if it makes formula φ true.

Next, we specify the reduction. We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_{m+l}, \hat{B}_a x_{m+1})$, where:

- $P = \{x_1, \dots, x_m, x_{m+1}\}$, where x_{m+1} is a proposition that does not occur in φ ;
- $\mathcal{A} = \{a\}$;

- $s_0 = ((W, V, R), W_d) = \begin{array}{c} \curvearrowright \\ \bullet \\ w_1 \end{array}$
- $a_1 = ((E, Q, pre, post), e_1) = \begin{array}{ccc} \curvearrowright & & \curvearrowright \\ \bullet & \text{---} a \text{---} & \bullet \\ e_1 : \langle \top, \top \rangle & & e_2 : \langle \top, x_1 \rangle \end{array}$
- \vdots
- $a_m = \begin{array}{ccc} \curvearrowright & & \curvearrowright \\ \bullet & \text{---} a \text{---} & \bullet \\ e_1 : \langle \top, \top \rangle & & e_2 : \langle \top, x_m \rangle \end{array}$
- $a_{m+1} = \begin{array}{ccc} \curvearrowright & & \curvearrowright \\ \bullet & \text{---} a \text{---} & \bullet \\ e_1 : \langle \top, \top \rangle & & e_2 : \langle c_1, x_{m+1} \rangle \end{array}$
- $a_{m+2} = \begin{array}{ccc} \curvearrowright & & \curvearrowright \\ \bullet & \text{---} a \text{---} & \bullet \\ e_1 : \langle \neg x_{m+1}, \top \rangle & & e_2 : \langle c_2 \wedge x_{m+1}, x_{m+1} \rangle \end{array}$
- \vdots
- $a_{m+l} = \begin{array}{ccc} \curvearrowright & & \curvearrowright \\ \bullet & \text{---} a \text{---} & \bullet \\ e_1 : \langle \neg x_{m+1}, \top \rangle & & e_2 : \langle c_l \wedge x_{m+1}, x_{m+1} \rangle \end{array}$

We show that $\varphi \in \text{SAT}$ if and only if $R(\varphi) \in \text{DBU}$.

First we note that the intermediate state $s_{f'} = s_o \otimes a_1 \otimes \cdots \otimes a_m$ consists of $2^{|P|}$ worlds that are all connected. Furthermore, each possible assignment $\alpha : \text{var}(\varphi) \rightarrow \{0, 1\}$ corresponds with the valuation over P in exactly one of these worlds, and vice versa each of these worlds corresponds with such an assignment α . Actions a_{m+1}, \dots, a_{m+l} make sure that also in the final state $s_f = s_{f'} \otimes a_{m+1} \otimes \cdots \otimes a_{m+l}$ there is at least one world (possibly more than one) for each possible assignment $\alpha : \text{var}(\varphi) \rightarrow \{0, 1\}$, such that the valuation over P in that world corresponds with α . Also, all worlds in s_f are connected. Furthermore, for $m+1 \leq i \leq m+l$, a_i sets x_{m+1} to true in a world w if and only if the assignment corresponding to the valuation over P in w satisfies clauses c_1 to c_i and it eliminates worlds where x_{m+1} is true but whose corresponding assignment does not satisfy clause c_i . Therefore, in each world in s_f , x_{m+1} is true if and only if the valuation over P in that world is a satisfying assignment for $c_1 \wedge \cdots \wedge c_l = \varphi$. Hence, $\varphi \in \text{SAT}$ if and only if $R(\varphi) \in \text{DBU}$.

Since this reduction runs in polynomial time, we can conclude that DBU is NP-hard. □

Corollary 2. *DBU is co-NP-hard.*

Proof. We can modify the reduction in the proof of Proposition 1 into a polynomial-time reduction from UNSAT to DBU. We replace the formula $\hat{B}_a x_{m+1}$ with $\neg \hat{B}_a x_{m+1}$. Then $\varphi \in \text{UNSAT}$ if and only if $R(\varphi) \in \text{DBU}$. \square

Corollary 3. *DBU is not NP-complete, unless $\text{co-NP} = \text{NP}$.*

Proof. Assume that DBU is NP-complete, i.e., DBU is in NP. Then there is a co-NP-hard problem in NP and hence $\text{co-NP} \subseteq \text{NP}$. We show that also $\text{NP} \subseteq \text{co-NP}$. Let $L \in \text{NP}$ and let \bar{L} be its complement. Then $\bar{L} \in \text{co-NP}$. Since $\text{co-NP} \subseteq \text{NP}$, we know that $\bar{L} \in \text{NP}$ and thus $L \in \text{co-NP}$. \square

In Section 4.2 we will use the proof of Proposition 1 to derive a para-NP-hardness result (Corollary 7). Below, we present an alternative proof for Proposition 1, which we will use to derive a different para-NP-hardness result (Corollary 11). The reduction in this alternative proof of Proposition 1 also serves as an introduction to the reduction in the proof of Theorem 4, which is based on the same principle.

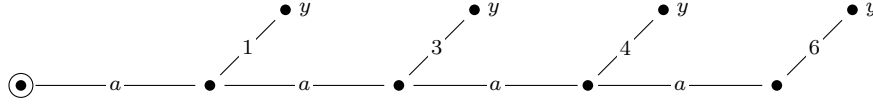
Alternative proof of Proposition 1. To show NP-hardness, we specify a polynomial-time reduction R from SAT to DBU.

First, we sketch the general idea behind the reduction. Let φ be a Boolean formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. Similarly to the first proof of Proposition 1, we use the reduction to list all possible assignments to $\text{var}(\varphi)$. The difference with the reduction in the proof of Proposition 1 is that in that reduction we used individual worlds to represent particular truth assignments, while here we will use groups of worlds (which are R_a -equivalence classes) to represent truth assignments. In the proof of Proposition 1 we marked the true variables (under a particular assignment) by setting their corresponding proposition to true or false in one particular world. In this alternative proof, we will use a group of worlds for this, where each world in the group represents a true variable (under a particular assignment).

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 φ will be represented by such a group of worlds. 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). Since we want to keep the number of propositions constant in this reduction (which we will use to derive para-NP-hardness results in the next section), we cannot use variables x_1, \dots, x_m . Instead, we use different agents: agent 1 to agent m .

We give an example of such a group of worlds that represents assignment $\alpha = \{x_1 \mapsto \text{T}, x_2 \mapsto \text{F}, x_3 \mapsto \text{T}, x_4 \mapsto \text{T}, x_5 \mapsto \text{F}, x_6 \mapsto \text{T}\}$. 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 symmetric, we draw relations as lines (leaving out arrows at the end).



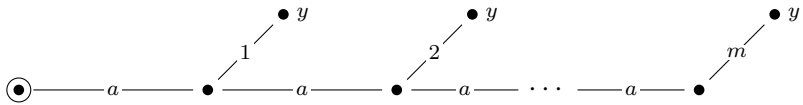
A Boolean formula $\psi(x_1, \dots, x_6)$ is true under assignment α , if in the model above $[\psi]$ is true – where $[\psi]$ is the following adaptation of formula ψ . For $1 \leq i \leq 6$, every occurrence of x_i in ψ is replaced by $\hat{B}_a \hat{B}_i y$. We refer to worlds w_1, \dots, w_4 as the *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 .

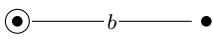
The final updated model will contain such a group of worlds for exactly every possible assignment to the variables in the given formula. Between the different groups, there are no R_a -relations, only R_b -relations. Between the worlds in a given group there are no R_b -relations, only R_a -relations. So ‘jumping’ from one group to another can be done only with an R_b -relation, while jumping between worlds within a group can be done only with an R_a -relation. To illustrate how this reduction works, we give an example for the formula $\psi = x_1 \wedge x_2$. Figure 4.3 shows the final updated model, in which all truth assignments to $\{x_1, x_2\}$ are represented. 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 (i.e., the red worlds) 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. Now, checking whether $x_1 \wedge x_2$ is satisfiable can be done by checking whether $\hat{B}_b[x_1 \wedge x_2] = \hat{B}_b(\hat{B}_a \hat{B}_1 y \wedge \hat{B}_a \hat{B}_2 y)$ is true in the model in Figure 4.3.

We now formally specify the polynomial-time reduction R . Let φ be a Boolean formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. First, we define the following polynomial-time computable mappings. For $1 \leq i \leq m$, let $[x_i] = \hat{B}_i y$. Then $[\varphi]$ is the adaptation of formula φ , where every occurrence of x_i in φ is replaced by $\hat{B}_a[x_i]$.

We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_m, \hat{B}_b[\varphi])$, where:

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

- $s_0 = ((W, V, R), W_d) =$ 

- $a_1 = ((E, Q, \text{pre}, \text{post}), e_1) =$  $e_1 : \langle \top, \top \rangle$ $e_2 : \langle \neg \hat{B}_1 y \vee y, \top \rangle$

⋮

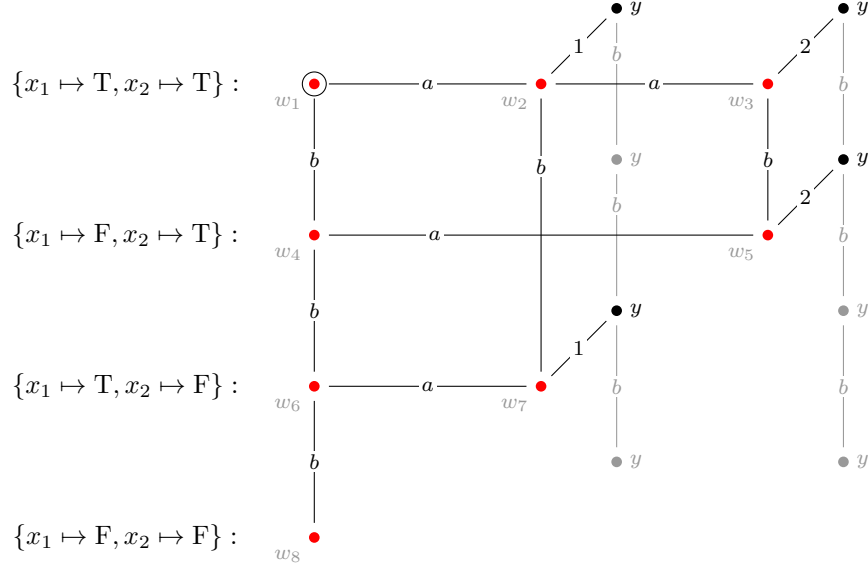


Figure 4.3: Example for the reduction in the alternative proof of Proposition 1; a final updated model for the formula $\psi = x_1 \wedge x_2$.

$$\bullet \quad a_m = \begin{array}{ccc} \begin{array}{c} \downarrow \\ \bullet \\ \downarrow \end{array} & \text{---} b \text{---} & \begin{array}{c} \downarrow \\ \bullet \\ \downarrow \end{array} \\ e_1 : \langle \top, \top \rangle & & e_2 : \langle \neg \hat{B}_m y \vee y, \top \rangle \end{array}$$

We illustrate how the actions develop the final updated model, which has a corresponding group of worlds for each particular truth assignments over $var(\varphi)$. For each i , let s_i be $s_{i-1} \otimes a_i$. For each action a_i , the first event in the action will copy the updated model so far, i.e., s_{i-1} . The second event will again copy s_{i-1} , except for the bottom worlds that represent variable x_i , i.e., the worlds with an outgoing i -arrow to a world where proposition y is true. The first part of the precondition in e_2 , i.e., formula $\neg \hat{B}_1 y$, makes sure that these particular bottom worlds are not copied. The second part of the precondition, i.e., proposition y , makes sure that all the worlds where y is true are copied. Without this second part all the worlds where y is true would not be copied, because in all of them $\neg \hat{B}_1 y$ is false.

We show that $\varphi \in \text{SAT}$ if and only if $R(\varphi) \in \text{DBU}$. Assume that φ is satisfiable. Then there is some assignment α over $var(\varphi)$ that satisfies φ . By construction, in the final updated model there will be some group of worlds that represents assignment α . The bottom worlds in this group will be R_b -accessible from the designated world and they will make $[\varphi]$ true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \models \hat{B}_b[\varphi]$.

Assume that φ is not satisfiable. Then there is no assignment α over $var(\varphi)$ that satisfies φ . By construction, all assignments α will be represented by a group of worlds in the final updated model (and the bottom worlds in these groups are the only worlds that are R_b -accessible from the

designated state). Furthermore, none of the bottom worlds in these groups will make $[\varphi]$ true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \not\models \hat{B}_b[\varphi]$.

Since this reduction runs in polynomial time, we can conclude that DBU is NP-hard. \square

For our next proof we define quantified Boolean formulas and the associated problem TRUE QUANTIFIED BOOLEAN FORMULA (TQBF). A *(fully) quantified Boolean formula (QBF)* is a formula of the form $Q_1x_1Q_2x_2\dots Q_mx_m.\psi$, where each quantifier Q_i is either \forall or \exists , the x_i are propositional variables, and ψ is a Boolean formula over the variables x_1, \dots, x_m .

Since there are no free variable in a (fully) quantified Boolean formula, it is a statement that is either true or false. To illustrate, the quantified Boolean formula $\forall x.\psi$ expresses that for both truth assignments α to the variable x , formula ψ is true (under α). Similarly, the quantified Boolean formula $\exists x.\psi$ expresses that there exists a truth assignment α to the variable x , formula ψ is true (under α).

Then the decision problem TQBF is defined as follows. This problem is PSPACE-complete (Stockmeyer & Meyer, 1973).

TQBF

Instance: A quantified Boolean formula $\varphi = Q_1x_1Q_2x_2\dots Q_mx_m.\psi$.

Question: Is φ true?

Theorem 4. DBU is PSPACE-hard.

Proof. To show PSPACE-hardness, we specify a polynomial-time reduction R from TQBF to DBU. First, we sketch the general idea behind the reduction. Essentially, this reduction works in the same way as the reduction in the alternative proof of Proposition 1. Here we add a trick to be able to adequately translate the quantifiers in front of the Boolean formula into our model. Let φ be a Boolean formula with $var(\varphi) = \{x_1, \dots, x_m\}$. Again, we use the reduction to list all possible assignments to $var(\varphi)$. Just as in the original reduction, this 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 φ will be represented by a group of worlds. 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).

The main difference is that in the original reduction, ‘jumping’ from one group to another could only be done by R_b , while in this reduction, this can only be done by 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 in that group 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 B_i and every existential quantifier Q_j to \hat{B}_j .

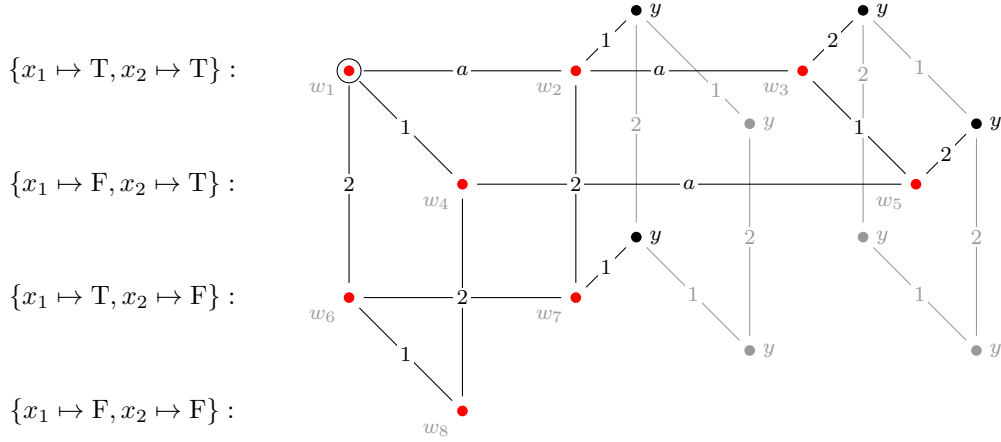


Figure 4.4: Example for the reduction in the proof of Theorem 4; a final updated model for a quantified Boolean formula with variables x_1 and x_2 .

To illustrate how this reduction works, we give an example. Figure 4.4 shows the final updated model for a quantified Boolean formula with variables x_1 and x_2 . We will represent variable x_1 by \hat{B}_1y and variable x_2 by \hat{B}_2y . Then, 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 in the model in Figure 4.4, 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 $\varphi = Q_1 x_1 \dots Q_m x_m \cdot \psi$ be a quantified Boolean formula with quantifiers Q_1, \dots, Q_m and $\text{var}(\psi) = \{x_1, \dots, x_m\}$. First, 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 polynomial-time reduction R . We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_m, [\varphi])$, where:

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

• $s_0 = ((W, V, R), W_d) =$

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.

$$\begin{aligned}
\bullet \quad a_1 &= ((E, Q, pre, post), e_1) = \begin{array}{ccc} \bullet & \xrightarrow{1} & \bullet \\ e_1 : \langle \top, \top \rangle & & e_2 : \langle \neg \hat{B}_1 y \vee y, \top \rangle \end{array} \\
&\vdots \\
\bullet \quad a_m &= \begin{array}{ccc} \downarrow & & \downarrow \\ \bullet & \xrightarrow{m} & \bullet \\ e_1 : \langle \top, \top \rangle & & e_2 : \langle \neg \hat{B}_m y \vee y, \top \rangle \end{array}
\end{aligned}$$

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 $Q_i x_i \dots Q_m x_m \cdot \psi$ 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 any bottom world of a group γ that agrees with α . We show that the formula $Q_j \dots Q_m \cdot \psi$ is true under α if and only if $[Q_j] \dots [Q_m][\psi]$ is true in w .

Case 1: Assume that $Q_j \dots Q_m \cdot \psi$ is true under α .

Case 1.1: Assume that $Q_j = \forall$. Then for both assignments $\alpha' \supseteq \alpha$ to the variables x_1, \dots, x_j , formula $Q_{j+1} \dots Q_m \cdot \psi$ 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 .

Case 1.2: Assume that $Q_j = \exists$. Then there is some assignment $\alpha' \supseteq \alpha$ to the variables x_1, \dots, x_j , such that $Q_{j+1} \dots Q_m \cdot \psi$ 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 some bottom world w' that is R_j -reachable from w – formula $[Q_{j+1}] \dots [Q_m][\psi]$ is true in w' . Since $[Q_j] = \hat{B}_j$, this means that $[Q_j] \dots [Q_m][\psi]$ is true in w .

Case 2: Assume that $Q_j \dots Q_m \cdot \psi$ is not true under α .

Case 2.1: Assume that $Q_j = \forall$. Then there is some assignment $\alpha' \supseteq \alpha$ to the variables x_1, \dots, x_j , such that $Q_{j+1} \dots Q_m \cdot \psi$ 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 .

Case 2.2: Assume that $Q_j = \exists$. Then for both assignments $\alpha' \supseteq \alpha$ to the variables x_1, \dots, x_j , formula $Q_{j+1} \dots Q_m.\psi$ 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 all bottom worlds w' that are R_j -reachable from w – formula $[Q_{j+1}] \dots [Q_m][\psi]$ is true in w' . Since $[Q_j] = \hat{B}_j$, this means that $[Q_j] \dots [Q_m][\psi]$ is not true in w .

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 5. *DBU is PSPACE-complete.*

Proof. To show PSPACE-membership, we specify an algorithm that solves DBU in polynomial space. In order to do so, we introduce a few additional constructs to the language \mathcal{L}_B , which have been considered before (in a slight notational variant) in the context of DEL (cf. van Ditmarsch et al., 2008). We add the following two operators, where $\mathcal{E} = (E, Q, \text{pre}, \text{post})$ is an event model, $e \in E$, and $E_d \subseteq E$. If φ is a formula in the language, then $[\mathcal{E}, e]\varphi$ and $[\mathcal{E}, E_d]\varphi$ are also formulas in the language. We define the semantics of these operators as follows:

$$\begin{aligned} \mathcal{M}, w \models [\mathcal{E}, E_d]\varphi & \text{ iff } \mathcal{M}, w \models [\mathcal{E}, e]\varphi \text{ for all } e \in E_d; \\ \mathcal{M}, w \models [\mathcal{E}, e]\varphi & \text{ iff } \mathcal{M}, w \models \text{pre}(e) \text{ implies } \mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi. \end{aligned}$$

Intuitively, these operators express that a formula holds in the model after updating with a particular event model, i.e., an action. With these additional operators, we can express the problem DBU as a particular case of checking whether a formula $\varphi \in \mathcal{L}_B$ is true in a given model (\mathcal{M}, W_d) . Let $x = (P, s_0, a_1, \dots, a_m, \varphi)$, where $s_0 = (\mathcal{M}, w)$ and where $a_i = (\mathcal{E}_i, E_d^i)$ for each $1 \leq i \leq m$. Then $x \in \text{DBU}$ if and only if $\mathcal{M}, w \models [\mathcal{E}_1, E_d^1] \dots [\mathcal{E}_m, E_d^m]\varphi$. This follows directly from the definition of DBU and the truth definitions of the operators introduced above.

We will describe an algorithm to solve the problem of deciding whether a given (single-pointed) epistemic model (\mathcal{M}, w) satisfies a given formula in \mathcal{L}_B (possibly including the newly defined constructs) in polynomial space – this is also called the model checking problem of DEL. If we want to decide whether a multi-pointed epistemic model (\mathcal{M}, W_d) satisfies a given DEL formula φ , we can simply call the algorithm for each $w \in W_d$. Therefore, this suffices to show that DBU is in PSPACE. Additionally, this shows that model checking for DEL (with postconditions) can be done in polynomial space.

The algorithm that we describe is similar to the algorithm given by Aucher & Schwarzentruber (2013). We slightly modify the presentation of the algorithm from their presentation, to match the

notation that we use. Additionally, we modify their algorithm to take into account postconditions in the event models and to take into account multi-pointed event models (in our notation).

The algorithm is given in Algorithm 4.5. We analyze the algorithm and its complexity. The algorithm M-Check takes three arguments: the first argument is the (single-pointed) model (\mathcal{M}, w) ; the second argument is a sequence $\langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle$ of (single-pointed) event models (\mathcal{E}_j, e_j) , for $1 \leq j \leq i$; the third argument is a DEL formula φ (possibly including the additional operators). The algorithm checks whether the formula φ is true in $\mathcal{M} \otimes \mathcal{E}_1 \otimes \dots \otimes \mathcal{E}_i, (w, e_1, \dots, e_i)$. The background assumption for calling the algorithm is that $(w, e_1, \dots, e_i) \in \mathcal{M} \otimes \mathcal{E}_1 \otimes \dots \otimes \mathcal{E}_i$; this condition is maintained throughout the algorithm as an invariant.

Termination of the algorithm follows from the fact that for each recursive call, the following size measure μ of the input strictly decreases:

$$\mu(x) = |\mathcal{M}| + \sum_{j=1}^i |\mathcal{E}_j| + |\varphi|.$$

Here $x = ((\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle, \varphi)$ denotes the input for the algorithm. Correctness of the algorithm follows straightforwardly from the truth definitions of the language constructs. All that remains is to show that the algorithm requires only polynomial space. As the input of the size strictly decreases with each recursive call, there are only linearly many recursive calls in the call stack at any point. Each of these recursive calls needs only a polynomial amount of space for storing the values of the local variables (e.g., the variables used in the for-loops in the algorithm). Therefore, the algorithm runs in polynomial space. \square

We consider the following problem, which is a special case of DBU.

DBU-NO-POST – DYNAMIC BELIEF UPDATE WITHOUT POSTCONDITIONS

Instance: A set of propositions P and a set of agents \mathcal{A} . An initial state s_o , where $s_o = ((W, V, R), W_d)$ is an 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 with postcondition \top . A formula $\varphi \in \mathcal{L}_B$.

Question: Does $s_o \otimes a_1 \otimes \dots \otimes a_k \models \varphi$?

Corollary 6. DBU-NO-POST is PSPACE-complete.

Proof. Since DBU-NO-POST is a special case of DBU, Algorithm 4.5 can also be used to solve DBU-NO-POST in polynomial space. Thus $\text{DBU} \in \text{PSPACE}$. Furthermore, since the reduction from TQBF to DBU in the proof of Theorem 4 does not make use of postconditions, i.e., all actions given by the reduction have postcondition \top , it is also a reduction from TQBF to DBU-NO-POST. \square

```

function M-Check( $(\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle, \varphi$ )
switch  $\varphi$  do
  case  $p$ 
    for  $j \in \{i, i-1, \dots, 1\}$  do
      if  $p \in \text{post}(e_j)$  then
        return true;
      else if  $\neg p \in \text{post}(e_j)$  then
        return false;
    return  $V(w, p)$ ;
  case  $\neg\psi$ 
    return not M-Check( $(\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle, \psi$ );
  case  $\psi_1 \wedge \psi_2$ 
    return M-Check( $(\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle, \psi_1$ ) and
      M-Check( $(\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle, \psi_2$ );
  case  $B_a\psi$ 
    for  $u \in R_a(w)$  do
      for  $u_1 \in R_a(w_1)$  do
        if M-Check( $(\mathcal{M}, u), \langle \rangle, \text{pre}(u_1)$ ) then
          for  $u_2 \in R_a(e_2)$  do
            if M-Check( $(\mathcal{M}, u), \langle \mathcal{E}'_1, u_1 \rangle, \text{pre}(u_2)$ ) then
               $\vdots$ 
              for  $u_i \in R_a(e_i)$  do
                if M-Check( $(\mathcal{M}, u), \langle \mathcal{E}'_1, u_1; \dots; \mathcal{E}_{i-1}, u_{i-1} \rangle, \text{pre}(u_i)$ ) then
                  if not M-Check( $(\mathcal{M}, u), \langle \mathcal{E}'_1, u_1; \dots; \mathcal{E}_i, u_i \rangle, \psi$ ) then
                    return false;
              return true;
          return true;
      return true;
  case  $[\mathcal{E}', e]\psi$ 
    if M-Check( $(\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle, \text{pre}(e)$ ) then
      return M-Check( $(\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i; \mathcal{E}', e \rangle, \psi$ );
    else
      return true;
  case  $[\mathcal{E}', E_d]\psi$ 
    for  $e' \in E_d$  do
      if not M-Check( $(\mathcal{M}, w), \langle \mathcal{E}_1, e_1; \dots; \mathcal{E}_i, e_i \rangle, [\mathcal{E}', e']\varphi$ ) then
        return false;
    return true;

```

Figure 4.5: Polynomial-space algorithm for DEL model checking.

The results in Theorems 4 and 5 and Corollary 6 resolve an open question in the literature about the computational complexity of DEL. Aucher & Schwarzentruber (2013) already showed that the model checking problem for DEL, in general (that is, without any restrictions on the models), is PSPACE-complete. However, their result for PSPACE-hardness does not work when the input is restricted to S5 (or KD45) models. Moreover, their hardness proof also relies on the use of multi-pointed models (which in their notation is captured by means of a union operator). They leave the question whether model checking for DEL restricted to S5 models and without the use of multi-pointed models has the same complexity as an open problem. In Theorem 4 and Corollary 6 we answered this question by showing that DEL model checking is PSPACE-complete even when restricted to single-pointed S5 models.

Furthermore, in their PSPACE-membership result, Aucher & Schwarzentruber only deal with trivial postconditions, i.e., the case where all postconditions are \top . Hardness carries over to the case where non-trivial postconditions are allowed. However, it is not self-evident that membership in PSPACE would also hold for this more general case. In Theorem 5, we extended Aucher & Schwarzentruber’s polynomial-space algorithm to handle non-trivial postconditions as well.

4.3 Parameterized Complexity Results

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 4.1. 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:

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

For the size of a formula we count the number of occurrences of propositional variables and

Parameter	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
p	number of propositions in P
u	number of actions

Table 4.1: Overview of the different parameters for DBU.

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

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

4.3.1 Intractability Results

In the following, we will identify several parameterized versions of DBU that are fixed-parameter intractable. 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 to give stronger intractability results. For instance, the proof of Theorem 4 already shows that $\{p\}$ -DBU is para-PSPACE hard, since the reduction in this proof uses a constant number of propositions. However, since in this thesis 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 we restrict ourselves to showing $W[1]$ -hardness and para-NP-hardness. This is also the reason why we will not show completeness for any of the (parameterized) intractability classes; showing hardness is enough to indicate intractability.

Corollary 7. $\{a, c, e, f, o\}$ -DBU is para-NP-hard.

Proof. To show para-NP-hardness, it suffices to show that DBU is NP-hard for a constant value of the parameters (Flum & Grohe, 2003). Parameters a , c , e , f , and o – respectively, the number of agents, the maximum size of the preconditions, the maximum number of events in the actions (the event models), the size of the formula, and the modal depth of the formula – have constant values

in the proof of Theorem 1 (namely $a = 1, e = 2, c = 10, f = 4, o = 1$). Hence, the proposition follows. \square

For our next result we consider the following parameterized problem $\{k\}$ -WSAT[2CNF]. (We already introduced this problem in Section 4.1.2; we repeat it here for the sake of clarity.) This problem is W[1]-complete (Downey & Fellows, 1995).

$\{k\}$ -WSAT[2CNF]
Instance: A Boolean formula φ in 2CNF and an integer k .
Parameter: k .
Question: Is there an assignment $V : \text{var}(\varphi) \rightarrow \{0, 1\}$ that sets k variables in $\text{var}(\varphi)$ to true and that satisfies φ ?

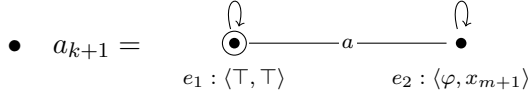
Proposition 8. $\{a, f, o, u\}$ -DBU is W[1]-hard.

Proof. To show W[1]-hardness, we specify an fpt-reduction R from $\{k\}$ -WSAT[2CNF] to $\{a, f, o, u\}$ -DBU. Here, a is the number of agents, f the size of the formula, o the modal depth of the formula and u the number of updates, i.e., the number of actions. Let φ be an arbitrary 2CNF formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. The idea behind this reduction is similar to the proof of Proposition 1. Here, we list all possible assignments α to $\text{var}(\varphi)$ that set k variables to true.

Next, we formally specify the fpt-reduction. We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_{k+1}, \hat{B}_a x_{m+1})$, where:

- $P = \{x_1, \dots, x_m, x_{m+1}\}$, where x_{m+1} is a proposition that does not occur in φ ;
- $\mathcal{A} = \{a\}$;
- $s_0 = ((W, V, R), W_d) = \begin{array}{c} \curvearrowright \\ \bullet \\ w_1 \end{array}$
- $a_1 = ((E, Q, pre, post), e_1) = \begin{array}{c} \curvearrowright \qquad \qquad \qquad \curvearrowright \\ \bullet \text{---} a \text{---} \bullet \text{---} a \text{---} \dots \text{---} a \text{---} \bullet \\ e_1 : \langle \neg x_1, x_1 \rangle \qquad e_2 : \langle \neg x_2, x_2 \rangle \qquad e_m : \langle \neg x_m, x_m \rangle \end{array}$
- \vdots
- $a_k = \begin{array}{c} \curvearrowright \qquad \qquad \qquad \curvearrowright \qquad \qquad \qquad \curvearrowright \\ \bullet \text{---} a \text{---} \bullet \text{---} a \text{---} \dots \text{---} a \text{---} \bullet \\ e_1 : \langle \neg x_1, x_1 \rangle \qquad e_2 : \langle \neg x_2, x_2 \rangle \qquad e_m : \langle \neg x_m, x_m \rangle \end{array}$

(Note that the states e_1, \dots, e_m in the event models a_1, \dots, a_k are entirely connected by an equivalence relation.)



We show that $\varphi \in \{k\}$ -WSAT[2CNF] if and only if $R(\varphi) \in \{a, f, o, u\}$ -DBU.

First, we note that the intermediate state $s_{f'} = s_o \otimes a_1 \otimes \cdots \otimes a_k$ consists of $O(|P|^k)$ worlds that are all connected. Furthermore, for each possible assignment $\alpha : \text{var}(\varphi) \rightarrow \{0, 1\}$ that sets k variables in $\text{var}(\varphi)$ to true, there is at least one world in $s_{f'}$, such that the valuation over P in that world corresponds with α . Action a_{k+1} makes sure that in the same way as in $s_{f'}$, for each possible assignment $\alpha : \text{var}(\varphi) \rightarrow \{0, 1\}$ that sets k variables in $\text{var}(\varphi)$ to true, there is at least one world in the final state $s_f = s_{f'} \otimes a_{k+1}$, such that the valuation over P in that world corresponds with α . Furthermore, in each world (resulting from e_2), x_{m+1} is true if and only if the valuation over P in that world is a satisfying assignment for φ that sets k variables in $\text{var}(\varphi)$ to true. Also, all worlds in s_f are connected. Therefore, $\varphi \in \{k\}$ -WSAT[2CNF] if and only if $R(\varphi) \in \{a, f, o, u\}$ -DBU.

Since this reduction runs in polynomial time, parameter u depends only on parameter k (namely $u = k + 1$), and parameters a, c, o have a constant value (namely value 1), we can conclude that $\{a, f, o, u\}$ -DBU is W[1]-hard. \square

Corollary 9. $\{a, c, o, u\}$ -DBU is W[1]-hard.

Proof. We can modify the reduction in the proof of Proposition 8 into an fpt-reduction from $\{k\}$ -WSAT[2CNF] to $\{a, c, o, u\}$ -DBU. We omit proposition x_{m+1} and action a_{k+1} , and replace the formula $\hat{B}_i x_{m+1}$ with $\hat{B}_i \varphi$. By this modification, parameter f – the size of the formula – no longer has a constant value, but parameter c – the maximum size of the preconditions – does. Then $\varphi \in \{k\}$ -WSAT[2CNF] if and only if $R(\varphi) \in \{a, c, o, u\}$ -DBU, parameter u depends only on parameter k (namely $u = k$), and parameters a, c, o have a constant value (namely value 1). \square

In fact, Proposition 8 and Corollary 9 both follow from Proposition 10, which we will prove next. However, since the proof of Proposition 10 is rather intricate, we decided to also incorporate Proposition 8 with a separate proof that is easier to read. We consider the following parameterized problem, that we will use in our proof of Proposition 10. This problem is W[1]-complete (Fellows et al., 2009).

$\{k\}$ -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 10. $\{a, c, f, o, u\}$ -DBU is W[1]-hard.

Proof. We specify an fpt-reduction R from $\{k\}$ -MULTICOLORED CLIQUE to $\{a, c, f, o, u\}$ -DBU. Here, a is the number of agents, c the maximum size of the preconditions, f the size of the formula, o the modal depth of the formula and u the number of updates, i.e., the number of actions. Let (G, c) be an instance of $\{k\}$ -MULTICOLORED CLIQUE, where $G = (N, E)$. The 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 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 where for each possible edge between k differently colored nodes its corresponding proposition is true. This is only the case when G has a k -clique with k different colors.

Next, we specify the reduction. We let $R(G, c) = (P, \mathcal{A}, s_0, a_1, \dots, a_k, a_{k+1}, \dots, a_{k+\binom{k}{2}}, \hat{B}_a \varphi)$, where:

- $P = \{x_1, \dots, x_{|N|}, z\} \cup \{r_{ij}; 1 \leq i < j \leq k\}$. We define propositions x_i for each vertex in N , a proposition z and propositions r_{ij} for all colors i, j such that $i < j$.
- $\mathcal{A} = \{a\}$.

- $s_0 = ((W, V, R), W_d) = \begin{array}{c} \curvearrowright \\ \bullet \\ w_1 \end{array}$

Let $D_1 = \{d_{11}, \dots, d_{1u_1}\} = \{x_i \in P; c(x_i) = 1\}, \dots, D_k = \{d_{k1}, \dots, d_{ku_k}\} = \{x_i \in P; c(x_i) = k\}$. The set D_j contains all propositions that correspond with a node in N with color j , and u_j is the number of nodes in N that have color j .

- $a_1 = ((E, Q, pre, post), e_1) = \begin{array}{ccccccc} \curvearrowright & & \curvearrowright & & & & \curvearrowright \\ \bullet & \text{---} a \text{---} & \bullet & \text{---} a \text{---} & \dots & \text{---} a \text{---} & \bullet \\ e_1 : \langle \top, d_{11} \rangle & & e_2 : \langle \top, d_{12} \rangle & & & & e_{u_1} : \langle \top, d_{1u_1} \rangle \end{array}$

⋮

- $a_k = \begin{array}{ccccccc} \curvearrowright & & \curvearrowright & & & & \curvearrowright \\ \bullet & \text{---} a \text{---} & \bullet & \text{---} a \text{---} & \dots & \text{---} a \text{---} & \bullet \\ e_1 : \langle \top, d_{k1} \rangle & & e_2 : \langle \top, d_{k2} \rangle & & & & e_{u_k} : \langle \top, d_{ku_k} \rangle \end{array}$

Let $t = \binom{|N|}{2}$. Let the set of all possible edges between the vertices in N be $E' = \{\{x'_1, x''_1\}, \dots, \{x'_t, x''_t\}\}$. (Note that x'_ℓ and x''_ℓ are names for propositions of the form x_i .) For each $\{x'_l, x''_l\} \in E'$ we define:

$$\psi_l = \begin{cases} r_{ij} & \text{if } \{x'_l, x''_l\} \in E \text{ and } c(x'_l) = i, c(x''_l) = j, \\ z & \text{otherwise.} \end{cases}$$

$$\bullet \quad a_{k+1} = \begin{array}{c} \begin{array}{ccccccc} \circlearrowleft & & \circlearrowleft & & \circlearrowleft & & \circlearrowleft \\ \bullet & \text{---} a \text{---} & \bullet & \text{---} a \text{---} & \bullet & \text{---} a \text{---} & \dots & \text{---} a \text{---} & \bullet \\ e_0 : \langle \top, \top \rangle & e_1 : \langle x'_1 \wedge x''_1 \wedge \neg\psi_1, \psi_1 \rangle & e_2 : \langle x'_2 \wedge x''_2 \wedge \neg\psi_2, \psi_2 \rangle & & & & & e_t : \langle x'_t \wedge x''_t \wedge \neg\psi_t, \psi_t \rangle \end{array} \\ \vdots \end{array}$$

$$\bullet \quad a_{k+\binom{k}{2}} = \begin{array}{c} \begin{array}{ccccccc} \circlearrowleft & & \circlearrowleft & & \circlearrowleft & & \circlearrowleft \\ \bullet & \text{---} a \text{---} & \bullet & \text{---} a \text{---} & \bullet & \text{---} a \text{---} & \dots & \text{---} a \text{---} & \bullet \\ e_0 : \langle \top, \top \rangle & e_1 : \langle x'_1 \wedge x''_1 \wedge \neg\psi_1, \psi_1 \rangle & e_2 : \langle x'_2 \wedge x''_2 \wedge \neg\psi_2, \psi_2 \rangle & & & & & e_t : \langle x'_t \wedge x''_t \wedge \neg\psi_t, \psi_t \rangle \end{array} \end{array}$$

(Note that all the states in event models $a_1, \dots, a_k, a_{k+1}, \dots, a_{k+t}$ are entirely connected by an equivalence relation.)

$$\bullet \quad \varphi = \bigwedge_{1 \leq i < j \leq k} r_{ij}$$

We show that $(G, c) \in \{k\}$ -MULTICOLORED CLIQUE if and only if $R(G, c) \in \{a, c, f, o, u\}$ -DBU.

First we note that in the intermediate state $s_{f'} = s_o \otimes a_1 \otimes \dots \otimes a_k$, for each combination of k vertices with k different colors, there is at least one world w in $s_{f'}$ in which the valuation over P corresponds with that combination of vertices. The same holds for the final state, $s_f = s_o \otimes a_1 \otimes \dots \otimes a_k \otimes a_{k+1} \otimes \dots \otimes a_{k+\binom{k}{2}}$. Furthermore, by construction, in the final state s_f , proposition $\psi_l = r_{ij}$ is true in world w only if x'_l and x''_l are true in w and $\{x'_l, x''_l\}$ is an edge between vertices x'_l and x''_l in G . Hence, we have that φ is true in a world w , only if the k vertices that correspond with world w are all connected by an edge. Conversely, if there is a world w corresponding to a set of k vertices with different colors that are all connected to each other by an edge in E , then in this world all propositions r_{ij} are true. Therefore, G has a k -colored clique if and only if $s_f \models \hat{B}_a \varphi$. Thus, we can conclude that $(G, c) \in \{k\}$ -MULTICOLORED CLIQUE if and only if $R(G, c) \in \{a, c, f, o, u\}$ -DBU.

Since this reduction runs in polynomial time, parameters a , c and o have a constant value (namely $a = 1$, $c = 6$, and $o = 1$), and parameters f and u depend only on parameter k (namely $f = 2\binom{k}{2} - 1$ and $u = k + \binom{k}{2}$), we can conclude that $\{a, c, f, o, u\}$ -DBU is $W[1]$ hard. \square

Corollary 11. $\{c, e, o, p\}$ -DBU is para-NP-hard.

Proof. To show para-NP-hardness, it suffices to show that DBU is NP-hard for a constant value of the parameters (Flum & Grohe, 2003). Since parameters c , e , o , and p – respectively, the maximum size of the preconditions, the maximum number of events in the actions (the event models), the modal depth of the formula and the number of propositions – have constant values in the alternative proof of Proposition 1 (namely $c = 13$, $e = 2$, $o = 3, 11$ and $p = 1$), we can conclude that $\{c, e, o, p\}$ -DBU is para-NP-hard. \square

Proposition 12. $\{c, e, f, o, p\}$ -DBU is para-NP-hard.

Proof. To show para-NP-hardness, it suffices to show that DBU is NP-hard for a constant value of the parameters (Flum & Grohe, 2003). We can modify the reduction in the alternative proof of

Proposition 1 in such a way that in addition to parameters c , e , o , and p – respectively, the maximum size of the preconditions, the maximum number of events in the actions (the event models), the modal depth of the formula and the number of propositions – also parameter f – the size of the formula – has a constant value. First, we explain the general idea of this modification. Then, we formally specify the modified reduction.

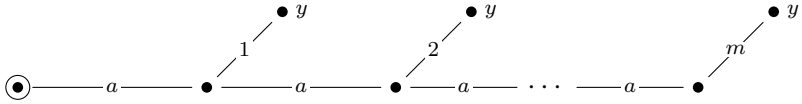
We add a trick to the reduction in the alternative proof of Proposition 1, to also keep the size of the formula that we check in the final updated model, of constant size. We do this by checking the satisfiability of the formula clause by clause. For each clause in a given 3CNF formula φ , we add one action. This action (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.

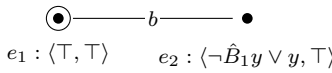
Now, we specify the fpt-reduction R from SAT to DBU, where parameters c , e , o , p , and f have constant values. Let φ be a Boolean formula with $var(\varphi) = \{x_1, \dots, x_m\}$. Without loss of generality we assume that φ is a 3CNF formula with clauses c_1 to c_l . We use similar polynomial-time computable mappings as in the reduction in the alternative proof of Proposition 1. For $1 \leq i \leq m$, let $[x_i] = \hat{B}_i y$. Then $[c_j]$ is the adaptation of clause c_j , where every occurrence of x_i in c_j is replaced by $\hat{B}_i[x_i]$.

Next, we formally specify the reduction R . We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_{m+l}, \hat{B}_b \hat{B}_c z)$, where:

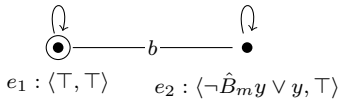
- $P = \{y, z\}$;

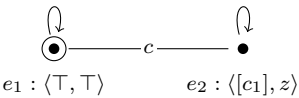
- $\mathcal{A} = \{a, b, c, 1, \dots, m\}$;

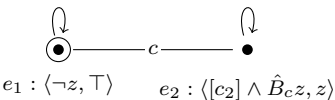
- $s_0 = ((W, V, R), W_d) =$ 

- $a_1 = ((E, Q, pre, post), e_1) =$ 

⋮

- $a_m =$ 

- $a_{m+1} =$ 

- $a_{m+2} =$ 

$$\begin{array}{c}
\vdots \\
\bullet \quad a_{m+l} = \begin{array}{ccc}
\begin{array}{c} \Downarrow \\ \bullet \end{array} & \xrightarrow{c} & \begin{array}{c} \Downarrow \\ \bullet \end{array} \\
e_1 : \langle \neg z, \top \rangle & & e_2 : \langle [c_m] \wedge \hat{B}_c z, z \rangle
\end{array}
\end{array}$$

We show that $\varphi \in \text{SAT}$ if and only if $R(\varphi) \in \text{DBU}$.

Assume that φ is satisfiable. Then there is some assignment α over $\text{var}(\varphi)$ that satisfies clauses c_1 tot c_j . By construction, in the final updated model there will be exactly one group of worlds that represents assignment α . One of the bottom worlds in this group will be R_b -accessible from the designated world and it will make clauses c_1 tot c_j true. Action a_{m+1} to a_{m+l} make sure that this world has an R_c relation to a world where proposition z is true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \models \hat{B}_b \hat{B}_c z$.

Assume that φ is not satisfiable. Then there is no assignment α over $\text{var}(\varphi)$ that satisfies φ . By construction, all assignments α will be represented by a group of worlds in the final updated model (and every world that is R_b -accessible from the designated world is part of one of these groups). Furthermore, none of the bottom worlds in these groups will make clause c_1 to c_l true. This means that the precondition of the second event (e_2) of some action a_{m+1} to a_{m+l} is not satisfied by the model and therefore there will be no world in the model where proposition z is true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \not\models \hat{B}_b \hat{B}_c z$.

Since this reduction runs in polynomial time and parameters c , e , o , and p have constant values (namely $c = 10$, $e = 2$, $o = 3$, and $p = 2$), we can conclude that $\{c, e, o, p\}$ -DBU is para-NP-hard. \square

Proposition 13. $\{a, e, f, o, p\}$ -DBU is para-NP-hard.

Proof. To show para-NP-hardness, it suffices to show that DBU is NP-hard for a constant value of the parameters (Flum & Grohe, 2003). We specify a polynomial-time reduction R from SAT to DBU, where parameters a , e , f , o and p – respectively, the number of agents, the maximum number of events in the actions (the event models), the size of the formula, the modal depth of the formula and the number of propositions – have constant values. Let φ be a Boolean formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. Without loss of generality we assume that m is even. (This assumption has no structural function in the proof, but we use it to simplify the description of initial state s_0 .) The reduction is based on the same principle as the one used in the alternative proof of Proposition 1. 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 .

To illustrate this encoding of the variables, we give an example. Figure 4.6 shows the final updated model for formula $\psi = x_1 \vee x_2$. Checking whether ψ is satisfiable can be done by checking whether $\hat{B}_b(\hat{B}_a \hat{B}_b y \vee \hat{B}_a(\hat{B}_b \hat{B}_a y \wedge \neg \hat{B}_b y))$ is true in the model in Figure 4.6, which is the case.

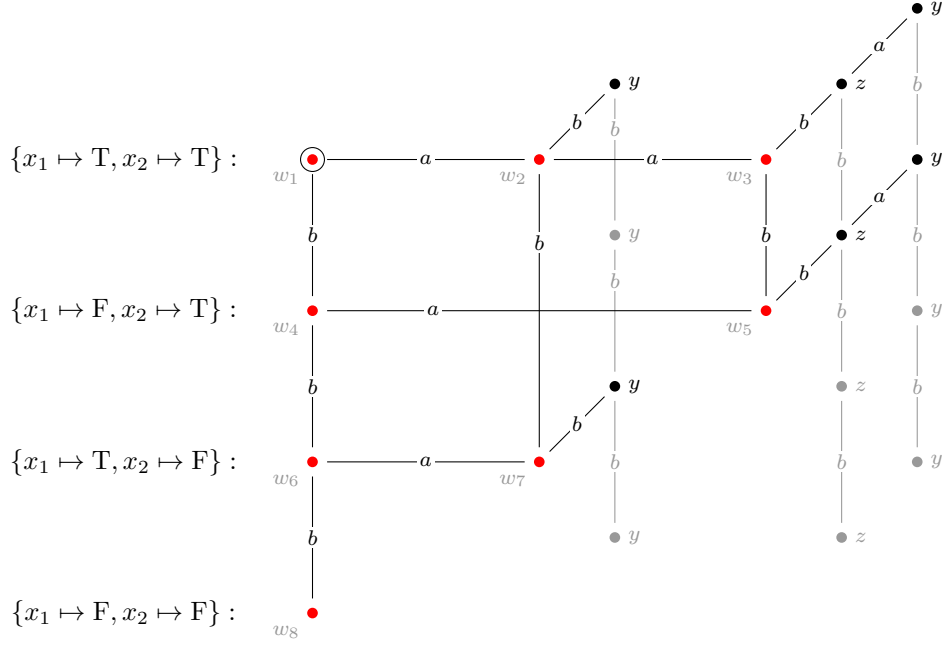


Figure 4.6: Example for the reduction in the proof of Proposition 13 (*not* including action a_{m+1}); a final updated model for formula $\psi = x_1 \vee x_2$.

Now, we continue with the formal details of the encoding of the variables and the formula. We define the following polynomial-time computable mappings. For $1 \leq i \leq m$, we define $[x_i]$ inductively as follows: $[x_1] = \hat{B}_b$,

$$[x_{j+1}] = \begin{cases} [x_j]\hat{B}_b & \text{if } [x_j] \text{ ends with } \hat{B}_a, \\ [x_j]\hat{B}_a & \text{if } [x_j] \text{ ends with } \hat{B}_b. \end{cases}$$

Then $[\varphi]$ is the adaptation of formula φ , where for $2 \leq i \leq m$, every occurrence of x_i in φ is replaced by $\hat{B}_a([x_i]y \wedge \neg[x_{i-1}]y)$ and every occurrence of x_1 is replaced by $\hat{B}_a[x_1]$. We say that a group of bottom worlds represents an assignment α to variables x_1, \dots, x_m if (1) for all x_i with $2 \leq i \leq m$ that are set to true by α , in all bottom worlds in the group $\hat{B}_a([x_i]y \wedge \neg[x_{i-1}]y)$ is true, and (2) if α sets x_1 to true, then in all bottom worlds in the group $\hat{B}_a[x_1]y$ is true.

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 in a similar way as in the proof of Proposition 8, by adding an extra action; here that is a_{m+1} . 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. Figure 4.7 shows how the final updated model for formula $\psi = x_1 \wedge x_2$ looks like when we use this extra action a_{m+1} .

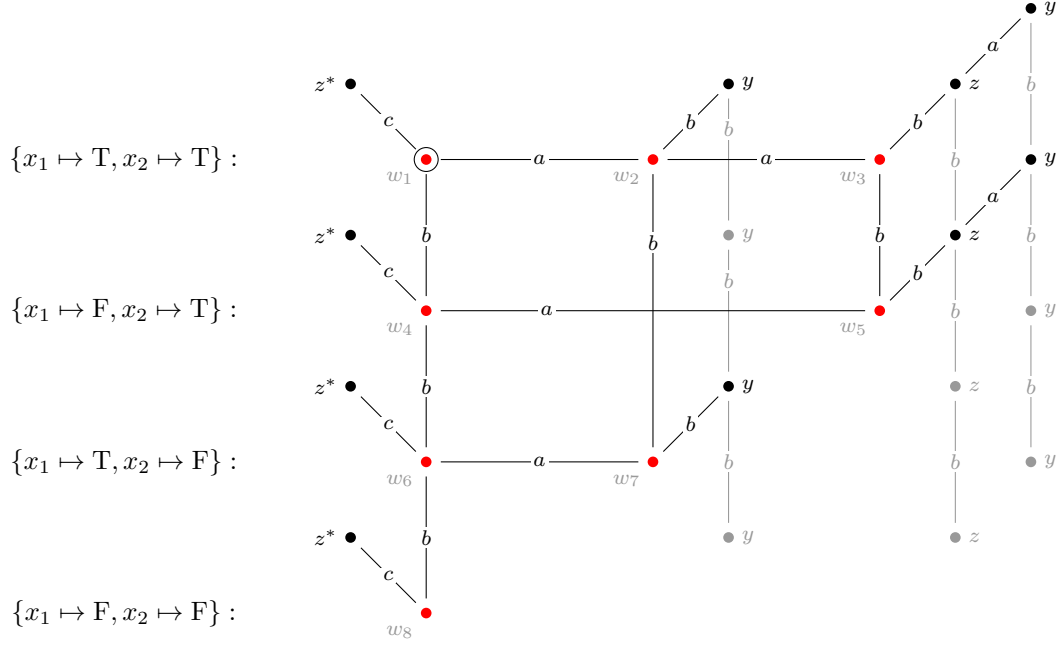
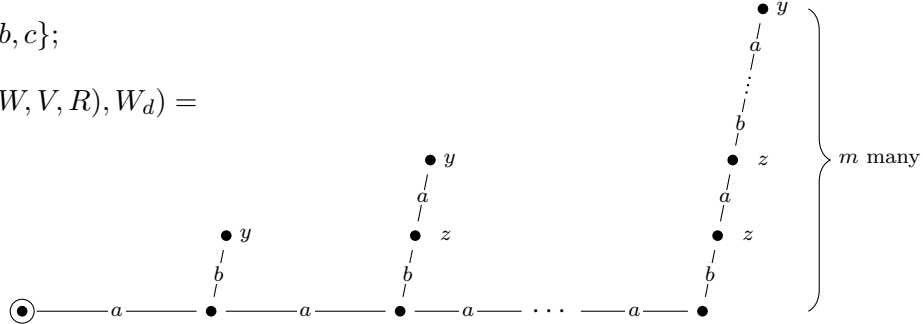


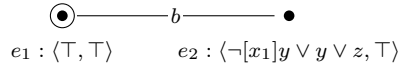
Figure 4.7: Example for the reduction in the proof of Proposition 13 (including action a_{m+1}); a final updated model for a formula $\psi = x_1 \vee x_2$.

We now formally specify the reduction. We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_m, \hat{B}_b \hat{B}_c z^*)$, where:

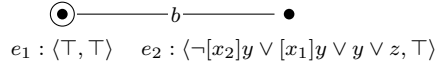
- $P = \{y, z, z^*\}$;
- $\mathcal{A} = \{a, b, c\}$;
- $s_0 = ((W, V, R), W_d) =$



- $a_1 = ((E, Q, pre, post), e_1) =$

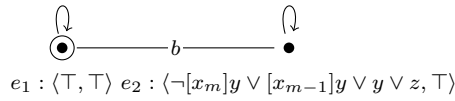


- $a_2 = ((E, Q, pre, post), e_1) =$



⋮

- $a_m =$



$$\bullet \quad a_{m+1} = \begin{array}{ccc} \begin{array}{c} \downarrow \\ \bullet \\ \downarrow \\ e_1 : \langle \top, \top \rangle \end{array} & \xrightarrow{c} & \begin{array}{c} \downarrow \\ \bullet \\ \downarrow \\ e_2 : \langle [\varphi], z^* \rangle \end{array} \end{array} .$$

We show that $\varphi \in \text{SAT}$ if and only if $R(\varphi) \in \text{DBU}$. This follows by an argument that is similar to an argument in the alternative proof of Proposition 1, which we repeat here for the sake of clarity.

Assume that φ is satisfiable. Then there is some assignment α over $\text{var}(\varphi)$ that satisfies φ . By construction, in the final updated model there will be exactly one group of worlds that represents assignment α . One of the bottom worlds in this group will be R_b -accessible from the designated world and it will make $[\varphi]$ true. Action a_{m+1} makes sure that this world has an R_c relation to a world where proposition z^* is true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \models \hat{B}_b \hat{B}_c z^*$.

Assume that φ is not satisfiable. Then there is no assignment α over $\text{var}(\varphi)$ that satisfies φ . By construction, all assignments α will be represented by a group of worlds in the final updated model (and every world that is R_b -accessible from the designated world is part of one of these groups). Furthermore, none of the bottom worlds in these groups will make $[\varphi]$ true. Then the precondition of the second event (e_2) of action a_{m+1} is not satisfied and therefore proposition z^* will not be true in any world in the model. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \not\models \hat{B}_b \hat{B}_c z^*$.

Since this reduction runs in polynomial time and parameters a, e, f, o and p have constant values (namely $a = 3, e = 2, f = 3, o = 2$ and $p = 3$), we can conclude that $\{a, e, f, o, p\}$ -DBU is para-NP-hard. \square

Proposition 14. $\{c, o, p, u\}$ -DBU is W[1]-hard.

Proof. We specify the following fpt-reduction R from $\{k\}$ -WSAT[2CNF] to $\{c, o, p, u\}$ -DBU. Here, c is the maximum size of the preconditions, o the modal depth of the formula, p the number of propositions and u the number of updates, i.e., the number of actions. We already specified an fpt-reduction from $\{k\}$ -WSAT[2CNF] to $\{a, f, o, u\}$ -DBU in the proof of Proposition 8. The reduction that we present here is based on the same principle, but then implemented in a manner that is similar to the reduction in the alternative proof of Proposition 1.

First, we sketch the general idea behind the reduction. Let φ be a Boolean 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. In the reduction from $\{k\}$ -WSAT[2CNF] to $\{a, f, o, u\}$ -DBU in the proof of Proposition 8, we listed all possible assignments to the variables in a formula that set k variables to true. Somewhat similarly, in this reduction we list all possible assignments to $\text{var}(\varphi') = \text{var}(\varphi)$ that set $m - k$ variables to true. To keep the number of propositions constant, we use the same principle as in the reduction in the alternative proof of Proposition 1; we use agents 1 to m to represent variables x_1, \dots, x_m . We represent each possible assignment to $\text{var}(\varphi)$ that sets $m - k$

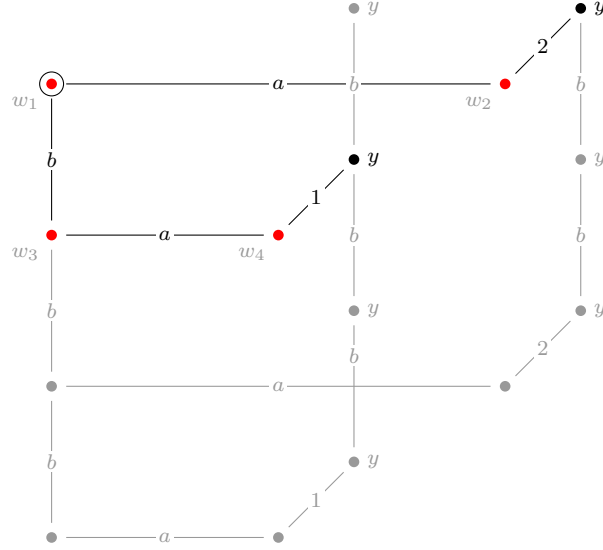


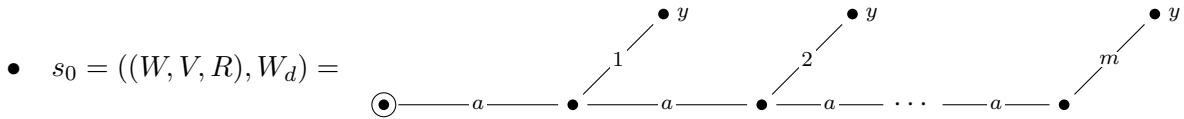
Figure 4.8: Example for the reduction in the proof of Proposition 14; a final updated model for (ψ, k) , with $\psi = x_1 \vee x_2$ and $k = 1$.

variables to true as a group of worlds. (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).

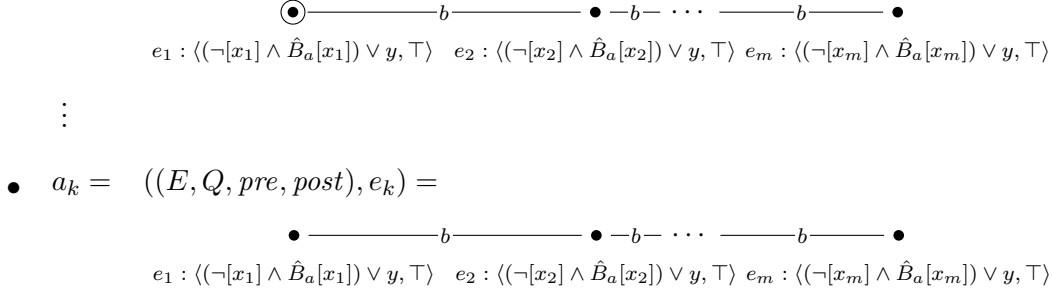
To illustrate, we give an example. Figure 4.6 shows the final updated model (the model that results from updating the initial state with the actions) for (ψ, k) , with $\psi = x_1 \vee x_2$ and $k = 1$. Checking whether φ is satisfiable by some assignment α that sets 1 variable to true, can be done by checking whether $\hat{B}_b(\neg\hat{B}_a\hat{B}_1y \vee \neg\hat{B}_a\hat{B}_2y)$ is true in the model in Figure 4.8.

We now continue with the formal details. Let φ be an arbitrary Boolean formula with $var(\varphi) = \{x_1, \dots, x_m\}$. We use similar polynomial-time computable mappings as in the reduction in the alternative proof of Proposition 1. For $1 \leq i \leq m$, let $[x_i] = \hat{B}_i y$. Then $[\varphi]$ is the adaptation of formula φ , where every occurrence of x_i in φ is replaced by $\neg\hat{B}_a[x_i]$. We let $R(\varphi) = (P, \mathcal{A}, s_0, a_1, \dots, a_k, \hat{B}_b[\varphi])$, where:

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



- $a_1 = ((E, Q, pre, post), e_1) =$



Note that in each action a_j , the designated event is e_j . This is to ensure that the actions are applicable.

We explain how the preconditions of the events in the actions work, i.e., how they make sure that the final updated model contains a corresponding group of worlds for each assignment α to $var(\varphi)$ that sets $m - k$ variables to true. For each i , let $s_i = s_{i-1} \otimes a_i$. For some action a_i , every event e_j in a_i copies all worlds in s_{i-1} where y is true and all bottom worlds in s_{i-1} that have an R_j relation to a world where y is true. Furthermore, e_j only copies groups of worlds that have a bottom world with an R_j relation to a world where y is true. This is done by means of the formula $\hat{B}_a[x_j]$ in the precondition. Otherwise, there could also be groups of worlds in the final updated model that set more than $m - k$ variables to true.

We show that $(\varphi, k) \in \{k\}\text{-WSAT}[2\text{CNF}]$ if and only if $R(\varphi, k) \in \{c, o, p, u\}\text{-DBU}$. Assume that $(\varphi, k) \in \{k\}\text{-WSAT}[2\text{CNF}]$. Then there is some assignment α that sets $m - k$ variables to true and that satisfies φ' . By construction, there is some group of worlds in the final updated model that represents α , and in all the bottom worlds of that group $[\varphi]$ is true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \models \hat{B}_b[\varphi]$.

Assume that $(\varphi, k) \notin \{k\}\text{-WSAT}[2\text{CNF}]$. Then there is no assignment α that sets $m - k$ variables to true and that satisfies φ' . By construction, for all the groups of worlds in the final updated model, in none of the bottom worlds of that group $[\varphi]$ is true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_m \not\models \hat{B}_b[\varphi]$.

Since this reduction runs in polynomial time, parameters c , o , and p have constant values (namely $c = 7$, $o = 3$, and $p = 1$) and parameter u depends only on parameter k (namely $u = k$), we can conclude that $\{c, o, p, u\}\text{-DBU}$ is $\text{W}[1]\text{-hard}$. \square

The reduction that we use to show the following result is similar to the reductions in the proofs of Proposition 8 and Proposition 14. In fact, Proposition 8 follows from the following result. For the sake of readability we include both proofs.

Proposition 15. $\{a, f, o, p, u\}\text{-DBU}$ is $\text{W}[1]\text{-hard}$.

Proof. We specify the following fpt-reduction R from $\{k\}\text{-WSAT}[2\text{CNF}]$ to $\{a, f, o, p, u\}\text{-DBU}$. We modify the latter reduction by adding some tricks to keep the values of parameters a and f – the number of agents and the size of the formula that we check in the final updated model – constant. After these modifications, the value of parameter c – the maximum size of the preconditions – will no longer be constant.

We continue with the formal details. Let φ be an arbitrary Boolean formula with $\text{var}(\varphi) = \{x_1, \dots, x_m\}$. Then let φ' be the formula obtained from φ by replacing every occurrence of x_i by $\neg x_i$. To keep the number of agents constant, we use the same strategy as in the reduction in the proof of Proposition 13, where variables x_i, \dots, x_m are represented by strings of worlds with alternating relations R_b and R_a . We use the same polynomial-time computable mapping for variables x_1, \dots, x_m as in the proof of Proposition 13, which we repeat here for clarity. For $1 \leq i \leq m$, we define $[x_i]$ inductively as follows: $[x_1] = \hat{B}_b$,

$$[x_{j+1}] = \begin{cases} [x_j]\hat{B}_b & \text{if } [x_j] \text{ ends with } \hat{B}_a, \\ [x_j]\hat{B}_a & \text{if } [x_j] \text{ ends with } \hat{B}_b. \end{cases}$$

Then $[\varphi]$ is the adaptation of formula φ , where for $2 \leq i \leq m$, every occurrence of x_i in φ is replaced by $\hat{B}_a([x_i]y \wedge \neg[x_{i-1}]y)$ and every occurrence of x_1 is replaced by $\hat{B}_a[x_1]$. We say that a group of bottom worlds represents an assignment α to variables x_1, \dots, x_m if (1) for all x_i with $2 \leq i \leq m$ that are set to true by α , in all bottom worlds in the group $\hat{B}_a([x_i]y \wedge \neg[x_{i-1}]y)$ is true, and (2) if α sets x_1 to true, then in all bottom worlds in the group, $\hat{B}_a[x_1]y$ is true.

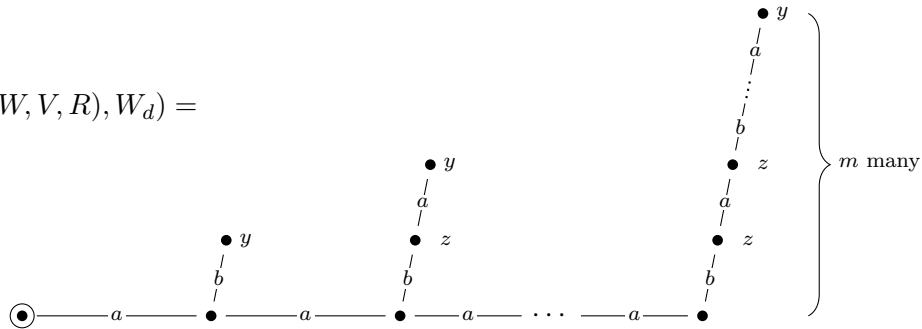
Just like in the proof of Proposition 13, the size of the formula (and consequently the modal depth of the formula) is kept constant by encoding the satisfiability of the formula with a single proposition. We do this in a similar way as in the proof of Proposition 8, by adding an extra action. Then each 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.

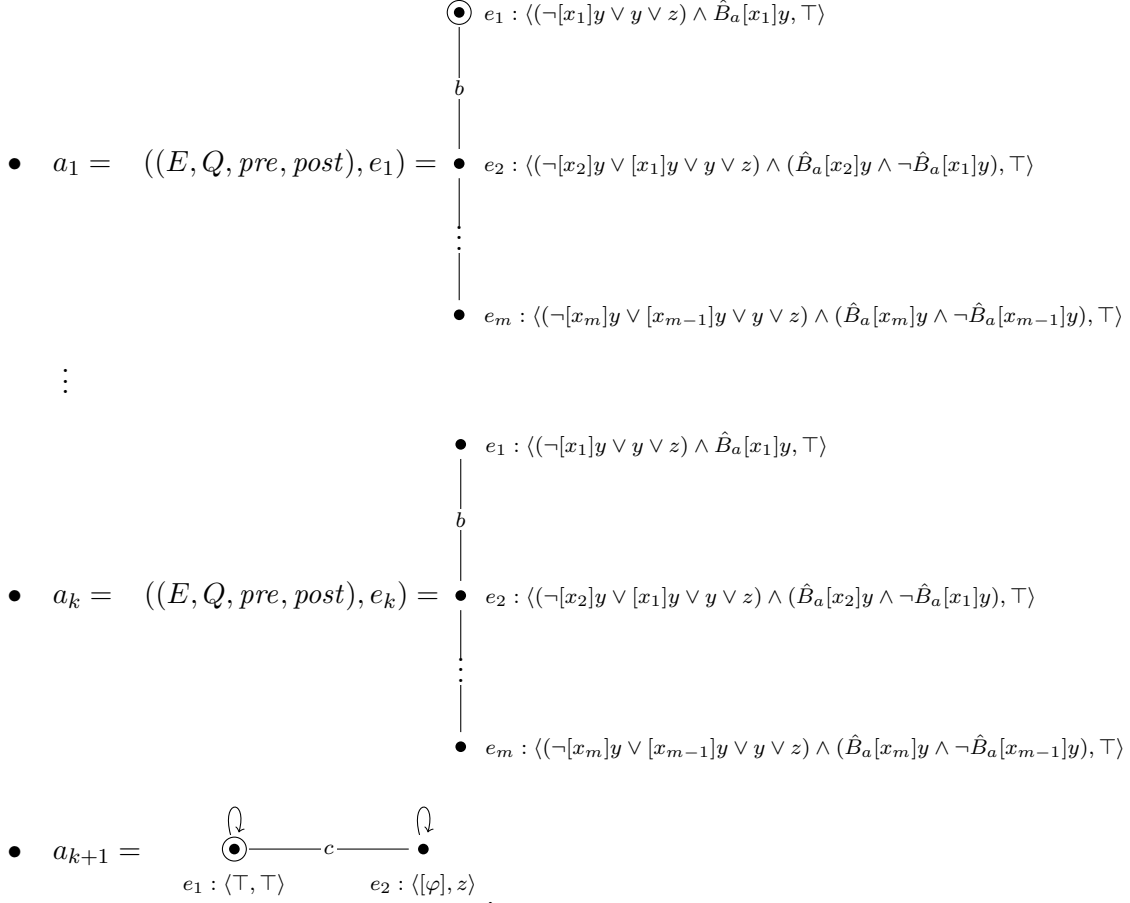
Now, we let $R(\varphi) = (P, s_0, a_1, \dots, a_k, a_{k+1}, \hat{B}_b\hat{B}_c z)$, where:

- $P = \{y, z, z^*\};$

- $\mathcal{A} = \{a, b, c\};$

- $s_0 = ((W, V, R), W_d) =$





Again, to ensure that the actions are applicable, in each action a_j for $1 \leq j \leq k$, the designated event is e_j . Here, the preconditions work by the same principle as the preconditions in Proposition 15, only now they look rather intricate, because the variables are represented by strings of worlds, like in Proposition 13. For each i , let $s_i = s_{i-1} \otimes a_i$. For each event e_j in some action a_i , the first conjunct makes sure that the bottom worlds in model s_{i-1} that represent variable x_j are not copied into model s_i . The second conjunct ensures that e_j only copies groups of worlds that have a bottom world that represents x_j .

We show that $(\varphi, k) \in \{k\}\text{-WSAT}[2\text{CNF}]$ if and only if $R(\varphi, k) \in \{a, f, o, p, u\}\text{-DBU}$. This follows by an argument that is similar to an argument in the proof of Proposition 14, which for the sake of clarity we repeat.

Assume that $(\varphi, k) \in \{k\}\text{-WSAT}[2\text{CNF}]$. Then there is some assignment α that sets $m - k$ variables to true and that satisfies φ' . By construction, there is some group of worlds in the final updated model that represents α , and in all the bottom worlds of that group $[\varphi]$ is true. In particular, one of the bottom worlds in this group will be R_b -accessible from the designated world and it will make $[\varphi]$ true. Action a_{k+1} makes sure that this world has an R_c relation to a world where proposition z^* is true. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_{k+1} \models \hat{B}_b \hat{B}_c z$.

Assume that $(\varphi, k) \notin \{k\}\text{-WSAT}[2\text{CNF}]$. Then there is no assignment α that sets $m - k$ variables to true and that satisfies φ' . By construction, for all the groups of worlds in the final updated model, in none of the bottom worlds of that group $[\varphi]$ is true. Then the precondition of the second event (e_2) of action a_{k+1} is not satisfied and therefore proposition z will not be true in any world in the model. Hence, $s_o \otimes a_1 \otimes \dots \otimes a_{k+1} \not\models \hat{B}_b \hat{B}_c z$.

Since this reduction runs in polynomial time, parameters a, f, o and p have constant values (namely $a = 3, f = 2, o = 1$, and $p = 2$), and parameter u depends only on parameter k (namely $u = k + 1$), we can conclude that $\{a, f, o, p, u\}$ -DBU is $\text{W}[1]$ -hard. \square

4.3.2 Tractability Results

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

Theorem 16. $\{e, u\}$ -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 (the event models) and u is the number of updates, i.e., the number of actions. Firstly, the algorithm computes a final updated model s_f by updating the initial state with the sequence of actions. Then it checks whether φ is true in s_f . To present this fpt-algorithm, we use the following claim:

Claim 1. Given an epistemic model M and a modal logic formula φ , deciding whether $M \models \varphi$ can be done in time polynomial in the size of M plus the size of φ .

First, we note that for every modal logic formula φ , we can construct in polynomial time a first-order logic formula ψ , with two variables, such that checking whether φ is true in a given epistemic model M can be done by checking the truth of ψ in this model. We can construct this ψ by means of the standard translation (van Benthem, 1977, Definition 2.1), whose definition can straightforwardly be adapted to the case of multiple agents. This adapted definition can also be used for the slightly more general case of multi-pointed models.

Furthermore, given a model and a formula in first-order logic with a constant number of variables, checking whether the formula is true in the model can be done in polynomial time in the size of the model plus the size of the formula (Vardi, 1995, Proposition 3.1). Therefore we can decide the truth of a given modal logic formula in a given model in polynomial time.

Now, we continue with our description of the fpt-algorithm. 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 by Claim 1, for each update, checking the preconditions can be done in time polynomial in $e^u \cdot |x|$. This means that computing final state s_f can be done in fpt-time.¹

¹We point out that for any computable function f and any polynomials p, q , we can find another computable

Then, the algorithm decides whether φ is true in s_f . By Claim 1, 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 $O(|s_0| \cdot e^u) + |\varphi|$, and 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

4.4 Overview of the Complexity 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 4.9, 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.

function f' and another polynomial p' , such that for all $n, k \in \mathbb{N}$ it holds that $q(f(k) \cdot p(n)) \leq f'(k) \cdot p'(n)$. Intuitively, this expresses that a polynomial composed with an ‘fpt-function’, is still an ‘fpt-function’.

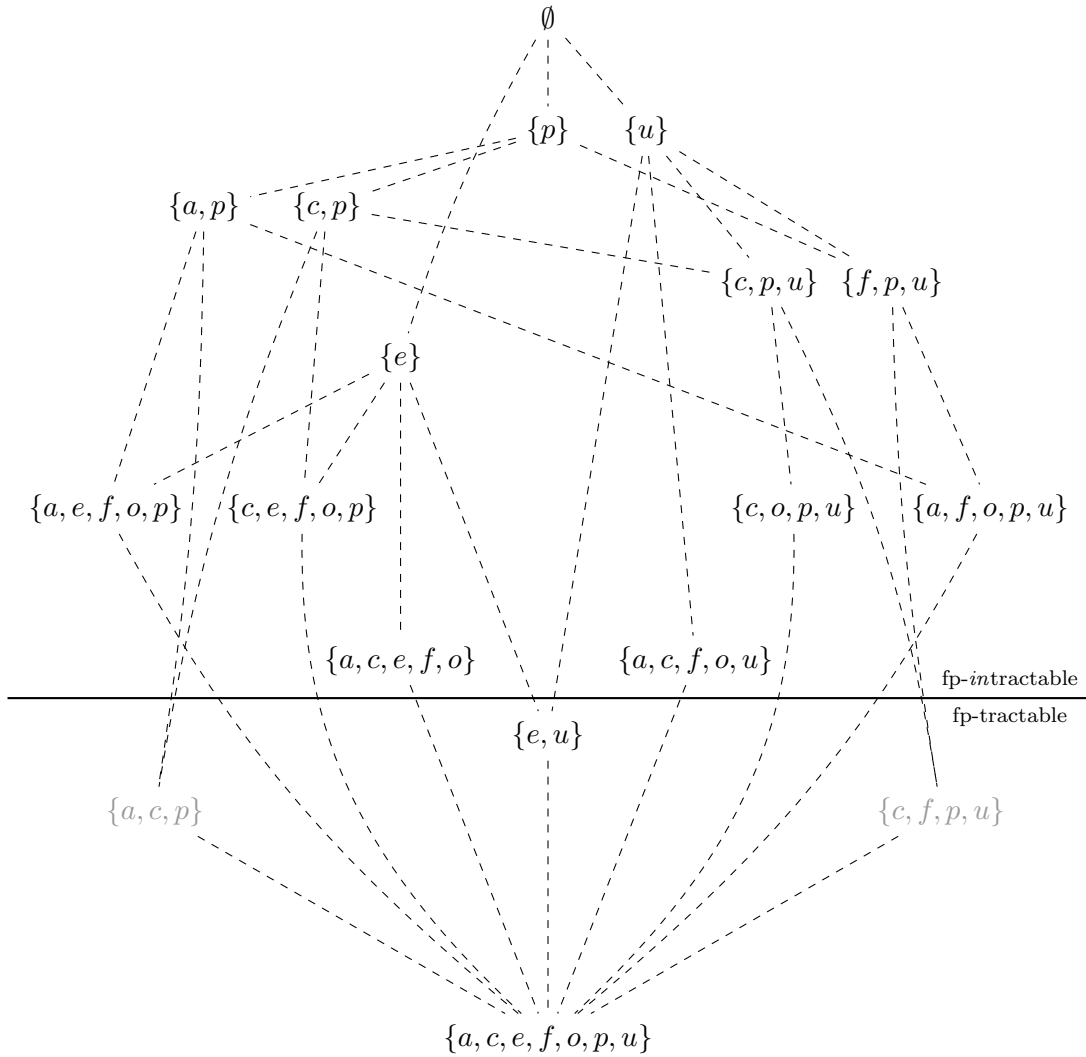


Figure 4.9: Overview of the parameterized complexity results for the different parameterizations of DBU, and the line between fp-tractability and fp-intractability (under the assumption that the cases for $\{a, c, p\}$ and $\{c, f, p, u\}$ are fp-tractable).

Chapter 5

Discussion

We presented the DYNAMIC BELIEF UPDATE model and analyzed its complexity. Here, we will discuss how our results contribute to both the field of cognitive science and to the area of logic. We will also discuss the interpretation of the complexity results and some open theoretical questions concerning the model. The aim of our model was to provide a formal framework in which the meaning and validity of various complexity claims in cognitive science and philosophy literature concerning ToM can be adequately interpreted and evaluated. In this way, the thesis hopes to contribute to disentangling convoluted debates in cognitive science and philosophy regarding the complexity of ToM. Furthermore, we hope that this thesis contributes to more collaboration between two (relatively) disconnected research communities, namely the DEL community and the computational cognitive science community.

In Section 3.3 we showed that DBU can be used to model several ToM tasks, and we illustrated how it captures an essential part of ToM, namely the attribution of beliefs and preferences to some agent on the basis of the observation of actions of this agent in an initial situation. In Section 4.2, we proved that DBU is PSPACE-hard. Since our model is situated at Marr’s (1982) computational-level, our results hold without any assumptions on the particular algorithms used to solve it. This result thus means that (without additional constraints), there is no algorithm that computes DBU in a reasonable (i.e., cognitively plausible) 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.

Interpretation of the complexity results The fact that people have difficulty understanding higher-order theory of mind 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 fixed-parameter tractable for the smaller parameter set $\{e, u\}$, this does not indicate that the order parameter is a source of complexity. So our complexity results do not explain why people have more trouble with higher-order ToM than they do with first-order theory of mind.

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 – our model 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 updating the initial state with the actions). 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. Our result that $\{e, u\}$ -DBU is fixed-parameter tractable indicates that the size of the final updated model can be a source of intractability.

The question arises how we can interpret parameters e and u in terms of their cognitive counterparts. With 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 (from the perspective of the *FPT-Cognition thesis*).

The number of event models (i.e., actions) that are used to update the initial state s_0 can be seen as corresponding to how complicated the situation is that is being considered, in terms of how much new information is taken into account to update existing beliefs about the situation. It could be the case that when we update our beliefs on the basis of new information (like actions by some agent), we usually do not take into account too much of this information at the same time. This is an empirical hypothesis that could potentially be tested experimentally.

The maximum number of events in an event model is somewhat tricky to interpret. One could see it as the level of uncertainty of the event that is being considered. However, in the case of single-pointed actions, this is not the uncertainty of the modeler (in our case the observer), and also not necessarily the uncertainty of the agents that are modeled. Even though a large amount of events can cause an event model to be rather intricate, in the single-pointed case, the model expresses that the modeler has perfect knowledge of the situation (since they designated an actual world) and the agents being modeled could also have high certainty (or even perfect knowledge) about the situation (this is not necessarily the case, but it is possible). In the case of single-pointed models, the number of events in an event model can be seen as the general level of uncertainty in the model, in the sense that many different possibilities, concerning what could have happened, are being taken into account.

In the case of multi-pointed (perspectival) actions (for some agent), we think that the number of events in an action might indeed represent the uncertainty of the agent (which could either be the modeler or the target being modeled). When there are many events in the perspectival action

for some agent, this means that the agent considers many different possibilities of what the action really entails. It might be good to stress that an event model does not only model the observation of something that happened, but (potentially) also the possible consequences of that what happened. Many events in an event model can indicate either that an agent is uncertain about what really happened in terms of what they observed, or that they are uncertain about what the consequences are of that what happened.

We would like to emphasize that it is not straightforward 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 worlds in daily life might adequately apply to some degree but could also be misleading. A more structural way of interpreting these parameters is called for. We think this is a very interesting topic for future research.

If our interpretation of the parameters is correct, then this means that if in a certain situation we (1) only consider for a limited number of happenings (at a time) how they influence the situation and the beliefs of the people involved, and (2) if the level of uncertainty about what these happenings precisely involve and what their consequences are is limited, then, according to our model, updating our beliefs on the basis of these happenings is computationally tractable. In the formalizations of the tasks that we modeled we indeed used a limited amount of actions with a limited amount of events in each action (we used a maximum of six). 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 remains a question for experimental research.

Futhermore, a restriction on the number of agents that are being considered also does not render the model tractable. Already for the case of just one agent, DBU is NP-hard. This means that $\{a\}$ -DBU is not fixed-parameter tractable. The same holds for the size of the formula and for the number of propositions that are being considered. So already in the case of attributing just a simple belief statement, where only a limited number of propositions (facts) are taken into account, the model is intractable (when there are no limitations on other factors, like the number of actions and events in the actions).

Lastly, the same holds for the maximum size of the preconditions, parameter c . It is difficult to give an intuitive interpretation of the size of the preconditions. The consideration of this parameter originates from technical reasons. We noticed that in some of our proofs, the size of the preconditions plays a crucial role. Therefore, we wondered whether this parameter (in combination with other parameters) is a source of complexity, i.e., whether there is some minimal parameter set, including c , for which DBU is fixed-parameter tractable. We expect this to be the case, but so far we have not been able to prove this. This is an interesting topic for future work.

Open theoretical questions Next, we turn to the open theoretical questions that remain. If people indeed exploit parameters e and u when they update their beliefs in dynamic situations, then we have shown how at least an essential part of ToM can be performed tractably. However,

our model says nothing about the other parts of ToM, like the use of attributed beliefs to predict behavior. These aspects of ToM remain to be modeled in a tractable way. It would be interesting to see how we could extend our model to incorporate also other parts of ToM, and to see how this influences the (parameterized) complexity of the model.

With our fpt result for $\{e, u\}$ -DBU, we showed that parameters e and u can be sources of the complexity of DBU. Since the values of e and u together are responsible for the exponential blow-up of the final updated model (in terms of the size of the initial state), it seems that an important factor for the intractability of DBU is the exponential blow-up of the initial state after updating it with the actions. The question arises whether this blow-up is an artifact of DEL, and in particular of the definition of the product update. It would be interesting to investigate whether there are other updating mechanisms possible for DEL that do not result in such a blow-up, that are still able to capture the change of epistemic models in dynamic situations¹.

The more general question is whether our model is a minimal model needed to capture ToM. In other words, can we claim that our model has the minimal complexity of all models that capture ToM, or could there be a different model with lower complexity that can also capture the essential parts of ToM. Another way of putting it is: is our model too general? In any case, we showed that our complexity results do not depend on the use of postconditions, since also without postconditions the model is PSPACE-hard. Furthermore, our results do not depend on our choice for KD45 models, they also hold for S5 models (and for arbitrary relations). An interesting question is whether there exist models with other properties than KD45 and S5 that are (to some extent) cognitively plausible, for which our results do not hold.

Lastly, there is the issue that our model assumes that the relevant aspects of a situation are given in the input. This means that a large part of the solution is given for free, i.e., selecting the relevant aspects of a situation is a computational problem in its own, that is not accounted for in our model. This relates to the frame problem or problem of relevance that we (briefly) discussed in Section 2.4. It is a challenge for the field of computational cognitive science at large to come up with formal methods to capture the complexity of this ‘front door’ part of computational problems, which is now given for free. Even though the frame problem does indeed occur, our results are relevant because they apply with respect to fixed frames, and our intractability results would carry over also to variable frames.

Besides the role that our results play in the investigation of (the complexity) of ToM our complexity results are also of interest in and of themselves. With our proof of Theorem 4 we solved an open question regarding the complexity of DEL. Aucher & Schwarzentruber (2013) proved PSPACE-hardness of DEL model checking for models with arbitrary relations, but the proof uses models that are not reflexive and does therefore not hold for S5 models. They ask whether model checking for DEL is PSPACE-hard also for S5 models. We proved that DBU is PSPACE-hard, for

¹Note that if such a modified update rule does not lead to a blow-up of the initial model, then it does not map the same function. If it would, then DBU would be tractable (i.e., polynomial-time computable), and thus $P = NP$.

models with arbitrary relations, but in particular for S5 models, since all the models that we use in our proof are S5 models. Furthermore, our hardness result also holds for the problem of model checking for DEL, since DBU is a special case of this problem. Moreover, because our proof does not use postconditions, model checking for DEL is even PSPACE-hard when the update models have no postconditions. This means that model checking for DEL is indeed PSPACE-complete for 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.

Chapter 6

Conclusion

Theory of Mind (ToM) is an important cognitive capacity, that is by many held to be ubiquitous in social interaction. However, at the same time, ToM seems to involve solving problems that are intractable and thus cannot be performed by humans in a (cognitively) plausible amount of time. Several cognitive scientists and philosophers have made claims about the intractability of ToM, and they argue that their particular theories of social cognition circumvent this problem of intractability. We argued that it is not clear how these claims regarding the intractability of ToM can be interpreted and/or evaluated and we argued that a formal framework is needed to make such claims more precise. In this thesis we proposed such a framework by means of a DEL-based model of ToM.

We introduced the *FPT-Cognition thesis* and explained how it formalizes the notion of tractability in the context of computational-level theories of cognition, making it possible to derive general results and abstract away from machine details. To be able to analyze the complexity of ToM, we proposed a computational-level theory based on dynamic epistemic logic, the DYNAMIC BELIEF UPDATE model (DBU). We showed that DBU can be used to model several ToM tasks and we proposed that it captures an essential part of ToM, namely the attribution of beliefs (or preferences) to some agent on the basis of the observation of actions performed by this agent (or other factors of change) in an initial situation.

We analyzed the (parameterized) complexity of the model; we showed that without any additional restrictions the model is intractable (PSPACE-complete). So in its general form (without any restrictions on its input domain) DBU cannot be a plausible theory of people's ability to engage in ToM. We did not find an effect of the order parameter (the level of belief attribution) on the (parameterized) complexity of the model. However, we found that DBU is fixed-parameter tractable for the parameter set $\{e, u\}$, which means that for simple situations, with a small amount of actions that contain a limited amount of events, according to our model, the attribution of beliefs in changing situations is tractable. Whether people indeed exploit these parameters in the way they perform ToM is an interesting hypothesis for experimental research.

Finally, our complexity results contribute to existing research on the complexity of DEL. We proved that DEL model checking is PSPACE-complete also for S5 models, which was an open problem in the literature.

Bibliography

- Anderson, J. R. (1978). Arguments concerning representations for mental imagery. *Psychological Review*, 85(4), 249.
- Anderson, J. R. (1990). *The adaptive character of thought*. Psychology Press.
- Anderson, J. R. (1993). *Rules of the mind*. Lawrence Erlbaum Associates.
- Apperly, I. (2011). *Mindreaders: the cognitive basis of “theory of mind”*. Psychology Press.
- Arora, S. & Barak, B. (2009). *Computational complexity: a modern approach*. Cambridge University Press.
- Arslan, B., Taatgen, N., & Verbrugge, R. (2013). Modeling developmental transitions in reasoning about false beliefs of others. In *Proceedings of the 12th International Conference on Cognitive Modeling, Ottawa: Carleton University*, (pp. 77–82).
- Arslan, B., Wierda, S., Taatgen, N., & Verbrugge, R. (2015). The role of simple and complex working memory strategies in the development of first-order false belief reasoning: A computational model of transfer of skills. In *Proceedings of the 13th International Conference on Cognitive Modeling*. In press.
- Aucher, G. (2010). An internal version of epistemic logic. *Studia Logica*, 94(1), 1–22.
- Aucher, G. & Schwarzentruher, F. (2013). On the complexity of dynamic epistemic logic. In *Proceedings of the Fourteenth conference on Theoretical Aspects of Rationality and Knowledge (TARK)*.
- Baker, C. L., Saxe, R. R., & Tenenbaum, J. B. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, (pp. 2469–2474).
- Baltag, A., Moss, L. S., & Solecki, S. (1998). 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)*, (pp. 43–56). Morgan Kaufmann Publishers Inc.

- Baltag, A. & Smets, S. (2006). Dynamic belief revision over multi-agent plausibility models. In *Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision Theory (LOFT 2006)*, volume 6, (pp. 11–24).
- Baron-Cohen, S., Leslie, A. M., & Frith, U. (1985). Does the autistic child have a “theory of mind”? *Cognition*, *21*(1), 37–46.
- Belkaid, M. & Sabouret, N. (2014). A logical model of theory of mind for virtual agents in the context of job interview simulation. *CoRR*, *abs/1402.5043*.
- Bennett, J. (1978). Some remarks about concepts. *Behavioral and Brain Sciences*, *1*(04), 557–560.
- van Benthem, J. (1977). *Modal Correspondence Theory*. PhD thesis, Universiteit van Amsterdam.
- van Benthem, J. (1989). Semantic parallels in natural language and computation. *Studies in Logic and the Foundations of Mathematics*, *129*, 331–375.
- van Benthem, J. (2008). Logic and reasoning: do the facts matter? *Studia Logica*, *88*(1), 67–84.
- van Benthem, J. (2011). *Logical dynamics of information and interaction*. Cambridge University Press.
- van Benthem, J., Hodges, H., & Hodges, W. (2007). Introduction. *Topoi*, *26*(1–2). Special issue on Logic and Psychology.
- Bergwerff, G., Meijering, B., Szymanik, J., Verbrugge, R., & Wierda, S. M. (2014). Computational and algorithmic models of strategies in turn-based games. In Bello, P., M. McShane, Guarini, M., & Scassellati, B. (Eds.), *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, (pp. 1778–1783).
- Blokpoel, M. (2015). *Understanding understanding: A computational-level perspective*. PhD thesis, Radboud University, Nijmegen, the Netherlands.
- Blokpoel, M., Kwisthout, J., van der Weide, T., & van Rooij, I. (2010). How action understanding can be rational, bayesian and tractable. In Ohlsson, S. & Catrambone, R. (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, (pp. 1643–1648). Cognitive Science Society.
- Blokpoel, M., van Kesteren, M., Stolk, A., Haselager, P., Toni, I., & van Rooij, I. (2012). Recipient design in human communication: simple heuristics or perspective taking? *Frontiers in human neuroscience*, *6*.
- Blokpoel, M., Wareham, T., Haselager, P., Toni, I., & van Rooij, I. (2015). Understanding by analogy: A computational-level perspective. Manuscript under review.

- Bolander, T. (2014). Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In *European Conference on Social Intelligence (ECSI 2014)*, (pp. 87–107).
- Bolander, T. & Andersen, M. B. (2011). Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1), 9–34.
- Bräuner, T. (2013). Hybrid-logical reasoning in false-belief tasks. In Schipper, B. (Ed.), *Proceedings of the Fourteenth conference on Theoretical Aspects of Rationality and Knowledge (TARK)*.
- Camerer, C. (2010). *Behavioral game theory*. New Age International.
- Cherniak, C. (1990). *Minimal rationality*. MIT Press.
- Church, A. (1936a). A note on the entscheidungsproblem. *The journal of symbolic logic*, 1(1), 40–41.
- Church, A. (1936b). An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2), 345–363.
- Churchland, P. M. (1981). Eliminative materialism and the propositional attitudes. *The Journal of Philosophy*, 78(2), 67–90.
- Cook, S. A. (1971). The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing (STOC 1971)*, (pp. 151–158)., New York.
- Copeland, B. J. (2008). The Church-Turing thesis. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2008 ed.).
- Cummins, R. (2000). “How does it work?” versus “what are the laws?": Two conceptions of psychological explanation. In F. C. Keil & R. A. Wilson (Eds.), *Explanation and Cognition* (pp. 117–144). MIT Press Cambridge, MA.
- Dégremont, C., Kurzen, L., & Szymanik, J. (2014). Exploring the tractability border in epistemic tasks. *Synthese*, 191(3), 371–408.
- Dennett, D. C. (1978). Beliefs about beliefs. *Behavioral and Brain Sciences*, 1(4), 568–570.
- Dennett, D. C. (1984). Cognitive wheels: The frame problem of AI. In C. Hookway (Ed.), *Minds, Machines and Evolution*. Cambridge University Press.
- Dennett, D. C. (1987). *The intentional stance*. MIT press.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.

- van Ditmarsch, H. & Kooi, B. (2006). Semantic results for ontic and epistemic change. In *Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision Theory (LOFT 2006)*, (pp. 87–117). Amsterdam University Press.
- van Ditmarsch, H. & Labuschagne, W. (2007). My beliefs about your beliefs: a case study in theory of mind and epistemic logic. *Synthese*, 155(2), 191–209.
- van Ditmarsch, H., van der Hoek, W., & Kooi, B. P. (2008). *Dynamic epistemic logic*. Springer.
- Downey, R. G. & Fellows, M. R. (1995). Fixed-parameter tractability and completeness. II. On completeness for $W[1]$. *Theoretical Computer Science*, 141(1–2), 109–131.
- Downey, R. G. & Fellows, M. R. (1999). *Parameterized Complexity*. Monographs in Computer Science. New York: Springer Verlag.
- Downey, R. G. & Fellows, M. R. (2013). *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer Verlag.
- Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3), 449–467.
- Eiter, T. & Gottlob, G. (1995). The complexity of logic-based abduction. *Journal of the ACM*, 42(1), 3–42.
- Fellows, M. R., Hermelin, D., Rosamond, F. A., & Vialette, S. (2009). On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1), 53–61.
- Flax, L. (2006). Logical modelling of leslie’s theory of mind. In *Proceedings of the 5th IEEE International Conference on Cognitive Informatics (ICCI 2006)*, volume 1, (pp. 25–30).
- Flobbe, L., Verbrugge, R., Hendriks, P., & Krämer, I. (2008). Children’s application of theory of mind in reasoning and language. *Journal of Logic, Language and Information*, 17(4), 417–442.
- Flum, J. & Grohe, M. (2003). Describing parameterized complexity classes. *Information and Computation*, 187(2), 291–319.
- Flum, J. & Grohe, M. (2006). *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Berlin: Springer Verlag.
- Fodor, J. A. (1983). *The modularity of mind: An essay on faculty psychology*. MIT press.
- Fodor, J. A. (1987). *Psychosemantics: The problem of meaning in the philosophy of mind*. MIT Press.
- Fortnow, L. (2009). The status of the P versus NP problem. *Communications of the ACM*, 52(9), 78–86.

- Frith, U. (2001). Mind blindness and the brain in autism. *Neuron*, 32(6), 969–979.
- Frixione, M. (2001). Tractable competence. *Minds and Machines*, 11(3), 379–397.
- Garey, M. R. & Johnson, D. R. (1979). *Computers and Intractability*. San Francisco: WH Freeman.
- Gasarch, W. I. (2012). Guest column: the second P =? NP poll. *SIGACT News*, 43(2), 53–77.
- Gerbrandy, J. & Groeneveld, W. (1997). Reasoning about information change. *Journal of logic, language and information*, 6(2), 147–169.
- Gierasimczuk, N. & Szymanik, J. (2011). A note on a generalization of the muddy children puzzle. In Apt, K. R. (Ed.), *TARK*, (pp. 257–264). ACM.
- Goldman, A. I. (1989). Interpretation psychologized. *Mind & Language*, 4(3), 161–185.
- Goldman, A. I. (2006). *Simulating minds: The philosophy, psychology, and neuroscience of mindreading*. Oxford University Press.
- Gopnik, A. (1997). *Words, thoughts, and theories*. MIT Press.
- Gopnik, A., Meltzoff, A. N., & Kuhl, P. K. (1999). *The scientist in the crib: Minds, brains, and how children learn*. William Morrow & Co.
- Gopnik, A. & Wellman, H. M. (1994). The theory theory. In L. Hirschfeld & S. Gelman (Eds.), *Mapping the mind: Domain specificity in cognition and culture* (pp. 257–293). Cambridge University Press.
- Gordon, R. M. (1986). Folk psychology as simulation. *Mind & Language*, 1(2), 158–171.
- Haselager, W. F. G. (1997). *Cognitive Science and Folk Psychology: The Right Frame of Mind*. Sage Publications.
- Hedden, T. & Zhang, J. (2002). What do you think I think you think?: Strategic reasoning in matrix games. *Cognition*, 85(1), 1–36.
- Heider, F. (1958). *The psychology of interpersonal relations*. New York: Wiley.
- Hiatt, L. M. & Trafton, J. G. (2010). A cognitive model of theory of mind. In *Proceedings of the 10th International Conference on Cognitive Modeling*, (pp. 91–96).
- Isaac, A. M., Szymanik, J., & Verbrugge, R. (2014). Logic and complexity in cognitive science. In A. Baltag & S. Smets (Eds.), *Johan van Benthem on Logic and Information Dynamics*, volume 5 of *Outstanding Contributions to Logic* (pp. 787–824). Springer International Publishing.

- Kinderman, P., Dunbar, R., & Bentall, R. P. (1998). Theory-of-mind deficits and causal attributions. *British Journal of Psychology*, *89*(2), 191–204.
- Kleene, S. C. (1936). Lambda-definability and recursiveness. *Duke mathematical journal*, *2*(2), 340–353.
- Kleene, S. C. (1967). *Mathematical logic*. New York: Wiley.
- Kugel, P. (1986). Thinking may be more than computing. *Cognition*, *22*(2), 137–198.
- Kwisthout, J. (2012). Relevancy in problem solving: a computational framework. *The Journal of Problem Solving*, *5*(1), 4.
- van Lambalgen, M. & Counihan, M. (2008). Formal models for real people. *Journal of Logic, Language and Information*, *17*(4), 385–389.
- Leitgeb, H. (2008). Introduction to the special issue. *Studia Logica*, *88*(1), 1–2.
- Leslie, A. M. (2005). Developmental parallels in understanding minds and bodies. *Trends in cognitive sciences*, *9*(10), 459–462.
- Levesque, H. J. (1988). Logic and the complexity of reasoning. *Journal of Philosophical Logic*, *17*(4), 355–389.
- Levin, L. A. (1973). Universal sequential search problems. *Problems of Information Transmission*, *9*(3), 265–266.
- Levinson, S. C. (2006). On the human ‘interaction engine’. In N. J. Enfield & S. C. Levinson (Eds.), *Roots of human sociality: Culture, cognition and interaction* (pp. 39–69). Oxford: Berg.
- Lorini, E. & Schwarzenrüber, F. (2011). A logic for reasoning about counterfactual emotions. *Artificial Intelligence*, *175*(3-4), 814–847.
- Lucas, J. R. (1961). Minds, machines and gödel. *Philosophy*, *36*(137), 112–127.
- Lyons, M., Caldwell, T., & Shultz, S. (2010). Mind-reading and manipulation – is machiavellianism related to theory of mind? *Journal of Evolutionary Psychology*, *8*(3), 261–274.
- Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: WH Freeman.
- Meijering, B., van Rijn, H., Taatgen, N. A., & Verbrugge, R. (2012). What eye movements can tell about theory of mind in a strategic game. *PLoS ONE*, *7*(9), e45961.
- Miller, S. A. (2009). Children’s understanding of second-order mental states. *Psychological bulletin*, *135*(5), 749–773.

- Mostowski, M. & Wojtyniak, D. (2004). Computational complexity of the semantics of some natural language constructions. *Annals of Pure and Applied Logic*, 127(1–3), 219–227.
- Nichols, S. & Stich, S. P. (2003). *Mindreading: An integrated account of pretence, self-awareness, and understanding other minds*. Clarendon Press/Oxford University Press.
- Niedermeier, R. (2006). *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford: Oxford University Press.
- O’Grady, C., Kliesch, C., Smith, K., & Scott-Phillips, T. C. (2015). The ease and extent of recursive mindreading, across implicit and explicit tasks. *Evolution and Human Behavior*. In press.
- Onishi, K. H. & Baillargeon, R. (2005). Do 15-month-old infants understand false beliefs? *Science*, 308(5719), 255–258.
- Penrose, R. (1989). *The Emperor’s New Mind: Concerning Computers, Minds, And The Laws Of Physics*. Oxford University Press.
- Penrose, R. (1994). *Shadows of the Mind*, volume 52. Oxford University Press.
- Pitt, D. (2013). Mental representation. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2013 ed.).
- Plaza, J. (2007). Logics of public communications. *Synthese*, 158(2), 165–179.
- Plaza, J. A. (1989). Logics of public communications. In Emrich, M. L., Pfeifer, M. S., Hadzikadic, M., & Ras, Z. (Eds.), *Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems: poster session program*, (pp. 201–216). Oak Ridge National Laboratory. Reprinted as (Plaza, 2007).
- Premack, D. & Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(04), 515–526.
- Putnam, H. (1967). Psychological predicates. In W. H. Capitan & D. D. Merrill (Eds.), *Art, mind, and religion* (pp. 37–48). University of Pittsburgh Press, Pittsburgh.
- Pylyshyn, Z. W. (1978). When is attribution of beliefs justified? *Behavioral and brain sciences*, 1(04), 592–593.
- Pylyshyn, Z. W. (1987). *The robot’s dilemma: The frame problem in artificial intelligence*. Ablex Norwood, NJ.
- Ravenscroft, I. (2010). Folk psychology as a theory. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2010 ed.).

- Rietveld, E. (2012). Context-switching and responsiveness to real relevance. In J. Kiverstein & M. Wheeler (Eds.), *Heidegger and Cognitive Science* (pp. 105–135). Palgrave Macmillan.
- van Rooij, I. (2003). *Tractable cognition: Complexity theory in cognitive psychology*. PhD thesis, University of Victoria.
- van Rooij, I. (2008). The tractable cognition thesis. *Cognitive science*, *32*(6), 939–984.
- van Rooij, I., Kwisthout, J., Blokpoel, M., Szymanik, J., Wareham, T., & Toni, I. (2011). Intentional communication: Computationally easy or difficult? *Frontiers in Human Neuroscience*, *5*(52), 1–18.
- van Rooij, I. & Wareham, T. (2012). Intractability and approximation of optimization theories of cognition. *Journal of Mathematical Psychology*, *56*(4), 232–247.
- Slors, M. (2012). The model-model of the theory-theory. *Inquiry*, *55*(5), 521–542.
- Stenning, K. & Van Lambalgen, M. (2008). *Human reasoning and cognitive science*. MIT Press.
- Stiller, J. & Dunbar, R. I. (2007). Perspective-taking and memory capacity predict social network size. *Social Networks*, *29*(1), 93–104.
- Stockmeyer, L. J. & Meyer, A. R. (1973). Word problems requiring exponential time (preliminary report). In *Proceedings of the 5th Annual ACM Symposium on the Theory of Computing (STOC 1973)*, (pp. 1–9). ACM.
- Szymanik, J. (2013). Backward induction is PTIME-complete. In H. H. D. Grossi, O. Roy (Ed.), *Proceedings of the Fourth International Workshop on Logic, Rationality and Interaction, Lecture Notes in Computer Science*, volume 8196 (pp. 352–356). Heidelberg: Springer.
- Szymanik, J., Meijering, B., & Verbrugge, R. (2013). Using intrinsic complexity of turn-taking games to predict participants’ reaction times. In Knauff, M., Pauen, M., Sebanz, N., & Wachsmuth, I. (Eds.), *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, (pp. 1426–1432)., Austin, TX. Cognitive Science Society.
- Triona, L. M., Masnick, A. M., & Morris, B. J. (2002). What does it take to pass the false belief task? An ACT-R model. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*.
- Tsotsos, J. K. (1990). Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, *13*(03), 423–445.
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *J. of Math*, *58*(345-363), 5.

- van Emde Boas, P. (1990). Machine models and simulations. In J. van. Leeuwen (Ed.), *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity* (pp. 1–66). MIT Press.
- Vardi, M. Y. (1995). On the complexity of bounded-variable queries (extended abstract). In *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1995)*, (pp. 266–276)., New York, NY, USA. ACM.
- Verbrugge, R. (2009). Logic and social cognition: the facts matter, and so do computational models. *Journal of Philosophical Logic*, 38(6), 649–680.
- Wareham, H. T. (1998). *Systematic parameterized complexity analysis in computational phonology*. PhD thesis, University of Victoria.
- Wason, P. C. (1968). Reasoning about a rule. *The Quarterly Journal of Experimental Psychology*, 20(3), 273–281.
- Wellman, H. M., Cross, D., & Watson, J. (2001). Meta-analysis of theory-of-mind development: The truth about false belief. *Child Development*, 72(3), 655–684.
- Wheeler, M. (2008). Cognition in context: Phenomenology, situated robotics and the frame problem. *International Journal of Philosophical Studies*, 16(3), 323–349.
- Wimmer, H. & Perner, J. (1983). Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. *Cognition*, 13(1), 103–128.
- Zawidzki, T. W. (2013). *Mindshaping: A New framework for understanding human social cognition*. MIT Press.

Appendix

Turing Machines

The machine model that underlies our complexity-theoretic results is the standard multitape Turing machine model. Intuitively, a Turing machine is a computing device that can read and write symbols and has a (limited) internal memory. It has multiple tapes that are divided into cells, which each contain a symbol. Furthermore, on each tape there is a “head”, positioned over one of the cells, that can read off the symbol in that cell and can replace a given symbol with another symbol. After each “step in the computation” each head can move one cell to the right or left, or can remain at its position.

What a Turing machine can do is remember what state it currently is in and read which symbols are currently under its heads. Depending on this information it will make a certain transition: its (internal) state changes (or remains the same); on each tape it writes a certain symbol (overwriting the previous symbol) on the cell under the head (which can be same symbol as it is currently reading), and each head moves to the right, left, or remains in the same position. These transitions are defined by a “transition relation” and it is this transition relation that defines a Turing machine. Together with a given input string, the transition relation determines the behavior of a Turing machine.

A Turing machine starts with a given input string on its first tape and “blank symbols” on the rest of its cells. Then, according to its transition relation, it will start “computing”: its heads will read, write and move over the tape cells, and it will change its state accordingly, either forever or until it ends up in a halting state, which can be a rejecting state or an accepting state. To summarize, given a certain input string, a Turing machine can either accept or reject this string, or end up in an infinite loop.

We give a formal definition:

Definition 6.0.1 (Turing machine). *We use the same notation as Flum and Grohe (Flum & Grohe, 2006, Appendix A.1). A Turing machine is a tuple*

$$\mathbb{M} = (S, \Sigma, \Delta, s_0, F),$$

where:

- S is the finite set of states,
- Σ is the alphabet,
- $s_0 \in S$ is the initial state,
- $F \subseteq S$ is the set of accepting states,
- The symbols $\$, \square \notin \Sigma$ are special symbols. “\$” marks the left end of any tape. It cannot be overwritten and only allows **R**-transitions.¹ “ \square ” is the blank symbol.
- $\Delta \subseteq S \times (\Sigma \cup \{\$, \square\})^m \times S \times (\Sigma \cup \{\$\})^m \times \{\mathbf{L}, \mathbf{R}, \mathbf{S}\}^m$ is the transition relation. Here $m \in \mathbb{N}$ is the number of tapes. If for all $(s, \bar{a}) \in S \times (\Sigma \cup \{\$, \square\})^m$ there is at most one (s', \bar{a}', \bar{d}') such that $(s, \bar{a}, s', \bar{a}', \bar{d}') \in \Delta$, then the Turing machine \mathbb{M} is called deterministic; otherwise \mathbb{M} is nondeterministic. (The elements of Δ are the transitions.)

Intuitively, the tapes of our machine are bounded to the left and unbounded to the right. The leftmost cell, the 0-th cell, of each tape carries a “\$”, and initially, all other tape cells carry the blank symbol. The input is written on the first tape, starting with the first cell, the cell immediately to the right of the “\$”. A configuration is a tuple $C = (s, x_1, p_1, \dots, x_m, p_m)$, where $s \in S$, $x_i \in \Sigma^*$, and $0 \leq p_i \leq |x_i| + 1$ for each $1 \leq i \leq k$. Intuitively, $\$, x_i \square \square \dots$ is the sequence of symbols in the cells of tape i , and the head of tape i scans the p_i -th cell. The initial configuration for an input $x \in \Sigma^*$ is $C_0(x) = (s_0, x, 1, \epsilon, 1, \dots, \epsilon, 1)$, where ϵ denotes the empty word. A computation step of \mathbb{M} is a pair (C, C') of configurations such that the transformation from C to C' obeys the transition relation. We omit the formal details. We write $C \rightarrow C'$ to denote that (C, C') is a computation step of \mathbb{M} . If $C \rightarrow C'$, we call C' a successor configuration of C . A halting configuration is a configuration that has no successor configuration. A halting configuration is accepting if its state is in F .

A finite run of \mathbb{M} is a sequence (C_0, \dots, C_ℓ) where $C_{i-1} \rightarrow C_i$ for all $1 \leq i \leq \ell$, C_0 is an initial configuration, and C_ℓ is a halting configuration. An infinite run of \mathbb{M} is a sequence (C_0, C_1, C_2, \dots) where $C_{i-1} \rightarrow C_i$ for all $i \in \mathbb{N}$, C_0 is an initial configuration. If the first configuration C_0 of a run ρ is $C_0(x)$, then we call ρ a run with input x . A run is accepting if its last configuration is an accepting configuration. The length of a run is the number of steps it contains if it is finite, or ∞ if it is infinite.

The problem accepted by \mathbb{M} is the set $Q_{\mathbb{M}}$ of all $x \in \Sigma^*$ such that there is an accepting run of \mathbb{M} with initial configuration $C_0(x)$. If all runs of \mathbb{M} are finite, then we say that \mathbb{M} decides $Q_{\mathbb{M}}$, and we call $Q_{\mathbb{M}}$ the problem decided by \mathbb{M} .

¹To formally achieve that “\$” marks the left end of the tapes, whenever $(s, (a_1, \dots, a_m), s', (a'_1, \dots, a'_m), (d_1, \dots, d_m)) \in \Delta$, then for all $1 \leq i \leq m$ we have that $a_i = \$$ if and only if $a'_i = \$$ and that $a_i = \$$ implies $d_i = \mathbf{R}$.

Let $t : \mathbb{N} \rightarrow \mathbb{N}$ be a function. We say that a Turing machine \mathbb{M} runs in time t if for every $x \in \Sigma^*$ every run of \mathbb{M} with input x has length at most $t(|x|)$. A Turing machine \mathbb{M} runs in polynomial time if there exists a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ such that \mathbb{M} runs in time p .

Let $s : \mathbb{N} \rightarrow \mathbb{N}$ be a function. We say that a Turing machine \mathbb{M} runs in space s if for every $x \in \Sigma^*$ every run of \mathbb{M} with input x only consists of configurations of size at most $s(|x|)$.

Definition 6.0.2 (Oracle machine). Let C be a decision problem. A Turing machine T with access to a C oracle is a Turing machine with a dedicated oracle tape and dedicated states q_{oracle} , q_{yes} and q_{no} . Whenever T is in the state q_{oracle} , it does not proceed according to the transition relation, but instead it transitions into the state q_{yes} if the oracle tape contains a string x that is a yes-instance for the problem C , i.e., if $x \in C$, and it transitions into the state q_{no} if $x \notin C$.

Single and multi-pointed epistemic models

We give a simple proof that there exist multi-pointed epistemic models for which there exists no equivalent single-pointed epistemic model.

Proposition 17. *There exists a multi-pointed epistemic model (M, W_d) such that for all single-pointed epistemic models (M', w) there exists a formula $\varphi \in \mathcal{L}_B$ such that:*

$$M, W_d \models \varphi \quad \not\equiv \quad M', w \models \varphi.$$

Proof. We provide such a multi-pointed epistemic model (M, W_d) . Let p be an arbitrary proposition in P , and let a be an arbitrary agent.

$$(M, W_d) = \begin{array}{cc} a & a \\ \downarrow & \downarrow \\ \bullet & \bullet \\ p & \neg p \end{array}$$

It is straightforward to check that for the formulas $\varphi_1 = p$ and $\varphi_2 = \neg p$ it holds that $M, W_d \not\models \varphi_1$ and $M, W_d \not\models \varphi_2$. We show that for all single-pointed epistemic models (M', w) there exists some formula φ such that $M, W_d \models \varphi \not\equiv M', w \models \varphi$. Let (M', w) be an arbitrary single-pointed epistemic model. We distinguish two cases: either (1) $V(w, p) = 1$ or (2) $V(w, p) = 0$. In case (1), we know that $M', w \models \varphi_1$. However, we already established that $M, W_d \not\models \varphi_1$. Thus (M, W_d) and (M', w) are not equivalent. The other case is analogous. In case (2), we know that $M', w \models \varphi_2$. However, we established that $M, W_d \not\models \varphi_2$. Therefore (M, W_d) and (M', w) are not equivalent. Since the single-pointed model (M', w) was arbitrary, we can conclude that there is no single-pointed model that is equivalent to (M, W_d) . \square