The Complexity of Limited Belief Reasoning — The Quantifier-Free Case

Yijia Chen[1] and **Abdallah Saffidine**[2] and **Christoph Schwering**[3]
[1] Fudan University, Shanghai 201203, China
[2] Australian National University, Canberra ACT 2600, Australia
[3] University of New South Wales, Sydney NSW 2052, Australia
yijiachen@fudan.edu.cn, abdallah.saffidine@anu.edu.au, c.schwering@unsw.edu.au

Abstract

The classical view of epistemic logic is that an agent knows all the logical consequences of their knowledge base. This assumption of *logical omniscience* is often unrealistic and makes reasoning computationally intractable. One approach to avoid logical omniscience is to limit reasoning to a certain *belief level*, which intuitively measures the reasoning "depth."

This paper investigates the computational complexity of reasoning with belief levels. First we show that while reasoning remains tractable if the level is constant, the complexity jumps to PSPACE-complete—that is, beyond classical reasoning—when the belief level is part of the input. Then we further refine the picture using parameterized complexity theory to investigate how the belief level and the number of non-logical symbols affect the complexity.

1 Introduction

The standard way of modeling knowledge and belief[1] in epistemic logic is in terms of *possible worlds*: an agent knows a proposition if and only if it is true in all worlds the agent considers possible. A side-effect of this model is that agents are assumed to be *logically omniscient*, that is, they know all the consequences of what they know [Hintikka, 1975].

Unfortunately, the assumption of logical omniscience is inappropriate for most resource-bounded agents like humans or robots: it drives up the computational cost of reasoning and is usually far beyond the their capabilities. Theories of *limited belief* therefore aim to lift the omniscience assumption.

A number of theories of limited belief have been proposed, predominantly in the 1980s and 1990s [Konolige, 1986; Kaplan and Schubert, 2000; Vardi, 1986; Fagin and Halpern, 1987; Levesque, 1984; Patel-Schneider, 1990; Lakemeyer, 1994; Delgrande, 1995]. A common problem with these approaches is, however, that either their model of limiting belief is too fine-grained or it misses out on simple inferences.

A novel approach to limited belief developed over a series of papers [Liu *et al.*, 2004; Lakemeyer and Levesque, 2013; 2014; Klassen *et al.*, 2015; Schwering and Lakemeyer, 2016; Lakemeyer and Levesque, 2016; Schwering, 2017] attempts to address this issue. The basic idea is to stratify beliefs into *belief levels*, where the first one, level 0, only comprises the explicit beliefs, that is, what is written down expressly in the knowledge base, and higher belief levels $k+1$ draw additional conclusions based on what is believed at level k. Semantically the logic can be characterized using sets of clauses instead of possible worlds, and through *case splits*, that is, by branching on all the values some term can take and propagating the value.

As an example, consider the following knowledge base:

$$\mathrm{fatherOf}(\mathrm{Sally}) = \mathrm{Frank} \vee \mathrm{fatherOf}(\mathrm{Sally}) = \mathrm{Fred}$$
$$\mathrm{fatherOf}(\mathrm{Sally}) = n \supset \mathrm{rich}(n) = \top \ \text{ for } n \in \{\mathrm{Frank}, \mathrm{Fred}\}.$$

Here, Sally, Frank, Fred, $\top$ name distinct individuals ($\top$ is an auxiliary name for modeling propositions), whereas fatherOf and rich represent functions in the classical sense. From this knowledge we can deduce $\mathrm{rich}(\mathrm{Frank}) = \top \vee \mathrm{rich}(\mathrm{Fred}) = \top$ at level 1 by splitting on all potential fathers of Sally: if Frank is the father, then Frank is rich; if Fred is the father, then he is rich; every other potential father contradicts the first clause.

Logics of limited belief in general and the belief level mechanism in particular aim to provide means of controlling the reasoning effort in a comprehensible and explainable way, as contrasted with using a classical reasoner and terminating it after a timeout, for example. The rationale behind the belief level approach is that reasoning at small belief levels should be relatively cheap but still sufficient for the average problem a human or a robot faces during their daily operation. Experiments confirm this hypothesis for the confined domains of Sudoku and Minesweeper [Schwering, 2017].

Contribution

In this paper, we analyze reasoning with belief levels from the perspective of complexity theory. More precisely, we study the problem of deciding whether a knowledge base entails a query at a certain belief level.

For a constant belief level, the problem is indeed in PTIME and hence known to be tractable; the same holds when the knowledge base and query only mention a constant number of function terms.

However, we shall see that if both the belief level and the number of function terms are part of the problem input, then the complexity jumps to PSPACE-complete! This may come as a surprise given that classical, unlimited reasoning is in co-NP. So (large) belief levels appear to make reasoning harder.

[1] We use the terms knowledge and belief interchangeably.

$\|\mathcal{F}\|$	k	$\|\mathcal{N}\|$		
Input	Input	—	PSPACE-c	Theorem 9
Input	Param	Input	AW[P]-c	Theorem 11
Input	Param	Param Const	W[P]-c	Proposition 12
Param	Input Param	Input	co-W[P]-c	Theorem 13
Param	—	Param Const	FPT	Proposition 14
— Const	Const —	— —	PTIME	Corollary 8

Table 1: The classification of Limited Belief Reasoning depending on whether the belief level k, number of function terms $|\mathcal{F}|$, and the number of standard names $|\mathcal{N}|$ are input, parameters, or constant.

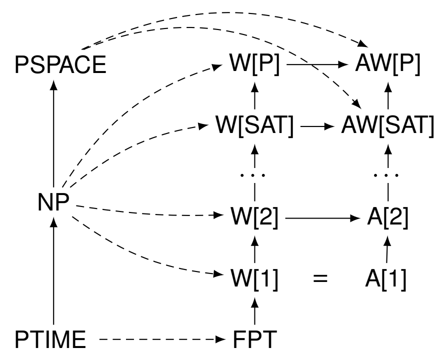

Figure 1: Overview of the classical and parameterized complexity classes relevant for this paper. The classes from the W-hierarchy include parameterized versions of different natural NP-complete problems; the A-hierarchy can be seen as parameterized version of the polynomial hierarchy. $C_1 \longrightarrow C_2$ means $C_1 \subseteq C_2$, and $C_1 \dashrightarrow C_2$ means that C_2 can be seen as a parameterized analogue of C_1.

Intuitively, the jump is caused by the belief level limiting a possibly scarce resource, namely the number of case splits, which needs to be utilized in an optimal way.

The gap between PTIME and PSPACE-completeness calls for a more refined analysis, which we carry out using parameterized complexity theory. We investigate three dimensions of parameters: (1) the belief level, (2) the number of function terms mentioned in the reasoning problem (in the above example, the function terms are fatherOf(Sally), rich(Frank), rich(Fred)), and (3) the number of mentioned so-called standard names (in the example, these names are Sally, Frank, Fred, $\top$). Parameterized complexity theory offers the W- and A-hierarchies to classify problems between PTIME and NP and between NP and PSPACE, respectively. We locate the parameterized variants of our problem within these hierarchies.

A comprehensive overview of the paper's findings is given in Table 1. Figure 1 illustrates the relationships between the complexity classes we deal with in this paper.

The paper is structured as follows. The next section introduces the logic of limited belief and defines the reasoning problem that we shall study. Section 3 introduces a gadget that we use in several reductions. Section 4 begins the complexity analysis from the perspective of classical complexity theory with PTIME and PSPACE results. Section 5 refines the picture using parameterized complexity theory. Then we conclude.

Full proofs of our results can be found in [Chen *et al.*, 2018].

2 The Logic of Limited Belief

In its most recent form, the logic of limited belief is a first-order logic with functions, equality, and epistemic modal operators [Lakemeyer and Levesque, 2016; Schwering, 2017]. In this paper, we limit our consideration to the *quantifier-free* case.

This section first introduces the syntax and semantics of this logic, and then defines the reasoning problem whose complexity we will study in the remainder of the paper: *if we know* KB *explicitly, do we believe α at level k?* The definitions and results of this section are adopted from [Schwering, 2017] with some minor simplifications to ease the technical treatment; these simplifications do not affect the expressivity or complexity of the reasoning task at hand.

2.1 The Language

A term is either a *standard name* (or *name* for short) or a *function term* $f(n_1, \ldots, n_j)$, where f is a function symbol and every n_i is a standard name. Standard names can be understood as special constants that satisfy the unique-names assumption and an infinitary version of domain closure. We assume an infinite supply of standard names as well as of function symbols.

A *literal* is an expression of the form $t = n$ or $\neg t = n$, where t is a function term and n is a standard name. A *formula* is a literal or an expression of the form $\neg\alpha$, $(\alpha \vee \beta)$, or $\mathbf{B}_k\,\alpha$, where α, β are formulas and $k \geq 0$ is a natural number. We read $\mathbf{B}_k\,\alpha$ as "α is known at belief level k;" in case $k = 0$ we also say "α is known explicitly." We use the usual abbreviations $t \neq n$, $(\alpha \wedge \beta)$, $(\alpha \supset \beta)$, and may omit brackets to ease readability.

A formula without $\mathbf{B}_k$ is called *objective*. Schwering [2017] has shown that there is a linear Turing reduction from the reasoning problem with nested beliefs to the non-nested case. Hence to simplify the presentation we henceforth assume that α in $\mathbf{B}_k\,\alpha$ is objective. As usual, a conjunction of disjunctions of literals is said to be in *conjunctive normal form* (CNF).

2.2 The Semantics

The semantics of limited belief is based on clause subsumption, unit propagation, and case splits. A *clause* is a set of literals. We abuse notation and identify a non-empty clause $\{\ell_1, \ldots, \ell_j\}$ with the formula $(\ell_1 \vee \ldots \vee \ell_j)$. In the rest of this paragraph, we implicitly assume that n, n' refer to distinct standard names. A clause c_1 *subsumes* another clause c_2 iff for every $t = n \in c_1$, either $t = n \in c_2$ or $t \neq n' \in c_2$, and for every $t \neq n \in c_1$, also $t \neq n \in c_2$. We say two literals ℓ_1, ℓ_2 are *complementary* when they are of the form $t = n$ and $t \neq n$ or of the form $t = n$ and $t = n'$. The *unit propagation* of a clause c and a literal ℓ is obtained by removing from c all literals complementary to ℓ. For a set of clauses s, we let $\mathsf{UP}(s)$ be the closure of s under unit propagation and subsumption.

The *truth relation* $\approx$ is defined between a formula α and a set of clauses s. Intuitively, s acts as a partial model. At belief level 0, α is broken down to clause level and then checked for subsumption by $\mathsf{UP}(s)$. Higher belief levels allow to branch on a function term t and all its values n and add $t = n$ to s, which may then trigger unit propagation in $\mathsf{UP}(s)$ and thus produce new inferences. The formal definition is as follows:

1. if c is a clause: $s \approx c$ iff $c \in \mathsf{UP}(s)$
2. if $(\alpha \vee \beta)$ is not a clause: $s \approx (\alpha \vee \beta)$ iff $s \approx \alpha$ or $s \approx \beta$
3. $s \approx \neg(\alpha \vee \beta)$ iff $s \approx \neg\alpha$ and $s \approx \neg\beta$
4. $s \approx \neg\neg\alpha$ iff $s \approx \alpha$
5. $s \approx \mathbf{B}_0\alpha$ iff $s \approx \alpha$
6. $s \approx \mathbf{B}_{k+1}\alpha$ iff for some function term t, for all names n, $s \cup \{t = n\} \approx \mathbf{B}_k\alpha$
7. $s \approx \neg\mathbf{B}_k\alpha$ iff $s \not\approx \mathbf{B}_k\alpha$

In the remainder, we refer to these definitions as Rules 1–7. As usual, a formula α is *valid*, written $\approx \alpha$, iff $s \approx \alpha$ for every set of clauses s.

The belief level k in $\mathbf{B}_k\alpha$ specifies the number of case splits, which corresponds to the maximum permitted reasoning effort for proving α. Limited belief is monotonic in the belief level:

Lemma 1 $\approx \mathbf{B}_k\alpha \supset \mathbf{B}_{k+1}\alpha$.

Moreover, belief stabilizes at a high-enough belief level in the following sense:

Lemma 2 *Let $\mathcal{F}$ contain all function terms in s and α, and let $k \geq |\mathcal{F}|$. Then $s \approx \mathbf{B}_k\alpha$ iff $s \approx \mathbf{B}_{|\mathcal{F}|}\alpha$.*

Example

Let us revisit the example from the introduction to illustrate how the semantics works. Let s contain the clauses

$$\text{fatherOf(Sally)} = \text{Frank} \vee \text{fatherOf(Sally)} = \text{Fred}$$
$$\text{fatherOf(Sally)} \neq \text{Frank} \vee \text{rich(Frank)} = \top$$
$$\text{fatherOf(Sally)} \neq \text{Fred} \vee \text{rich(Fred)} = \top$$

and let c denote the clause $\text{rich(Frank)} = \top \vee \text{rich(Fred)} = \top$. Then $s \approx \mathbf{B}_1 c$ holds by splitting the cases for Sally's father:

- $\mathsf{UP}(s \cup \{\text{fatherOf(Sally)} = \text{Frank}\}) \ni \text{rich(Frank)} = \top$ by unit propagation with the second clause.
- $\mathsf{UP}(s \cup \{\text{fatherOf(Sally)} = \text{Fred}\}) \ni \text{rich(Fred)} = \top$ by unit propagation with the third clause.
- $\mathsf{UP}(s \cup \{\text{fatherOf(Sally)} = n\})$ for $n \notin \{\text{Frank}, \text{Fred}\}$ contains the empty clause by unit propagation with the first clause.

In each case, we obtain a clause that subsumes c, so for every potential father n, $c \in \mathsf{UP}(s \cup \{\text{fatherOf(Sally)} = n\})$.

Classical Semantics

For future reference, we briefly give the classical, "unlimited" semantics of objective formulas. A *world* w is a function from function terms to standard names. Truth of an objective formula α in a world w, written $w \models \alpha$, is defined as follows:

- $w \models t = n$ iff $w(t) = n$
- $w \models \neg\alpha$ iff $w \not\models \alpha$
- $w \models (\alpha \vee \beta)$ iff $w \models \alpha$ or $w \models \beta$

We write $s \models \alpha$ to say that for all w, if $w \models c$ for all $c \in s$, then $w \models \alpha$. Moreover, we write $\models \alpha$ for $\emptyset \models \alpha$.

Limited belief is sound as well as eventually complete with respect to classical semantics in the following sense:

Proposition 3 *For all finite s and all α, there is a (large-enough) belief level $k \geq 0$ such that $s \approx \mathbf{B}_k\alpha$ iff $s \models \alpha$.*

Proof sketch. Soundness holds because Rule 6 branches over all names. Eventual completeness holds because k can be chosen large enough to split all terms in s, α. □

2.3 The Limited Belief Reasoning Problem

The fundamental problem of reasoning about limited belief is to decide whether for a given knowledge base KB and a query α, *if* KB *is known explicitly, then α is known at belief level k*. In limited belief, KB is typically assumed to be CNF [Lakemeyer and Levesque, 2016]. The formal definition is:

Limited Belief Reasoning	
Instance:	Objective formulas KB and α over function terms $\mathcal{F}$ and standard names $\mathcal{N}$, KB in CNF, belief level $k \geq 0$.
Problem:	Decide whether $\approx \mathbf{B}_0\text{KB} \supset \mathbf{B}_k\alpha$.

We shall investigate this problem using classical complexity theory first and then refine the picture using parameterized complexity theory for parameters k, $|\mathcal{F}|$, $|\mathcal{N}|$. An overview of the results is in Table 1.

Since the knowledge base in Limited Belief Reasoning is assumed to be in CNF, it corresponds to a unique (modulo UP) set of clauses and the problem can be equivalently expressed as a model checking problem:

Lemma 4 *Let KB be in CNF with clauses $s = \{c_1, \ldots, c_j\}$. Then $\approx \mathbf{B}_0\text{KB} \supset \mathbf{B}_k\alpha$ iff $s \approx \mathbf{B}_k\alpha$.*

Thus and by Lemma 2 and Proposition 3, Limited Belief Reasoning is sound and eventually complete with respect to classical reasoning: $\approx \mathbf{B}_0\text{KB} \supset \mathbf{B}_{|\mathcal{F}|}\alpha$ iff $\models \text{KB} \supset \alpha$.

Finally, the following lemma says that in Rule 6 a finite number of function terms and standard names is sufficient.

Lemma 5 *Let $\mathcal{F}$ (resp. $\mathcal{N}$) contain all function terms (resp. standard names) in s, α, and let $\hat{n} \notin \mathcal{N}$ be an additional name. Then $s \approx \mathbf{B}_{k+1}\alpha$ iff for some $t \in \mathcal{F}$, for all $n \in \mathcal{N} \cup \{\hat{n}\}$, $s \cup \{t = n\} \approx \mathbf{B}_k\alpha$.*

Together, Lemmas 4 and 5 give rise to a *decision procedure* for Limited Belief Reasoning, which works as follows. First, the problem is turned into the equivalent model checking problem using Lemma 4. Then the procedure applies Lemma 5 to reduce the belief level, and finally follows Rules 1–5 to break α down to clause level and check the clauses for subsumption. It is already known that this procedure runs in time $\mathcal{O}(2^k(|\text{KB}| + |\alpha|)^{k+3})$ [Schwering, 2017].

3 Ordering Gadget

It is easy to see that the ordering in which terms are split can be relevant. For example, let s contain the following four clauses:

$$f = n \vee g_1 = n \vee h = n \qquad f \neq n \vee g_2 = n \vee h = n$$
$$f = n \vee g_1 \neq n \vee h = n \qquad f \neq n \vee g_2 \neq n \vee h = n$$

We can prove $s \approx \mathbf{B}_2 h = n$ by splitting f first and then, depending on the value of f, splitting g_1 or g_2 next, but not the other way around.

In this section we construct a gadget that generalizes this idea in order to enforce that a goal formula can only be proved by splitting terms in a certain order (at polynomial cost in space). This gadget is used repeatedly in the proofs of Sections 4 and 5. For example, in Theorem 9 we use it to preserve the quantifier ordering of the quantified Boolean formula.

To begin with, the following lemma shows how to make sure that one of the terms from a set F is split no later than at belief level k. We use the notation $[k]$ for $\{1, \dots, k\}$.

Lemma 6 *Let F be a non-empty finite set of function terms, and L be a set of literals where every term from F occurs exactly once. Let $\mathbf{B}_k \alpha$ be a formula with $k \geq 1$. Let s be a set of clauses such that for all $t \notin F$, for some name n, $s \cup \{t = n\} \not\models \bigvee_{\ell \in L} \ell$ and $s \cup \{t = n\} \not\models \bigvee_{\ell \in L} \neg\ell$.*

Let $\ell_i^{\mathrm{w}}, \ell_j^{\mathrm{o}}$ for $i \in [k-1], j \in [k]$ be literals with distinct function terms $f_i^{\mathrm{w}}, f_j^{\mathrm{o}}$ that do not occur in s or α. Let c_k^{o} stand for $\ell_1^{\mathrm{o}} \vee \ldots \vee \ell_k^{\mathrm{o}}$. Let s_k be the least set that includes s and for every $\ell \in L$ contains the clauses

- $\neg\ell \vee c_k^{\mathrm{o}}$
- $\ell \vee \ell_1^{\mathrm{w}} \vee \ldots \vee \ell_{k-1}^{\mathrm{w}} \vee c_k^{\mathrm{o}}$
- $\ell \vee \neg\ell_1^{\mathrm{w}} \vee c_k^{\mathrm{o}}, \ \ldots, \ \ell \vee \neg\ell_{k-1}^{\mathrm{w}} \vee c_k^{\mathrm{o}}$.

Then

$$s_k \approx \mathbf{B}_k (c_k^{\mathrm{o}} \wedge (\textstyle\bigvee_{\ell \in L} \neg\ell \vee \alpha)) \quad \textit{iff}$$
for some $t \in F$, for all names n,
$$s \cup \{t = n\} \approx \mathbf{B}_{k-1}(\textstyle\bigvee_{\ell \in L} \neg\ell \vee \alpha).$$

Proof. The proof proceeds in four steps.

Claim 1. $s_k \setminus s \not\approx \mathbf{B}_{k-1} c_k^{\mathrm{o}}$.

Proof of Claim 1. By assumption, no two literals in L are complementary. Hence the only way of proving c_k^{o} is by splitting some $t \in F$ and $f_1^{\mathrm{w}}, \dots, f_{k-1}^{\mathrm{w}}$, which requires belief level k. The proof is by induction on k.

Claim 2. For all $t \notin F \cup \{f_1^{\mathrm{w}}, \dots, f_{k-1}^{\mathrm{w}}, f_1^{\mathrm{o}}, \dots, f_k^{\mathrm{o}}\}$, for some n, $s_k \cup \{t = n\} \not\approx \mathbf{B}_{k-1} c_k^{\mathrm{o}}$.

Proof of Claim 2. By assumption, there is some n such that $s \cup \{t{=}n\} \not\models \bigvee_{\ell \in L} (\neg)\ell$ for all $\ell \in L$. Hence and since $f_i^{\mathrm{w}}, f_j^{\mathrm{o}}$ do not occur in s, $s_k \cup \{t{=}n\} \not\approx \mathbf{B}_{k-1} c_k^{\mathrm{o}}$ iff $s_k \setminus s \not\approx \mathbf{B}_{k-1} c_k^{\mathrm{o}}$, which holds by Claim 1.

Claim 3. For all $t \in F$, for all n, $s_k \cup \{t = n\} \approx \mathbf{B}_{k-1} c_k^{\mathrm{o}}$.

Proof of Claim 3. Let $t \in F$ and let n be an arbitrary name. Then for all names $n_1, \dots, n_k$, $c_k^{\mathrm{o}} \in \mathsf{UP}(s_k \cup \{t = n, f_1^{\mathrm{w}} = n_1, \dots, f_{k-1}^{\mathrm{w}} = n_{k-1}\})$. Thus $s_k \cup \{t = n\} \approx \mathbf{B}_{k-1} c_k^{\mathrm{o}}$.

Proof of the lemma. For the only-if direction, by Claim 2, $s_k \cup \{t = n\} \approx \mathbf{B}_{k-1}(c_k^{\mathrm{o}} \wedge (\bigvee_{\ell \in L} \neg\ell \vee \alpha))$ for all n for some $t \in F \cup \{f_1^{\mathrm{w}}, \dots, f_{k-1}^{\mathrm{w}}, f_1^{\mathrm{o}}, \dots, f_k^{\mathrm{o}}\}$. By Claim 3 and since $f_i^{\mathrm{w}}, f_j^{\mathrm{o}}$ do not occur in s or α, we can assume $t \in F$.

For the converse direction, if n is such that $t = n$ is complementary to $\neg\ell$ for some $\ell \in L$, then $c_k^{\mathrm{o}} \in \mathsf{UP}(s_k \cup \{t = n\})$. Otherwise, $t{=}n$ subsumes $\neg\ell$ for some $\ell \in L$ and by Claim 3, the remaining splits suffice to prove c_k^{o}. ☐

The next lemma represents our gadget. Despite its somewhat intimidating interface, it simply plugs together repeated applications of the previous lemma to completely determine the ordering of splitting terms from $F_1, \dots, F_l$:

Lemma 7 *Let $F_1, \dots, F_l$ be non-empty finite sets of function terms, $F = F_1 \cup \ldots \cup F_l$, and $L_1, \dots, L_l$ be sets of literals such that every term from F_i occurs exactly once in L_i. Let $\mathbf{B}_k \alpha$ be a formula with $k \leq l$. Let s be a set of clauses such that for all $k \in [l]$, for all $t_l \in F_l, \dots, t_{k+1} \in F_{k+1}, t_k \notin F$, for all $n_l, \dots, n_{k+1}, n_k$, $s \cup \{t_k = n_k, \dots, t_l = n_l\} \not\models \bigvee_{\ell \in L_k} \ell$.*

For every set of clauses s', let s_i' and c_i^{o} be as in Lemma 6 with respect to F_i, L_i, α. Let α_0 be α and α_i for $i > 0$ be $c_i^{\mathrm{o}} \wedge (\bigvee_{\ell \in L_i} \neg\ell \vee \alpha_{i-1})$.

Then

$$((s_1) \ldots)_k \approx \mathbf{B}_k \alpha_k \quad \textit{iff}$$
for some $t_k \in F_k$, for all names n_k, …,
for some $t_1 \in F_1$, for all names n_1,
$$s \cup \{t_1 = n_1, \dots, t_k = n_k\} \approx \textstyle\bigvee_{\ell \in L_i, i \in [k]} \neg\ell \vee \alpha.$$

Proof. By induction on k, where Lemma 6 can be applied since $s' \not\models \bigvee_{\ell \in L_i} (\neg)\ell$ implies $((s_1') \ldots)_j \not\models \bigvee_{\ell \in L_i} (\neg)\ell$. ☐

4 Classical Complexity

This section analyzes the complexity of Limited Belief Reasoning using classical complexity theory. The next tractability result follows from the decision procedure from Section 2.3:

Corollary 8 *Limited Belief Reasoning with constant k or constant $|\mathcal{F}|$ is in* PTIME.

Proof. The decision procedure runs in time polynomial with degree $k + 3$. By Lemmas 1 and 2, $k = |\mathcal{F}|$ suffices. ☐

Next, we consider the case where neither k nor $|\mathcal{F}|$ is constant. It comes as no surprise that the complexity then significantly increases with the number of case splits. Proposition 3 and Lemma 4 showed that Limited Belief Reasoning is sound and eventually complete with respect to classical reasoning. So clearly, Limited Belief Reasoning must be co-NP-hard, and eventual completeness may suggest that it is co-NP-complete as well. However, limiting the number of case splits further adds to the computational complexity: whereas in classical reasoning a decision procedure may "simply" split all function terms, a decision procedure for limited belief needs to make sure it makes use of the available case splits in the best possible way. This leads to the following result:

Theorem 9 *Limited Belief Reasoning with constant $|\mathcal{N}|$ is* PSPACE*-complete. The result also holds when $|\mathcal{N}|$ is input.*

Proof. Membership. The decision procedure from Section 2.3 runs in space $\mathcal{O}(m + k)$ where $m = |\mathrm{KB}| + |\alpha|$, since $\mathsf{UP}(s)$ can be represented in space $\mathcal{O}(|s|)$ because minimal clauses suffice.

Hardness. We reduce from True Quantified Boolean Formula, which is PSPACE-complete [Arora and Barak, 2009]. The problem input is a fully quantified Boolean formula $Q_k x_k \ldots Q_1 x_1 \psi$ for $Q_i \in \{\forall, \exists\}$ and a propositional formula ψ. Without loss of generality, we assume that ψ mentions negation only in front of variables. The question is whether this formula evaluates to TRUE, that is, for all (if $Q_k = \forall$) / some (if $Q_k = \exists$) assignment(s) of $x_k, \ldots$, for all / some assignment(s) of x_1, $(x_1, \ldots, x_k)$ satisfies ψ.

Let $\mathcal{N} = \{\top, \mathsf{W}\}$ contain two standard names. Let $\mathcal{F}_\forall = \{f_i \mid Q_i = \forall\}$, $\mathcal{F}_\exists = \{f_i, f_i' \mid Q_i = \exists\}$, $\mathcal{F} = \mathcal{F}_\forall \cup \mathcal{F}_\exists$, where f_i, f_i' are pairwise distinct function terms. We define a mapping $*$ from QBF to limited belief formulas: let x_i^* be $f_i = \top$, let $(\neg x_i)^*$ be $f_i \neq \top$ if $Q_i = \forall$ and $f_i' = \top$ if $Q_i = \exists$, let $(\psi_1 \lor \psi_2)^*$ be $(\psi_1^* \lor \psi_2^*)$, and let $(\psi_1 \land \psi_2)^*$ be $(\psi_1^* \land \psi_2^*)$. For $Q_i = \forall$, let $F_i = \{f_i\}$, $N_i = \{n \mid$ name n is distinct from $\mathsf{W}\}$, and $L_i = \{f_i \neq \mathsf{W}\}$; for $Q_i = \exists$, let $F_i = \{f_i, f_i'\}$, $N_i = \{\top\}$, and $L_i = \{f_i = \top, f_i' = \top\}$.

The idea is as follows. Universally quantified x_i are naturally translated to literals $f_i = \top$ so that the truth values TRUE and FALSE of x_i correspond to $f_i = \top$ and $f_i \neq \top$, respectively. For existentially quantified x_i, positive occurrences of x_i are replaced with $f_i = \top$ and negative ones with $f_i' = \top$ so that the truth values TRUE and FALSE of x_i correspond to $f_i = \top$ and $f_i' = \top$. The f_i or f_i' (if $Q_i = \exists$) then need to be split in the appropriate order.

We can show by induction on k and subinduction on ψ that ϕ evaluates to TRUE iff for some $t_k \in F_k$, for all $n_k \in N_k, \ldots$, for some $t_1 \in F_1$, for all $n_1 \in N_1$, $\{t_1 = n_1, \ldots, t_k = n_k\} \approx\!\!\!\mid \psi^*$. The theorem then follows because the restriction $n_i \in N_i$ on the right-hand side can be lifted by replacing ψ^* with $\bigvee_{\ell \in L_i, i \in [k]} \neg\ell \lor \psi^*$, which then reduces in polynomial time to Limited Belief Reasoning by Lemmas 7 and 4. $\square$

It is noteworthy that this reduction only uses two standard names. With a more involved reduction, even a single name suffices. Thus even the propositional case (where an atomic proposition p is simulated by $p = \top$) is PSPACE-complete.

5 Parameterized Complexity

The gap between tractability and PSPACE-completeness from the previous section calls for a more refined analysis. In this section we use parameterized complexity theory to investigate how the parameters k, $|\mathcal{F}|$, and/or $|\mathcal{N}|$ affect the complexity of Limited Belief Reasoning.

While many unparameterized problems can be classified with the classical classes PTIME, NP, or PSPACE, parameterized versions of these problems fall into a variety of complexity classes [Flum and Grohe, 2006]. The role of PTIME in parameterized complexity is taken on by FPT, which includes problems parameterized by k that are solvable in $f(k) \cdot p(n)$, where f is a computable function and p a polynomial. Other important parameterized classes come from the W- and A-hierarchies: the classes W[1] $\subseteq$ W[2] $\subseteq \ldots \subseteq$ W[SAT] $\subseteq$ W[P] include parameterized versions of different natural NP-complete problems; similarly, the classes A[1] $\subseteq$ A[2] $\subseteq \ldots \subseteq$ AW[SAT] $\subseteq$ AW[P] can be seen as a parameterized version of the polynomial hierarchy. This analogy is a simplification—links between PSPACE and W[1] have recently been exhibited [Bonnet *et al.*, 2017]—but it suffices for our purpose and we display an intuitive representation of the relationships between these classes in Figure 1.

Membership in classes of the W- and A-hierarchies can be shown using machines that restrict the number of nondeterministic steps [Chen *et al.*, 2005]. An NRAM is a random access machine with a nondeterministic `EXISTS` instruction which guesses a natural number less than or equal to a certain register and stores the number in that register. A problem is in W[P] iff it is decidable by an NRAM in $f(k) \cdot p(n)$ steps, at most $g(k)$ of them being nondeterministic, using at most the first $f(k) \cdot p(n)$ registers and only numbers $\leq f(k) \cdot p(n)$. A problem is in AW[P] iff it is decidable by an ARAM with the same constraints, where an ARAM is an NRAM with an additional nondeterministic `FORALL` instruction, the dual to `EXISTS`. Hardness in parameterized complexity is shown by way of fpt-reductions, which are reductions computable in time $f(k) \cdot p(n)$ and such that $k' \leq g(k)$, where k and k' are the parameters of the problems reduced from and reduced to, respectively, n is the input size, f and g are computable functions, and p is a polynomial.

Before starting our analysis, we introduce the *Quantified Monotone Circuit Satisfiability* problem. A *circuit* is a directed acyclic graph (V, E) whose vertices are partitioned into input-nodes X of in-degree 0, not-nodes of in-degree 1, and- and or-nodes of in-degree > 0, and a distinguished output-node v_0 of out-degree 0. An assignment $S \subseteq X$ sets inputs S to TRUE and the other ones to FALSE and propagates the values to the output node, whose value then determines whether or not S satisfies the circuit. A *monotone* circuit contains no not-nodes.

Quantified Monotone Circuit Satisfiability
Instance: A monotone circuit C with input-nodes partitioned into sets $X_1, \ldots, X_l$.
Parameter: $k_1 + \ldots + k_l$
Problem: Decide whether for all $S_1 \subseteq X_1$ with $|S_1| = k_1$, for some $S_2 \subseteq X_2$ with $|S_2| = k_2, \ldots$, the assignment $S_1 \cup \ldots \cup S_l$ satisfies C.

The following lemma states AW[P]-completeness for the problem, which has been claimed elsewhere before without explicit proof [Abrahamson *et al.*, 1995].

Lemma 10 *Quantified Monotone Circuit Satisfiability is* AW[P]*-complete.*

Proof. It is sufficient to reduce from Quantified Circuit Satisfiability, which is AW[P]-complete [Flum and Grohe, 2006]. Consider an instance with circuit $C = (V, E)$ and inputs $X_1, \ldots, X_l$. By De Morgan's laws we can assume the not-nodes are right above the inputs. Observe that a not-node $w \in V$ with input $x \in X_i$ is TRUE iff at least k_i variables in $X_i \setminus \{x\}$ are set to TRUE. The latter property can be expressed in a monotone circuit of polynomial size using or-nodes $v_{i_1,i_2,t}$ that represent that at least t of $x_{i_1}, \ldots, x_{i_2}$ are set to TRUE, and nodes conjoining pairs of these nodes to express that t' of $x_{i_1}, \ldots, x_{i'}$ and $t - t'$ of $x_{i'+1}, \ldots, x_{i_2}$ are set to TRUE. $\square$

With this lemma we can establish the complexity of Limited Belief Reasoning parameterized by the belief level:

Theorem 11 *Limited Belief Reasoning with parameter k is* AW[P]*-complete.*

Proof. Membership. We implement the decision procedure from Section 2.3 using an ARAM. Model checking at belief level 0 can clearly be done on a RAM in time $p(m)$, where p is a polynomial and $m = |\mathrm{KB}| + |\alpha|$. At belief level $k > 0$ we select one of the function terms from $\mathcal{F}$ with `EXISTS`, and the corresponding name from $\mathcal{N} \cup \{\hat{n}\}$ with FORALL. This amounts to $2 \cdot k$ nondeterministic steps and a total runtime $2 \cdot k \cdot p(m)$, so the problem is in AW[P].

Hardness. We reduce from Quantified Monotone Circuit Satisfiability, which is AW[P]-complete by Lemma 10. Let $C = (V, E)$ be a monotone circuit with inputs $X_1, \ldots, X_l$. We say X_i or $x \in X_i$ is *universal* (*existential*) iff i is odd (even).

Let $\mathcal{F} = \{f_v \mid v \in V\} \cup \{f_{i,j} \mid X_i \text{ universal}, j \in [k_i]\} \cup \{f_{x,j} \mid x \in X_i \text{ existential}, j \in [k_i]\}$ be function terms. Let $\mathcal{N} = \{\top, \mathsf{W}\} \cup \{n_x \mid x \in X_i \text{ universal}\}$ be standard names.

The idea is to represent that a node v is set to TRUE by $f_v = \top$. The truth assignment is selected by splitting $f_{i,1}, \ldots, f_{i,k_i}$ one after another for universal X_i, and by splitting some k_i of $\{f_{x,j} \mid x \in X_i, j \in [k_i]\}$ for existential X_i. Truth of a universal input $x \in X_i$ is represented by $f_{i,j} = n_x$ for some $j \in [k_i]$, and truth of an existential input $x \in X_i$ is represented by $f_{x,j} = \top$ for some $j \in [k_i]$, both of which trigger $f_x = \top$; these values are then propagated to the output node, so that $f_{v_0} = \top$ indicates that the circuit is satisfied.

This is encoded in a set of clauses s in the following way. For universal X_i let s_i be the least set that for all $j \in [k_i]$ and $x \in X_i$ contains $f_{i,j} \neq n_x \vee f_x = \top$, and $\bigvee_{x \in X_i} f_{i,j} = n_x \vee f_{i,j} = \mathsf{W}$. For existential let s_i be the least set that for all $j \in [k_i]$ and $x \in X_i$ contains $f_{x,j} \neq \top \vee f_x = \top$. Now let s be the least set such that $s \supseteq s_i$ for all $i \in [l]$, and that contains $\bigvee_{w \in W} f_w \neq \top \vee f_v = \top$ for every and-node v and its inputs $W = \{w \mid (w, v) \in E\}$, and $f_w \neq \top \vee f_v = \top$ for all or-nodes v and all inputs w with $(w, v) \in E$.

It is then straightforward to show by induction on l and subinduction on the depth of C that for all $S_1 \subseteq X_1$ with $|S_1| \leq k_1$, for some $S_2 \subseteq X_2$ with $|S_2| \leq k_2, \ldots$, the truth assignment $S_1 \cup \ldots \cup S_l$ satisfies C iff for some $t_{1,1} \in F_{1,1}$, for all $n_{1,1} \in N_{1,1}, \ldots$, for some $t_{l,k_l} \in F_{l,k_l}$, for all $n_{l,k_l} \in N_{l,k_l}$, $s \cup \{t_{1,1} = n_{1,1}, \ldots, t_{l,k_l} = n_{l,k_l}\} \mathrel{|\!\!\approx} f_{v_0} = \top$, where $F_{i,j} = \{f_{i,j}\}$ and $N_{i,j} = \{n_x \mid x \in X_i\}$ for universal X_i, and $F_{i,j} = \{f_{x,j} \mid x \in X_i\}$ and $N_{i,j} = \{\top\}$ for existential X_i. The right-hand side can be rewritten to match Lemma 7 for $L_{i,j} = \{f_{i,j} \neq \mathsf{W}\}$ for universal X_i and $L_{i,j} = \{f_{x,j} = \top \mid x \in X_i\}$ for existential X_i, and thus fpt-reduces to Limited Belief Reasoning by Lemmas 7 and 4, which gives us AW[P]-hardness. □

Membership in AW[P] is quite natural due to the alternation of existential and universal quantifications of case splits in Lemma 5. When the number of standard names $|\mathcal{N}|$ becomes a parameter as well, this gives us leverage to replace the nondeterministic FORALL steps that select the standard names with a simple loops. It is therefore not surprising that Limited Belief Reasoning parameterized by k and $|\mathcal{N}|$ is in W[P], the hardest NP-analogue of the W-hierarchy. The following result shows that the problem is in fact W[P]-complete:

Proposition 12 *Limited Belief Reasoning with parameters k and $|\mathcal{N}|$ is* W[P]*-complete. The result also holds when $|\mathcal{N}|$ is constant.*

Proof. Membership. We build an NRAM. For $k = 0$, it behaves like the ARAM in Theorem 11. For $k > 0$, we select a function term from $\mathcal{F}$ with EXISTS and loop over all names in $\mathcal{N} \cup \{\hat{n}\}$. This requires $(|\mathcal{N}| + k)^k$ nondeterministic steps.

Hardness. We reduce from Weighted Monotone Circuit Satisfiability, which is W[P]-complete [Abrahamson *et al.*, 1995] and identical to the quantified problem with only a single block of existential variables, and the proof accordingly carries over from Theorem 11. □

Next we consider the case where $|\mathcal{F}|$ becomes a parameter. The below theorem specifies co-W[P]-completeness:

Theorem 13 *Limited Belief Reasoning with parameters k and $|\mathcal{F}|$ is* co-W[P]*-complete. The result also holds when k is input.*

Proof. Membership. We show that the co-problem is in W[P] using an NRAM that finds a falsifying assignment of names for all split terms. As in Theorem 11, the case $k = 0$ is straightforward. For $k > 0$ we loop over all function terms in $\mathcal{F}$ and for each we select a standard name from $\mathcal{N} \cup \{\hat{n}\}$ with EXISTS. This requires $|\mathcal{F}|^{|\mathcal{F}|}$ nondeterministic steps.

Hardness. We reduce from the complement of Weighted Anti-Monotone Circuit Satisfiability, which is W[P]-complete [Flum and Grohe, 2006]. A circuit is anti-monotone when all inputs have out-degree 1 and feed into a not-node and there are no other not-nodes except those on top of some input. Let $C = (V, E)$ be an anti-monotone circuit with inputs X. For every $x \in X$ we denote the associated not-node by v_x.

Let $\mathcal{F} = \{f_i \mid i \in [k]\} \cup \{f\}$ be function terms. Let $\mathcal{N} = \{n_v \mid v \in V \setminus X\} \cup \{\mathsf{W}\}$ be standard names.

The idea is to represent that a node v is set to FALSE by $f \neq n_v$. The truth assignment is selected by splitting $f_1, \ldots, f_k$. Truth of an input x is represented by $f_i = n_{v_x}$ for some $i \in [k]$, which triggers $f \neq n_{v_x}$; these values are propagated to the output node, so that $f \neq n_{v_0}$ indicates that the circuit is falsified.

This is encoded in a set of clauses s as follows. For every $i \in [k]$, let s_i be the least set that contains $f_i \neq n_{v_x} \vee f \neq n_{v_x}$ for every $x \in X$, and $\bigvee_{x \in X} f_i = n_{v_x} \vee f_i = \mathsf{W}$. Let s be the least set such that $s \supseteq s_i$ for all $i \in [k]$, and $f_i \neq n_{v_x} \vee f_j \neq n_{v_x}$ for every $i, j \in [k]$ with $i \neq j$ and $x \in X$, and an encoding of the and- and or-nodes analogous to the one from the proof of Theorem 11.

We then prove that every $S \subseteq X$ with $|S| = k$ falsifies C iff for all $n_1, \ldots, n_k \in \{n_{v_x} \mid x \in X\}$, $s \cup \{f_1 = n_1, \ldots, f_k = n_k\} \mathrel{|\!\!\approx} f \neq n_{v_0}$ by induction on the depth of C. The right-hand side can be rewritten to match Lemma 7 using $L_i = \{f_i \neq \mathsf{W}\}$, and thus fpt-reduces to Limited Belief Reasoning by Lemmas 7 and 4, which gives us co-W[P]-hardness. □

Finally, the only remaining case is when Limited Belief Reasoning is parameterized by both $|\mathcal{F}|$ and $|\mathcal{N}|$:

Proposition 14 *Limited Belief Reasoning with parameters $|\mathcal{F}|$ and $|\mathcal{N}|$ is in* FPT*. This also holds when $|\mathcal{N}|$ is constant.*

Proof. The decision procedure runs in time $(|\mathcal{F}| \cdot (|\mathcal{N}| + k))^k \cdot p(m)$. By Lemma 2 we can estimate $k \leq |\mathcal{F}|$. □

6 Conclusion

We have analyzed the complexity of Limited Belief Reasoning. While tractable for constant belief levels, the complexity jumps to PSPACE-complete in the general case. Using parameterized complexity theory, we showed how parameterized versions of the problem populate the space between these two extremes.

We believe our findings are relevant to the future development of the theory of limited belief. In particular, the insight that the limited belief level can actually increase the computational cost should be considered in future versions.

In light of PSPACE-completeness, one might implement a reasoning system using an off-the-shelf QBF-solver. Also, limited belief may be suitable as a modeling language for other problems in PSPACE.

So far, we have only considered Limited Belief Reasoning without first-order quantification; lifting this restriction would be a natural next step. Moreover, additional parameters could be studied, for example, parameters exploiting the structure of the knowledge base and the query, like backdoors [Gaspers and Szeider, 2012].

Another interesting question is whether our findings carry over to other approaches to resource-bounded reasoning using a similar splitting technique [D'Agostino, 2015].

References

[Abrahamson *et al.*, 1995] Karl A. Abrahamson, Rodney G. Downey, and Michael R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1995.

[Arora and Barak, 2009] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[Bonnet *et al.*, 2017] Édouard Bonnet, Serge Gaspers, Antonin Lambilliotte, Stefan Rümmele, and Abdallah Saffidine. The parameterized complexity of positional games. In *44th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 90:1–90:14, 2017.

[Chen *et al.*, 2005] Yijia Chen, Jörg Flum, and Martin Grohe. Machine-based methods in parameterized complexity theory. *Theoretical Computer Science*, 339(2-3):167–199, 2005.

[Chen *et al.*, 2018] Yijia Chen, Abdallah Saffidine, and Christoph Schwering. The complexity of limited belief reasoning — the quantifier-free case. In *arXiv:1805.02912*, 2018.

[D'Agostino, 2015] Marcello D'Agostino. An informational view of classical logic. *Journal of Theoretical Computer Science*, 606(C):79–97, 2015.

[Delgrande, 1995] James P. Delgrande. A framework for logics of explicit belief. *Computational Intelligence*, 11(1):47–88, 1995.

[Fagin and Halpern, 1987] Ronald Fagin and Joseph Halpern. Belief, awareness, and limited reasoning. *Artificial intelligence*, 34(1):39–76, 1987.

[Flum and Grohe, 2006] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.

[Gaspers and Szeider, 2012] Serge Gaspers and Stefan Szeider. Backdoors to satisfaction. In *The Multivariate Algorithmic Revolution and Beyond*, pages 287–317. Springer, 2012.

[Hintikka, 1975] Jaakko Hintikka. Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4(4):475–484, 1975.

[Kaplan and Schubert, 2000] Aaron N. Kaplan and Lenhart K. Schubert. A computational model of belief. *Artificial Intelligence*, 120(1):119–160, 2000.

[Klassen *et al.*, 2015] Toryn Q. Klassen, Sheila A. McIlraith, and Hector J. Levesque. Towards tractable inference for resource-bounded agents. In *12th International Symposium on Logical Formalizations of Commonsense Reasoning*, pages 89–95, 2015.

[Konolige, 1986] Kurt Konolige. A deduction model of belief. *Research notes in Artificial Intelligence*, 1986.

[Lakemeyer and Levesque, 2013] Gerhard Lakemeyer and Hector J. Levesque. Decidable reasoning in a logic of limited belief with introspection and unknown individuals. In *23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 969–975, 2013.

[Lakemeyer and Levesque, 2014] Gerhard Lakemeyer and Hector J. Levesque. Decidable reasoning in a fragment of the epistemic situation calculus. In *14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 468–477, 2014.

[Lakemeyer and Levesque, 2016] Gerhard Lakemeyer and Hector J. Levesque. Decidable reasoning in a logic of limited belief with function symbols. In *15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 288–297, 2016.

[Lakemeyer, 1994] Gerhard Lakemeyer. Limited reasoning in first-order knowledge bases. *Artificial Intelligence*, 71(2):213–255, 1994.

[Levesque, 1984] Hector J. Levesque. A logic of implicit and explicit belief. In *4th National Conference on Artificial Intelligence (AAAI)*, pages 198–202, 1984.

[Liu *et al.*, 2004] Yongmei Liu, Gerhard Lakemeyer, and Hector J. Levesque. A logic of limited belief for reasoning with disjunctive information. In *9th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 587–597, 2004.

[Patel-Schneider, 1990] Peter F. Patel-Schneider. A decidable first-order logic for knowledge representation. *Journal of Automated Reasoning*, 6(4):361–388, 1990.

[Schwering and Lakemeyer, 2016] Christoph Schwering and Gerhard Lakemeyer. Decidable reasoning in a first-order logic of limited conditional belief. In *22nd European Conference on Artificial Intelligence (ECAI)*, pages 1379–1387, 2016.

[Schwering, 2017] Christoph Schwering. A reasoning system for a first-order logic of limited belief. In *26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1247–1253, 2017.

[Vardi, 1986] Moshe Y. Vardi. On epistemic logic and logical omniscience. In *1st Conference on Theoretical Aspects of Reasoning about Knowledge (TARK)*, pages 293–305, 1986.