Resource-bounded inference with three-valued neighborhood semantics

Toryn Qwyllyn Klassen
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
toryn@cs.toronto.edu

January 6, 2015

Abstract

The traditional view in epistemic logic says that agents see all logical consequences of the information they have, but this would give agents capabilities far beyond what humans display or what is physically realizable. A theory that aims to specify behavior for intelligent agents should provide more realistic guidelines for what reasoning is expected from the agents. To work towards developing such a theory, we introduce an epistemic logic which combines a three-valued version of neighborhood semantics with a mechanism for talking about the amount of effort used in drawing conclusions. We discuss the advantages of this logic over some preceding approaches. However, we also argue that further work is needed to find a notion of effort that can better describe human performance, and make some suggestions for how this could be done.

Acknowledgements This work was funded by the National Science and Engineering Research Council of Canada, and by an Ontario Graduate Scholarship. Much of the material in this paper is based on (Klassen et al., forthcoming).

Contents

1 Introduction **3**
1.1 Effort and human reasoning 4

2 Background **6**
2.1 Preliminaries 6
2.1.1 Mathematical notation 6
2.1.2 Logic 6
2.2 Logical omniscience and alternatives 9
2.2.1 Levels of belief 11
2.2.2 Neighborhood semantics 14
2.2.3 Three-valued neighborhood semantics 15

3 A logic combining neighborhood semantics and levels of belief **17**
3.1 Closure properties 17
3.2 Syntax 18
3.3 Semantics 19
3.4 Properties 20
3.4.1 Associativity 22
3.4.2 Distributivity 24
3.5 A reasoning service 25
3.5.1 Complexity 25

4 Evaluating effort **28**
4.1 Memory 30

5 Extensions **31**
5.1 A parameterized logic 31
5.1.1 On the bushiness of trees 33
5.2 Introspection 34
5.3 Other possibilities 34

6 Conclusion **35**

1 Introduction

Much work within the field of artificial intelligence, especially in the knowledge representation community, has focused on finding more efficient ways to solve complicated puzzles. In a critical remark, Brooks (1990) famously noted that elephants don't play chess, yet still have an interesting level of intelligence. We may also note that most of the reasoning that people engage in day-to-day does not seem to involve complicated planning or scheduling problems. While in chess games people may have to calculate carefully, most tasks – even intellectual tasks beyond the ability of elephants, like casual reading – tend to involve much less effort. That commonsense reasoning is easy is not incidental, but rather what makes such reasoning widespread and useful in everyday life. Part of understanding common sense is knowing what it does *not* encompass, e.g. being able to solve complicated puzzles that merely happen to mention commonplace objects like piggy banks or broken eggs.

We would like to have a formal system that tells us which inferences are reasonably easy for an agent to make and which are hard – an epistemic logic in which we can say that things are obvious (or a doxastic logic, but we will not be distinguishing between belief and knowledge). Let us note that the standard approach in epistemic logic, following Hintikka (1962), does not fit our purpose at all. In the standard approach, an agent's uncertainty about the world is modeled with a set of "possible worlds", each of which is associated with a truth assignment that describes one of the different ways the agent thinks reality might be. The agent believes whatever is true in all the worlds. If all sentences in a set Γ are true in every world, then so is any sentence α that is a logical consequence of Γ. Hence the agent believes all logical consequences of its beliefs.

This unrealistic property, termed "logical omniscience" (or "deductive omniscience"), has inspired considerable discussion in the literature. That knowledge and belief in (Hintikka, 1962) were modeled as being "much too strong" was pointed out in (Castañeda, 1964), an otherwise largely positive book review. Hocutt wrote that real people are "logically obtuse" rather than omniscient. Stalnaker (1991) wrote that

> [A]ny kind of information processing or computation is unintelligible as an activity of a deductively omniscient agent. It is hard to see what a logic of knowledge could be for if it were a harmless simplification for it to ignore these activities that are so essential to rationality and cognition.

Hintikka (1962, Section 2.10) suggested an alternative interpretation (albeit an interpretation he did not favor) of his modal "knowledge" operator as indicating what follows from the agent's knowledge, rather than the knowledge itself. This interpretation was taken by Levesque (1984), who wrote that logical omniscience characterizes "implicit belief", but that "explicit" (i.e real) beliefs should be described by a different logic.

The distinction between implicit and explicit belief is useful, and one that we will fre-

quently refer to. However, a single type of explicit belief is not really enough. The amount of effort that an agent is expected to apply to a problem will depend on context. Therefore, we will be following the line of research developed in (Liu et al., 2004; Liu, 2006; Lakemeyer and Levesque, 2013, 2014) in which there is an infinite family of "levels of belief".

An intuitive understanding of a level of belief is that a sentence is in a level if it can be concluded with an amount of effort that is bounded by the number of that level. So, instead of talking about what agents do believe, the logics in these papers describe what agents would believe, conditioned on their spending a given amount of effort. An autonomous agent would need to have some mechanism to determine how much effort was appropriate to spend in a given situation. The logics are not meant to define such a mechanism, but to specify the behavior of a reasoning service that an agent with such a mechanism could make use of. For an idea as to how such a service could be used, see Kowalski (1995), which suggested defining agents to cyclically consume inputs, reason for a bounded amount of time, and take action. The bound would provide a compromise between operating in a deliberative and reactive manner.

The logic we will be developing in this paper is in large part inspired by the logic $\mathcal{ESL}$ from Lakemeyer and Levesque (2014). However, the semantics of $\mathcal{ESL}$ are defined in a rather ad hoc and syntactic way, whereas ours will be based on neighborhood semantics.

1.1 Effort and human reasoning

There are at least two ways in which having a formal measure of effort would be useful. Firstly, as we have said, we want to describe the complexity of tasks that we want a machine to be able to perform. Secondly, agents with introspection could reason about the difficulty of tasks in interesting ways.

To give an example of the first use of dealing with effort, let us consider the task of story understanding, i.e. of being able to answer questions about natural language narratives, a major challenge in AI research (see Mueller (1999)). If we try to model story understanding without taking effort into account, we might come up with something like Reiter (2000), who wrote that

> [W]e suggest that, in their full generality, narratives are best viewed as non-deterministic programs [... W]e define what it means to query a narrative, and discover that this task is formally identical to proving the correctness of programs as studied in computer science.

Querying a narrative, for Reiter, means asking questions about what holds after the actions described in the narrative occur. Reiter's point in viewing narratives as programs is that narratives are not just linear sequences of actions but may contain features associated with programs, like loops (e.g. a character may be described as performing an action until some

outcome is reached). The point is valid, but the problem with viewing narratives in "their full generality", is that that tells us nothing about to what extent we expect agents to understand them.

If we consider naturally occurring narratives, as found for example in works of fiction or news articles, then a couple of points are apparent:

1. Narratives are not, for the most part, designed to be puzzles.
2. To the extent that a narrative contains complicated puzzles, we are not surprised when a human reader fails to figure them out.

So, it is only in the abstract limit that narratives can be arbitrary programs, and we do not expect people to be able to answer all "queries" of a narrative. A theoretical understanding of human narrative understanding should, therefore, tell us what sort of complexity in narratives people can actually deal with.

Let us move on to consider introspective uses of effort. Autoepistemic logic (see (Moore, 1985)) allows for drawing conclusions based on what is and is not known. For illustration, consider Moore's well-known example of autoepistemic reasoning:

> [I]f I did have an older brother I would know about it; therefore, since I don't know of any older brothers, I must not have any.

If we formalize "know" in the brother example in the traditional (logically omniscient) manner, then determining whether "I don't know of any older brothers" may be very difficult. A better formalization might capture the following idea, which is probably what we normally really mean if we say that we would know if we had an other brother:

> If I had an older brother, it would be *obvious* to me that I did.

Some notion of effort would clearly be useful in formalizing that. Moore's brother problem was formalized in (Elgot-Drapkin and Perlis, 1990, Section 6.2) by equating knowing α with having already drawn the conclusion that α. This may suffice for the brother problem in particular, but is not very flexible.

For a more complicated example of agents reasoning about their own knowledge, consider the following problem:

> A classroom is full of students, about to write an exam. The instructor announces (truthfully) that the exam only requires material from up to chapter five in the textbook, and that she expects the exam to be easy.
>
> Formalize how the instructor's announcement might help the students.

Unlike the contrived puzzles often considered in epistemic logic (e.g. the "muddy children" problem discussed at length in (Fagin et al., 1995)), this problem actually describes something that could plausibly occur in everyday life. Furthermore, the machinery provided by

standard epistemic logics (even dynamic epistemic logic, see (van Ditmarsch et al., 2007)) does not seem to be of much help in capturing the important aspects of this problem.

Dealing with the instructor's expectation of the exam being easy clearly requires some notion of effort (and perhaps a great deal more, as the students and instructor might disagree in various ways on what is easy). The knowledge that the test only covers up to chapter five also has interesting consequences. Suppose, for example, that the test contains the following question:

> Name a general who won a major victory in 1976.

Now, the student may reasonably guess that the answer is a name that was mentioned in the first five chapters. So the student may be able to answer correctly, despite not actually recalling anything about what the general did, for example if General Alcazar was the only general mentioned in those chapters. Of course, more generally, the answer might not be simply "mentioned in" a chapter, but rather derivable from the information in the chapter, perhaps using a method described in the chapter.

So that a student can do well on a test does not necessarily mean that they really know much about the subject matter. Formalizing the sort of reasoning we have been describing could conceivably have applications in education. Unfortunately, the logic we will be presenting in this paper is rather too limited to deal with the examples we have presented here. In particular, it will not incorporate any sort of introspection, and – unlike $\mathcal{ESL}$ (Lakemeyer and Levesque, 2014) – our logic will be propositional instead of first-order. However, we hope that it may inform future research.

2 Background

2.1 Preliminaries

2.1.1 Mathematical notation

Suppose S and T are sets. We will write $S \to T$ to denote the set of all functions from S to T. The power set of S, i.e. the set of all subsets of S, will be denoted by $\mathscr{P}(S)$. If $\prec$ is a partial order, then $\min_{\prec}(S)$ is the set of minimal elements of S according to $\prec$.

2.1.2 Logic

A propositional *logic* is defined by three things: a language (set of sentences), a set of semantic objects, and a satisfaction relation between semantic objects and sentences (which indicates which semantic objects make which sentences true). The language determines the *syntax* of the logic, while the semantic objects and satisfaction relation together determine the *semantics*.

Let us assume that we have some non-empty set (possibly infinite) Φ of symbols, that we will call *atomic* symbols or *atoms*. The propositional language $\mathscr{L}(\Phi)$ is defined by the grammar

$$\alpha ::= p \mid (\alpha \wedge \alpha) \mid \neg\alpha$$

where $p \in \Phi$. As is conventional, $\wedge$ is meant to be understood as a conjunction operator and $\neg$ as a negation operator. We can define other operators like disjunction, the material conditional, and equivalence as the usual abbreviations:

$$\begin{aligned}(\alpha \vee \beta) &:= \neg(\neg\alpha \wedge \neg\beta)\\ (\alpha \supset \beta) &:= (\neg\alpha \vee \beta)\\ (\alpha \equiv \beta) &:= ((\alpha \supset \beta) \wedge (\beta \supset \alpha))\end{aligned}$$

Finally, we define the complement of a formula:

$$\overline{\alpha} = \begin{cases}\alpha_1 & \text{if } \alpha = \neg\alpha_1 \text{ for some formula } \alpha_1\\ \neg\alpha & \text{otherwise}\end{cases}$$

Lowercase Latin characters $p, q, r, \ldots$ will typically be used to denote atoms in Φ, and Greek letters $\alpha, \beta, \gamma, \ldots$ to denote sentences of $\mathscr{L}(\Phi)$. We may use subscripts on letters. The *length* of a sentence α, written $\mathsf{len}(\alpha)$, is defined inductively as follows:

$$\begin{aligned}\mathsf{len}(p) &= 1\\ \mathsf{len}(\neg\alpha) &= 1 + \mathsf{len}(\alpha)\\ \mathsf{len}((\alpha \wedge \beta)) &= 3 + \mathsf{len}(\alpha) + \mathsf{len}(\beta)\end{aligned}$$

Definition 1 (atoms mentioned by a sentence)**.** For $\alpha \in \mathscr{L}(\Phi)$, the set of atoms mentioned by α, written $\mathsf{at}(\alpha)$, is defined inductively as follows:

- $\mathsf{at}(p) = \{p\}$
- $\mathsf{at}(\neg\alpha) = \mathsf{at}(\alpha)$
- $\mathsf{at}((\alpha_1 \wedge \alpha_2)) = \mathsf{at}(\alpha_1) \cup \mathsf{at}(\alpha_2)$

In other words, the atoms mentioned by a sentence are just those that appear in the sentence when it is written down.

We will be considering two different logics using the language $\mathscr{L}(\Phi)$, the classical two-valued logic and a three-valued logic (specifically, Kleene's three-valued logic from (Kleene, 1938)). The two classical *truth values* are T ("true") and F ("false"), and for three-valued

logic there is a third truth value that we will call N. Let

$$\mathbb{C} \coloneqq \{\mathsf{T}, \mathsf{F}\}$$

and let

$$\mathbb{K} \coloneqq \{\mathsf{T}, \mathsf{F}, \mathsf{N}\}.$$

The semantic objects of three-valued logic are functions, called *truth assignments*, from the set $\Phi \to \mathbb{K}$. We will denote the satisfaction relation of three-valued logic by $\vDash_{\mathbb{K}}$. For $v \in \Phi \to \mathbb{K}$ and $\alpha \in \mathscr{L}(\Phi)$, $v \vDash_{\mathbb{K}} \alpha$ iff $v'(\alpha) = \mathsf{T}$, where $v' \in \mathscr{L}(\Phi) \to \mathbb{K}$ is the function defined in terms of v as follows:

- $v'(p) = v(p)$ for $p \in \Phi$
- $v'(\neg\alpha) = \begin{cases} \mathsf{T} & \text{if } v'(\alpha) = \mathsf{F} \\ \mathsf{F} & \text{if } v'(\alpha) = \mathsf{T} \\ \mathsf{N} & \text{if } v'(\alpha) = \mathsf{N} \end{cases}$
- $v'(\alpha \wedge \beta) = \begin{cases} \mathsf{T} & \text{if } v'(\alpha) = \mathsf{T} \text{ and } v'(\beta) = \mathsf{T} \\ \mathsf{F} & \text{if } v'(\alpha) = \mathsf{F} \text{ or } v'(\beta) = \mathsf{F} \\ \mathsf{N} & \text{otherwise} \end{cases}$

We can identify an element of $\Phi \to \mathbb{K}$ with the set of literals it makes true (a literal is an atom or the negation of an atom). This enables us to compare elements of $\Phi \to \mathbb{K}$ with the subset relation, to talk of them being finite or infinite, and to take intersections and (sometimes) unions. Note that if $u \in \Phi \to \mathbb{K}$ and $v \in \Phi \to \mathbb{K}$, $u \cup v$ might not be an element of $\Phi \to \mathbb{K}$, because there might be some $p \in \Phi$ such that both $p \in u \cup v$ and $\neg p \in u \cup v$.

Definition 2 (compatibility)**.** We say that $u \in \Phi \to \mathbb{K}$ and $v \in \Phi \to \mathbb{K}$ are compatible, written $u \diamond v$, if $u \cup v \in \Phi \to \mathbb{K}$.

Given the three-valued logic we have described, we can think of classical two-valued logic as a restriction of it, which differs only in that truth functions cannot map any atom to N. That is, the semantic objects in classical logic are elements of $\Phi \to \mathbb{C}$ (we can view $\Phi \to \mathbb{C}$ as a subset of $\Phi \to \mathbb{K}$ by identifying functions with their graphs), and the satisfaction relation $\vDash_{\mathbb{C}}$ in classical logic is just the restriction of $\vDash_{\mathbb{K}}$ to two-valued truth functions, i.e. $\vDash_{\mathbb{C}} = \{\langle v, \alpha\rangle \in \vDash_{\mathbb{K}} : v \in \Phi \to \mathbb{C}\}$.

The *valid* sentences or *tautologies* of a logic are those which are satisfied by every semantic object. If α is a sentence in the language of a logic with satisfaction relation $\vDash$, then we will write $\vDash \alpha$ to indicate that α is valid in that logic. Note that there are no valid sentences in Kleene's three-valued logic, since $\emptyset \in \Phi \to \mathbb{K}$ and $\emptyset$ does not make any sentence true. For

a set of sentences Γ, we will write $\Gamma \vDash \alpha$ if every semantic object that makes every $\gamma \in \Gamma$ true also makes α true. We may write $\gamma \vDash \alpha$ to mean $\{\gamma\} \vDash \alpha$.

In any logic, the *proposition* expressed by a sentence is the set of semantic objects that satisfy that sentence. For classical and three-valued logics, let us introduce some notation to denote the propositions expressed by sentences:

$$[\![\alpha]\!]^{\mathbb{C}} \coloneqq \{v \in \Phi \to \mathbb{C} : v \vDash_{\mathbb{C}} \alpha\}$$
$$[\![\alpha]\!]^{\mathbb{K}} \coloneqq \{v \in \Phi \to \mathbb{K} : v \vDash_{\mathbb{K}} \alpha\}$$

We will also find the following definition useful:

$$|\!|\alpha|\!| \coloneqq \min_{\subset}([\![\alpha]\!]^{\mathbb{K}})$$

That is, $|\!|\alpha|\!|$ is the set of truth assignments making α true while assigning classical truth values to as few atoms as possible. For example, $|\!|p|\!| = \{\{p\}\}$, where by $\{p\}$ we indicate the unique truth function that makes the literal p true but makes no other literals true. Other illustrative examples are $|\!|\neg p|\!| = \{\{\neg p\}\}$, $|\!|(p \vee q)|\!| = \{\{p\}, \{q\}\}$ and $|\!|(p \wedge q)|\!| = \{\{p, q\}\}$.

If there are an infinite number of atoms, $[\![\alpha]\!]^{\mathbb{K}}$ is always either infinite or empty, while $|\!|\alpha|\!|$ is always finite and all elements of it are finite. Note that for any $\alpha, \beta \in \mathscr{L}(\Phi)$, $v \vDash_{\mathbb{K}} \beta$ for all $v \in [\![\alpha]\!]^{\mathbb{K}}$ if and only if $v \vDash_{\mathbb{K}} \beta$ for all $v \in |\!|\alpha|\!|$.

2.2 Logical omniscience and alternatives

As we said in the introduction, the standard approach in epistemic logic results in ascribing logical omniscience to agents. We can illustrate the standard approach, in a simplified form, as follows. Consider the language $\{\mathsf{B}\alpha : \alpha \in \mathscr{L}(\Phi)\}$, where the intended reading of $\mathsf{B}\alpha$, for $\alpha \in \mathscr{L}(\Phi)$, is "$\alpha$ is believed", i.e., believed by the agent (for simplicity we will not consider the multiagent case, nor the case of nested beliefs). We will use sets of truth assignments as semantic objects (the idea is that a set of truth assignments expresses the different ways the agent thinks the world could be). A set W of truth assignment makes $\mathsf{B}\alpha$ true if every one of the truth assignments in W makes α true in classical logic:

$$W \vDash \mathsf{B}\alpha \text{ if, for all } w \in W,\ w \vDash_{\mathbb{C}} \alpha$$

Note that, for $\Gamma \subseteq \mathscr{L}(\Phi)$,

$$\{\mathsf{B}\gamma : \gamma \in \Gamma\} \vDash \mathsf{B}\alpha \text{ iff } \Gamma \vDash_{\mathbb{C}} \alpha$$

So we see that this approach to modeling belief has absolutely nothing to say about cognitive limitations or the complexity of reasoning. An agent's reasoning is just what classical

propositional logic allows. (The traditional approach is not quite as vacuous as our simplified version here, as it does provide for things like nested beliefs, but those elements are not our concern here.)

There have been numerous proposals for epistemic logics that avoid logical omniscience. We will briefly discuss some of them below, with emphasis on the approaches that we will be borrowing elements from. For more information, the reader is referred to the survey papers (McArthur, 1988; Sim, 1997; Moreno, 1998; Whitsey, 2003).

One alternative way of modeling belief is the deduction model of belief, due to Konolige (1984). In this deduction model, agents believe a set of sentences that is closed under some set of proof rules. The set of proof rules can be incomplete, so the agents are not required to believe all logical consequences of their beliefs. Levesque (1984) criticized the deduction model because choosing proof rules was an ad hoc process.

Another rather syntactic approach is the logic of "general awareness", described in (Fagin and Halpern, 1988). This approach takes the traditional possible world semantics and adds an extra component, the set of sentences the agent is "aware" of. Belief works as in the traditional manner, except that agent can only believe formulas that are in the awareness set. Clearly, this approach allows the agent's reasoning power to be arbitrarily limited by appropriate choice of the awareness set. We could imagine modeling different amounts of effort with different awareness sets. However, determining what goes into these awareness sets seems to be an entirely extralogical procedure. Konolige (1986) also criticized the logic of general awareness, writing that

> The practice of mixing sentential and possible-world elements in the semantics does not preserve the elegance of the latter, or offer any essential insight into the psychological nature of explicit belief.

A more semantic approach was taken by Levesque (1984, 1989), who argued that explicit belief should be modeled by a more limited logic than classical logic. He described a semantics (related to relevance logic) based on "situations", which we can think of as functions from $\Phi \to \mathbb{K}$. For $v \in \Phi \to \mathbb{K}$, a satisfaction relation can be defined by $v \models_E \alpha$ if $v'(\alpha) \neq \mathsf{F}$, where v' is defined in terms of v as described for Kleene's logic. An agent could then be modeled with a set of situations, instead of a set of possible worlds. The logic is much weaker than classical logic; for example, $\{p, (\neg p \vee q)\} \not\models_E q$ because the situation v such that $v(p) = \mathsf{N}$ and $v(q) = \mathsf{F}$ satisfies both p and $(\neg p \vee q)$. To strengthen the logic in a controlled way, we could parameterize the satisfaction relation by a set of atoms that are required to take classical truth values, as shown in Schaerf and Cadoli (1995).

However, (Levesque, 1984)'s notion of explicit belief was criticized by Vardi (1986), who wrote that

> [T]he agents in Levesque's model turn out to be perfect reasoners in Anderson's

and Belnap's relevance logic. Unfortunately, it does not seem that agents can reason perfectly in relevance logic any more than in classical logic.

The approach was later criticized by Levesque himself (Lakemeyer and Levesque, 2002) for being too weak in some cases (like the simple $\{p, (\neg p \vee q)\} \not\models_E q$ example) and sometimes requiring too much reasoning from agents (in a first-order version of the approach).

Duc (2001, Chapter 5) presents a logic with numbered knowledge operators, where the numbers are meant to indicate bounds on how much time it would take to verify that the formulas in question are true. However, most of the work in defining what those bounds would be for particular sentences is not done by the logic, but by cost functions that a user of the logic would have to provide.

A rather different way to deal with logical omniscience is provided by step-logics (Elgot-Drapkin, 1988; Elgot-Drapkin and Perlis, 1990). Step-logics are intended to model beliefs changing over time, rather than just specify static properties of belief. At each step in time, the agent has a finite set of believed sentences, and makes a finite number of observations (which are also sentences). The agent has an inference function that determines, based on the history of belief and observation sets, what sentences will constitute its belief set at the next step. In SL_7, the step-logic described in most detail, the inference function is determined by a set of proof rules: at step $i + 1$, the belief set will contain each sentence that can be derived from the union of step i's belief set and observation set using one proof rule application.

We could equate effort with time, and think of SL_7 as measuring effort in terms of the length of a proof (in a proof system in which rules can be used in parallel). In general, proof length is an obvious way to measure effort. However, Crawford and Etherington (1998) argued that reasoning by a particular technique called *unit propagation* is useful and should be supported by a semantics for tractable inference, even though such reasoning may involve long chains of steps. In the next section, we will look a logic that incorporates unit propagation directly into its semantics.

2.2.1 Levels of belief

We will describe a logic that we will call $\mathcal{LL}$, which is essentially a fragment of the logic $\mathcal{ESL}$ from (Lakemeyer and Levesque, 2014). Alternatively, $\mathcal{LL}$ can be seen as a propositional version of the logic $\mathcal{SL}$ described in (Liu et al., 2004; Liu, 2006; Lakemeyer and Levesque, 2013) with a modified splitting rule. By examining $\mathcal{LL}$ we will be able to see features and limitations of those logics that are relevant to us.

The syntax of $\mathcal{LL}$ is given by the grammar below, in which k is any nonnegative integer and $\alpha \in \mathscr{L}(\Phi)$ (except that disjunction instead of conjunction is taken as a primitive

operator):

$$\varphi ::= \mathsf{B}_k\alpha \mid (\varphi \vee \varphi) \mid \neg\varphi$$

An intuitive reading of $\mathsf{B}_k\alpha$ is that "upon being queried about the truth of α, confirmation takes at most k effort", though for brevity we suggest the conventional reading "α is in level k". We can think of B_0 as being an explicit belief operator. It matters that disjunctions are considered primitive, because *clauses* are built into the semantics.

Definition 3 (clause)**.** A literal is a clause, and if c_1 and c_2 are clauses, then so is $(c_1 \vee c_2)$. We may sometimes identify a clause with the set of literals it contains; in such a context, we also consider the empty set to be a clause (the empty clause can be thought of as representing a contradiction).

Unit propagation, usually considered a proof-theoretic notion, is also used in the semantics.

Definition 4. Let S be a set of clauses. Then $\mathcal{UP}(S)$, the closure of S under unit propagation, is the least superset of S such that if $\{\ell\} \in \mathcal{UP}(S)$ and $c \in \mathcal{UP}(S)$, then $c \backslash \{\overline{\ell}\} \in \mathcal{UP}(S)$.

Unit propagation is often defined so as to also remove clauses that are supersets of included unit clauses (probably because without such removal, the closure under unit propagation may be exponentially larger than the original set – with such removal, the closure can be computed in linear time). We will not make that part of our definition, though, since removing subsumed clauses does not change what inferences are licensed if we allow ourselves to reason by subsumption as well.

The semantic objects of $\mathcal{LL}$ are sets of clauses. The satisfaction relation is defined inductively as follows:

1. $S \models_{\mathcal{LL}} (\varphi \vee \psi)$ iff $S \models_{\mathcal{LL}} \varphi$ or $S \models_{\mathcal{LL}} \psi$
2. $S \models_{\mathcal{LL}} \neg\varphi$ iff $S \not\models_{\mathcal{LL}} \varphi$
3. $S \models_{\mathcal{LL}} \mathsf{B}_k\alpha$ iff at least one of the following holds:
   (a) $k = 0$, α is a clause, and there exists $c \in \mathcal{UP}(S)$ such that $c \subseteq \alpha$
   (b) $\alpha = (\alpha_1 \vee \alpha_2)$, and $S \models_{\mathcal{LL}} \mathsf{B}_k\alpha_1$ or $S \models_{\mathcal{LL}} \mathsf{B}_k\alpha_2$
   (c) $\alpha = \neg(\alpha_1 \vee \alpha_2)$, and $S \models_{\mathcal{LL}} \mathsf{B}_k\neg\alpha_1$ and $S \models_{\mathcal{LL}} \mathsf{B}_k\neg\alpha_2$
   (d) $\alpha = \neg\neg\alpha_1$, and $S \models_{\mathcal{LL}} \mathsf{B}_k\alpha$
   (e) $k > 0$ and there exists $p \in \Phi$ such that both $S \cup \{p\} \models_{\mathcal{LL}} \mathsf{B}_{k-1}\alpha$ and $S \cup \{\neg p\} \models_{\mathcal{LL}} \mathsf{B}_{k-1}\alpha$

Rules (1) and (2) deal with logical connectives outside the scope of modal operators. The interesting rules are the various part of rule (3).

Note how rule (3a) allows for reasoning using unit propagation and subsumption at level 0. Rules (3b-d) allow for building up beliefs syntactically in various ways. Rule (3e) determines how higher levels of belief are constructed by splitting cases (i.e., either p is true, or it isn't) and considering lower levels. The idea of using the depth of case-splitting allowed as a measure of effort can also be found in more proof-theory oriented papers like (Finger, 2004; D'Agostino and Floridi, 2009; D'Agostino et al., 2013). The notions of effort in (Crawford and Kuipers, 1991; Dalal, 1996; Crawford and Etherington, 1998) are also related.

The reader might also be reminded of the DPLL algorithm, which is based around unit propagation and case-splitting as well.

The numbered belief operators are increasingly powerful (if a formula is believed at level k, then it will also be believed at level $k+1$), and any classical consequence of S will appear in some (sufficiently high) level of belief. Liu et al. (2004) proposes defining a reasoning service in terms of levels of belief as follows: if you want to determine whether α can be derived using k effort from a "knowledge base" (i.e. a sentence) KB, you can ask whether $\mathsf{B}_0\mathsf{KB} \supset \mathsf{B}_k\alpha$ is valid. (Liu et al., 2004, Theorem 6) showed that, in the propositional version of $\mathcal{SL}$, whether $\mathsf{B}_0\mathsf{KB} \supset \mathsf{B}_k\alpha$ is valid can be determined in polynomial time – if k is held constant.

Unfortunately, the rather syntactic way the semantics are defined results in many sentences not being in a level even when intuitively they seem to follow – without the need to reason by cases – from other sentences in the level. For example, $\mathsf{B}_0(p \vee q) \supset \mathsf{B}_0(p \vee \neg\neg q)$ is not valid in $\mathcal{LL}$, since $(p \vee \neg\neg q)$ is not a clause. For a more elaborate example of the same sort of thing, recall that we are for $\mathcal{LL}$ defining conjunction in terms of the primitive disjunction operator, i.e. $(\alpha \wedge \beta) = \neg(\neg\alpha \vee \neg\beta)$. Suppose we define a new disjunction operator $\vee'$ by $(\alpha \vee' \beta) \coloneqq \neg(\neg\alpha \wedge \neg\beta)$. Then $\nvDash_{\mathcal{LL}} \mathsf{B}_0(p \vee q) \supset \mathsf{B}_0(p \vee' q)$, since $(p \vee' q) = \neg\neg(\neg\neg p \vee \neg\neg q)$ is not a clause (nor the double negation of a clause). Furthermore, associativity fails to hold within modal operators, since e.g. $\nvDash_{\mathcal{LL}} \mathsf{B}_0((p \vee q) \vee \neg\neg r) \supset \mathsf{B}_0(p \vee (q \vee \neg\neg r))$. Note though that $\vDash_{\mathcal{LL}} \mathsf{B}_0((p \vee q) \vee r) \supset \mathsf{B}_0(p \vee (q \vee r))$ because $((p \vee q) \vee r)$ and $(p \vee (q \vee r))$ are clauses.

A more serious problem than those that we have mentioned above is that we have

$$\mathsf{B}_0((p \wedge q) \vee (r \wedge s)) \vDash_{\mathcal{LL}} (\mathsf{B}_0(p \wedge q) \vee \mathsf{B}_0(r \wedge s)).$$

To illustrate why this is a problem, suppose we believe that there will be extreme weather – either it will be very hot and wet, or else very cold and dry. If we formalize this using two atoms – one atom for "hot and wet" and one for "cold and dry" – then of course from a belief in the disjunction of those atoms we can not (and should not) be able to conclude which one is true. The weird thing is that if we use four atoms – one each for "hot", "wet",

"cold", and "dry" – and group them appropriately with conjunctions, then $\mathcal{LL}$ says that there must be a belief as to which of the extreme conditions it will be.

So use of $\mathcal{LL}$ is highly sensitive to how things are formalized. The idea that at heart an agent's knowledge consists of a set of disjunctions of literals seems to be without philosophical or psychological motivation. In section 3, we will seek to combine levels of belief with a form of neighborhood semantics to ameliorate this issue.

2.2.2 Neighborhood semantics

Neighborhood semantics (sometimes called Montague-Scott semantics) for modal logic were suggested by Montague (1968) and Scott (1970). Various forms of these semantics have been used in AI for modeling belief; a survey can be found in (Sim, 1997, Section IV-B).

In this section we will sketch descriptions of a simple form of neighborhood semantics with only one agent and no support for nested beliefs.

The semantic objects are (two-valued) epistemic states, defined below:

Definition 5. A (two-valued) epistemic state is an element of $\mathscr{P}(\mathscr{P}(\Phi \to \mathbb{C}))$, i.e. a set of sets of truth assignments from $\Phi \to \mathbb{C}$.

The intuition is that if $\mathfrak{M}$ is an agent's epistemic state, then for each $V \in \mathfrak{M}$, the agent thinks the world is described by one of the truth assignments in V. If the agent were logically omniscient, it would therefore think that the real world corresponded to one of the truth assignments in $\bigcap \mathfrak{M}$. However, the point of the semantics is that agents do not have to be logically omniscient, i.e. explicit belief can be modeled.

There are two established ways in which we might go about defining how the satisfaction relation treats explicit belief, namely, the strict and loose neighborhood semantics. Let us introduce two modal operators, $[=]$ and $[\subseteq]$, one for each type of explicit belief. The strict neighborhood semantics defines explicit belief by

$$\mathfrak{M} \vDash [=]\alpha \text{ if there exists } V \in \mathfrak{M} \text{ such that } V = [\![\alpha]\!]^{\mathbb{C}}$$

while the loose neighborhood semantics defines it by

$$\mathfrak{M} \vDash [\subseteq]\alpha \text{ if there exists } V \in \mathfrak{M} \text{ such that } V \subseteq [\![\alpha]\!]^{\mathbb{C}}$$

Note that $V \subseteq [\![\alpha]\!]^{\mathbb{C}}$ iff $v \vDash_{\mathbb{C}} \alpha$ for every $v \in V$.

The "strict" and "loose" terminology and the $[=]$ and $[\subseteq]$ notation are from (Areces and Figueira, 2009). Both types of semantics have long been considered in AI research; Vardi (1986) used strict neighborhood semantics, and loose neighborhood semantics were used by the "logic of local reasoning" from (Fagin and Halpern, 1988, Section 6).

The intuitive way to understand the strict neighborhood semantics is to view an epistemic state as simply a set of every proposition that the agent explicitly believes. A problem with this semantics, noted by Vardi (1986) and others, is that if an agent believes a sentence α, then the agent believes any β equivalent to α (so, for example, if the agent believes any one tautology, then the agent believes all tautologies).

From the point of view of the loose neighborhood semantics, an epistemic state is not the set of everything believed, because inferences can be made from each proposition in the epistemic state. Note that if $\alpha \vDash_{\mathbb{C}} \beta$ and $\mathfrak{M} \vDash [\subseteq]\alpha$, then $\mathfrak{M} \vDash [\subseteq]\beta$. This is closer to logical omniscience (some authors have defined logical omniscience as being exactly this), but the agent still cannot bring together information from separate propositions. For example, consider

$$\{[\![p]\!]^{\mathbb{C}}, [\![q]\!]^{\mathbb{C}}\} \nVdash [\subseteq](p \wedge q)$$

or

$$\{[\![p]\!]^{\mathbb{C}}, [\![(p \supset q)]\!]^{\mathbb{C}}\} \nVdash [\subseteq]q.$$

Strict neighborhood semantics are the basis for the "active logic" described in (Nirkhe et al., 1995, Section 4). In this logic, time is represented, and epistemic states expand over time, which is meant to model an agent reasoning. However, as was criticized by (Jago, 2006, Section 4.4.2), all tautologies are believed from time 0 on.

2.2.3 Three-valued neighborhood semantics

Both the strict and weak neighborhood semantics are in a sense too strong, as exemplified in the way they treat belief in tautologies. As we will describe in this section, by basing neighborhood semantics on Kleene's three-valued logic, which has no tautologies, we can go some way towards improving matters. Three-valued neighborhood semantics do not appear to be nearly as much discussed in the literature, though they are described (under the name of "belief cell" semantics) by McArthur (1988, Section 4.2), recounting an unpublished paper by Levesque.

We will relax the definition of an epistemic state to allow it to involve three-valued truth assignments.

Definition 6 (epistemic state)**.** An epistemic state is an element of $\mathscr{P}(\mathscr{P}(\Phi \to \mathbb{K}))$, i.e. a set of sets of truth assignments from $\Phi \to \mathbb{K}$.

We can view two-valued epistemic states as a special case, in which none of the functions involved has N in its image.

The point of three-valued epistemic states is not that the agent thinks that the world is really three-valued. Rather, a three-valued truth assignment provides a *partial* description of the world. Let us make a definition:

Definition 7 (compatibility with an epistemic state)**.** If $\mathfrak{M}$ is an epistemic state and $u \in \Phi \to \mathbb{K}$, then we say that u is compatible with $\mathfrak{M}$ if for each $V \in \mathfrak{M}$ there is some $v \in V$ such that $v \diamond u$.

If an agent's epistemic state is $\mathfrak{M}$, then the agent (implicitly) thinks that the two-valued truth assignment that corresponds to the real world is compatible with $\mathfrak{M}$.

Definition 8 ($\Subset$)**.** Let $U \subseteq \Phi \to \mathbb{K}$ and $V \subseteq \Phi \to \mathbb{K}$. Then $V \Subset U$ if for every $v \in V$, there is some $u \in U$ such that $u \subseteq v$.

Proposition 1. *Let $V \subseteq \Phi \to \mathbb{K}$ and $\alpha \in \mathscr{L}(\Phi)$. Then $v \vDash_{\mathbb{K}} \alpha$ for all $v \in V$ iff $V \Subset \|\alpha\|$.*

Proof. We prove both directions:

- Suppose $V \Subset \|\alpha\|$. Let $v \in V$. Then there is some $u \in \|\alpha\|$ such that $u \subseteq v$. Since $u \vDash_{\mathbb{K}} \alpha$ (by the definition of $\|\alpha\|$), $v \vDash_{\mathbb{K}} \alpha$.
- Suppose $v \vDash_{\mathbb{K}} \alpha$ for all $v \in V$. Then $V \subseteq [\![\alpha]\!]^{\mathbb{K}}$. Since (by definition) $\|\alpha\| = \min_{\subset} [\![\alpha]\!]^{\mathbb{K}}$, it follows that for every $v \in V$, there is some $u \in \|\alpha\|$ such that $u \subseteq v$. Hence $V \Subset \|\alpha\|$.

This establishes the result. □

Note that if $U \subseteq \Phi \to \mathbb{C}$ and $V \subseteq \Phi \to \mathbb{C}$, then $V \Subset U$ iff $V \subseteq U$.

We can define a new modal operator $[\Subset]$ which can be thought of as the three-valued analogue to $[\subseteq]$ as follows:

$$\mathfrak{M} \vDash [\Subset]\alpha \text{ if there exists } V \in \mathfrak{M} \text{ such that } V \Subset \|\alpha\|$$

How does $[\Subset]$ compare with $[\subseteq]$ as an explicit belief operator? It is true that if $\alpha \vDash_{\mathbb{K}} \beta$ and $\mathfrak{M} \vDash [\Subset]\alpha$, then $\mathfrak{M} \vDash [\Subset]\beta$. However, this is often a much less onerous requirement for the agent to fulfill than the classical version of that. Consider that to decide whether $\{[\![(p \vee q)]\!]^{\mathbb{C}}\} \vDash [\subseteq]\alpha$ holds an agent may have to reflect not just on the truth values of p and q, but also on the atoms in α (since, for example, α might be a tautology). On the other hand, to determine if $\{\|(p \vee q)\|\} \vDash [\Subset]\alpha$ all that has to be done is check whether both of the two truth assignments in $\|(p \vee q)\|$ make α true. This is easy, especially since each truth assignment in $\|(p \vee q)\|$ is undefined on every atom but one.

We could also create a three-valued version of the strict neighborhood semantics, but we will not look into that here. We would like epistemic states to be, instead of enumerations of everything believed (which would often be infinite), reasonably compact objects which could be physically realized in a relatively straightforward way.

Let us make one last definition:

Definition 9 (atoms mentioned by an epistemic state)**.** For $\mathfrak{M}$ an epistemic state, the set of atoms mentioned by $\mathfrak{M}$ is the set

$$\mathsf{at}(\mathfrak{M}) \coloneqq \{p \in \Phi : \exists V \in \mathfrak{M}, \exists v \in V \text{ such that } v(p) \neq \mathsf{N}\}.$$

Of course, if $\mathfrak{M}$ is a two-valued epistemic state, then $\mathsf{at}(\mathfrak{M}) = \Phi$.

3 A logic combining neighborhood semantics and levels of belief

While the idea of levels of belief in $\mathcal{LL}$ is appealing, the use of sets of clauses as semantic objects makes the semantics of $\mathcal{LL}$ limited in quirky ways. We will introduce a new logic which is similar to $\mathcal{LL}$ but replaces sets of clauses with epistemic states from three-valued neighborhood semantics.

3.1 Closure properties

As a prelude to developing our logic, let us consider modeling closure properties in neighborhood semantics. We would like for an agent to have *some* ability to combine information from different elements of its epistemic state, without requiring unrealistic reasoning powers. For example, we might like for explicit belief in α and in β to make $(\alpha \wedge \beta)$ also explicitly believed. How can this be achieved?

One obvious approach (followed in e.g. (Vardi, 1986, Section 4)) is to impose a restriction on the set of semantic objects. Let us first make a definition:

Definition 10 ($\Cap$)**.** For $U, V \in \mathscr{P}(\Phi \to \mathbb{K})$, let $U \Cap V \coloneqq \{u \cup v : u \in U, v \in V, \text{ and } u \diamond v\}$.

The intuition behind $\Cap$ is that it is the semantic version of the $\wedge$ operator. Note that $[\![\alpha]\!] \Cap [\![\beta]\!] = [\![\alpha \wedge \beta]\!]$. Also, if none of the functions in U or V assign the value N to any atom, then $U \Cap V = U \cap V$.

Now, a restriction on semantic objects could be to require an epistemic state $\mathfrak{M}$ to satisfy that if $U \in \mathfrak{M}$ and $V \in \mathfrak{M}$, then $U \Cap V \in \mathfrak{M}$. Unfortunately, this sort of approach makes epistemic states much too strong. To illustrate, suppose that $\mathfrak{M}$ is such that $\{[\![\gamma]\!] : \gamma \in \Gamma\} \subseteq \mathfrak{M}$ for some set of sentences Γ. Then, in order to fulfill the requirement it must be that $[\![\bigwedge \Gamma]\!] \in \mathfrak{M}$. That means that the agent explicitly believes *all* the logical consequences (in Kleene's logic) of Γ.

To avoid requiring so much power, we will therefore leave the semantic objects alone. In our logic, we will instead expand the satisfaction relation by providing additional conditions under which explicit belief exists. This is like the approach that $\mathcal{LL}$ takes, of course.

However, this does not mean that restricting the set of semantic objects may not sometimes be useful. Closure under unit propagation plays an important role in $\mathcal{LL}$. We can define a condition for epistemic states that will play a similar role for us.

Definition 11 (harmonization)**.** Let $\mathfrak{M}$ be an epistemic state. The harmonization of $\mathfrak{M}$, written $\mathcal{H}(\mathfrak{M})$, is the least superset of $\mathfrak{M}$ satisfying the following condition: if V and $\{u\}$ are elements, then so is $\{v \in V : v \diamond u\}$.

An epistemic state $\mathfrak{M}$ is said to be harmonized if $\mathfrak{M} = \mathcal{H}(\mathfrak{M})$. The idea behind harmonization is to give a semantic generalization of the proof-theoretic notion of unit propagation. Harmonizing an epistemic state does not confer anything like logical omniscience; for example, if $\mathfrak{M} = \{[\![(p \vee q)]\!], [\![((p \vee q) \supset r)]\!]\}$ then $\mathfrak{M}$ is already harmonized and yet $\mathfrak{M} \nVdash [\Subset]r$.

Proposition 2. *Let S be a set of clauses, and let $\mathfrak{M}_S = \{[\![c]\!] : c \in S\}$. Then $c \in \mathcal{UP}(S)$ iff $[\![c]\!] \in \mathcal{H}(\mathfrak{M}_S)$.*

Proof. Given $u, v \in \Phi \to \mathbb{K}$ where $u = \{\ell_1\}$ and $v = \{\ell_2\}$, note that $v \diamond u$ iff $\ell_1 \neq \bar{\ell}_2$. Also note that if c is a clause, then $[\![c]\!] = \{\{\ell\} : \ell \in c\}$.

Therefore, if $V = [\![c]\!]$ for some clause c, and $u = \{\ell\}$, then

$$\{v \in V : v \diamond u\} = \{v \in \{\{\ell'\} : \ell' \in c\} : v \neq \{\bar{\ell}\}\} = \{\{\ell'\} : \ell' \in c\} \setminus \{\{\bar{\ell}\}\} = [\![c \setminus \{\bar{\ell}\}]\!].$$

So the result follows. □

We are now almost ready to formally define a logic based on three-valued neighborhood semantics that has a version of levels of belief. First, though, let us make a definition.

Definition 12 (expansion)**.** Let $\mathfrak{M}$ be an epistemic state and $\alpha \in \mathscr{L}(\Phi)$. Then the expansion of $\mathfrak{M}$ by α, written $\mathfrak{M}[\alpha]$, is the epistemic state $\mathcal{H}(\mathfrak{M} \cup \{[\![\alpha]\!]\})$.

$\mathfrak{M}[\alpha]$ could be thought of as the epistemic state that results from the agent learning or being told α. $\mathfrak{M}[\alpha]$ might also be a state temporarily entered when the agent assumes α for the sake of argument. A major reason for our using harmonization is so that, if α "obviously" conflicts with the information in $\mathfrak{M}$, $\mathfrak{M}[\alpha]$ will include the empty set (and so make every level of belief contain every sentence). This allows for reasoning by contradiction.

3.2 Syntax

Our logic will use the modal language $\mathscr{M}(\Phi)$, which is defined by the grammar below, in which $\alpha \in \mathscr{L}(\Phi)$, and k is any nonnegative integer.

$$\varphi ::= \mathsf{B}\alpha \mid \mathsf{B}_k\alpha \mid [\Subset]\alpha \mid [\alpha]\varphi \mid (\varphi \wedge \varphi) \mid \neg\varphi$$

Note that sentences of $\mathscr{L}(\Phi)$ cannot appear outside the scope of a modal operator. Also, $[\alpha]$ is the only sort of modal operator for which other modal operators can be in its scope.

3.3 Semantics

The semantic objects of our logic are harmonized epistemic states.

We will next define a satisfaction relation inductively. To make the induction well-founded we will have to use a slightly more complex order on sentences than just length. In preparation for defining this order, we will inductively define two functions f, g mapping $\mathscr{M}(\Phi)$ to integers.

$$\begin{aligned} f(\mathsf{B}\alpha) = f([\Subset]\alpha) &= -1 \\ f(\mathsf{B}_k\alpha) &= k \\ f([\alpha]\varphi) = f(\neg\varphi) &= f(\varphi) \\ f((\varphi \wedge \psi)) &= \max(f(\varphi), f(\psi)) \end{aligned}$$

Note that $f(\varphi)$ is the value of the highest subscript in φ if there is one, and -1 otherwise. We next define g:

$$\begin{aligned} g(\mathsf{B}\alpha) = g(\mathsf{B}_k\alpha) = g([\Subset]\alpha) &= 1 + \mathsf{len}(\alpha) \\ g([\alpha]\varphi) &= 2 + \mathsf{len}(\alpha) + g(\varphi_1) \\ g((\varphi \wedge \psi)) &= 3 + g(\varphi) + g(\psi) \\ g(\neg\varphi) &= 1 + g(\varphi) \end{aligned}$$

If we consider the B_k and $[\Subset]$ operators to have length 1, then $g(\varphi)$ is the length of φ.

Recall the purpose of f and g is to define a partial order on sentences. Let us say that $\varphi \prec \psi$ if $\langle f(\varphi), g(\varphi)\rangle$ lexicographically precedes $\langle f(\psi), g(\psi)\rangle$, i.e. if $f(\varphi) < f(\psi)$ or if both $f(\varphi) = f(\psi)$ and $g(\varphi) < g(\psi)$.

Now we can say that the satisfaction relation, $\models$, is defined by induction on the order $\prec$ as follows:

1. $\mathfrak{M} \models \mathsf{B}\alpha$ iff, for each $w \in \Phi \to \mathbb{C}$ that is compatible with $\mathfrak{M}$, $w \models_{\mathbb{C}} \alpha$
2. $\mathfrak{M} \models [\Subset]\alpha$ iff there exists $V \in \mathfrak{M}$ such that $V \Subset [\![\alpha]\!]$
3. $\mathfrak{M} \models [\alpha]\varphi$ iff $\mathfrak{M}[\alpha] \models \varphi$
4. $\mathfrak{M} \models (\varphi \wedge \psi)$ iff $\mathfrak{M} \models \varphi$ and $\mathfrak{M} \models \psi$
5. $\mathfrak{M} \models \neg\varphi$ iff $\mathfrak{M} \not\models \varphi$
6. $\mathfrak{M} \models \mathsf{B}_k\alpha$, where k is a nonnegative integer, iff at least one of the following holds:
   (a) $k = 0$ and $\mathfrak{M} \models [\Subset]\alpha$
   (b) $\alpha = (\alpha_1 \wedge \alpha_2)$, and $\mathfrak{M} \models \mathsf{B}_k\alpha_1$ and $\mathfrak{M} \models \mathsf{B}_k\alpha_2$

(c) $\alpha = \neg(\alpha_1 \wedge \alpha_2)$, and $\mathfrak{M} \models \mathsf{B}_k\neg\alpha_1$ or $\mathfrak{M} \models \mathsf{B}_k\neg\alpha_2$

(d) $\alpha = \neg\neg\alpha_1$ and $\mathfrak{M} \models \mathsf{B}_k\alpha_1$

(e) $k > 0$ and there exists $p \in \Phi$ such that both $\mathfrak{M} \models [p]\mathsf{B}_{k-1}\alpha$ and $\mathfrak{M} \models [\neg p]\mathsf{B}_{k-1}\alpha$

Rule (1) defines B as an implicit belief operator, which is easily seen to be characterized by logical omniscience. Suppose $\Gamma \models_{\mathbb{C}} \alpha$. If $\mathfrak{M} \models \mathsf{B}\gamma$ for every $\gamma \in \Gamma$, then each $w \in \Phi \to \mathbb{C}$ compatible with $\mathfrak{M}$ makes every element of Γ true, and so must make α true as well. Hence $\mathfrak{M} \models \mathsf{B}\alpha$.

The definition of $[\Subset]$ that we have seen before is repeated by rule (2). Rule (3) defines an operator for expansion by α, which could be compared to a public announcement of α in dynamic epistemic logic (van Ditmarsch et al., 2007). Rules (4) and (5) make connectives outside the scope of modal operators behave in their traditional ways.

The various parts of rule (6) define the infinite family of operators $\{\mathsf{B}_k : k \geq 0\}$. The sentence $\mathsf{B}_k\alpha$ can be read, as in $\mathcal{LL}$, as saying that α is in level k. Though we still have the operator $[\Subset]$, we will think of B_0 as indicating a form of explicit belief. Rule (6a) ensures that level 0 contains every α for which $[\Subset]\alpha$ is true. Rule (6b) allows for forming conjunctions from conjuncts that are separately believed. Note that this rule means that whether a sentence is in a level depends not just on what proposition it expresses, but also on its syntactic form. For example, that $((\alpha \vee \neg\beta) \wedge \beta)$ was in a level would not necessarily mean that $(\alpha \wedge \beta)$ was also. The rules (6c) and (6d) allow other simple ways of syntactically building up beliefs. Rule (6e) describes how the higher levels of belief are formed from the lower ones (the same way as in $\mathcal{LL}$). When k effort is allowed, reasoning by cases can be done, nested up to a depth of k.

3.4 Properties

In many ways, our logic behaves like $\mathcal{LL}$. However, the analogues in $\mathcal{LL}$ of our epistemic states are sets of clauses, which are a much less expressive class of objects. To make an epistemic state $\mathfrak{M}$ restricted in an analogous way, we would have to require that every $V \in \mathfrak{M}$ be finite (we typically would want that anyway) and, more seriously, that for each $v \in V$, exactly one atom is mapped to a non-N value by v. An obvious advantage of our more expressive semantic objects is that, unlike in $\mathcal{LL}$, we have

$$\mathsf{B}_0((p \wedge q) \vee (r \wedge s)) \not\models (\mathsf{B}_0(p \wedge q) \vee \mathsf{B}_0(r \wedge s))$$

since, for example, if $\mathfrak{M} = \{[\![((p \wedge q) \vee (r \wedge s))]\!]\}$ then $\mathfrak{M} \models ((p \wedge q) \vee (r \wedge s))$ but also $\mathfrak{M} \models (\neg\mathsf{B}_0(p \wedge q) \wedge \neg\mathsf{B}_0(r \wedge s)))$. Furthermore, because we define the satisfaction relation in a less syntactic way, we do not inherit $\mathcal{LL}$'s fiddliness over what is and isn't a clause. This is shown by, for example, how we have $\models \mathsf{B}_0(p \vee q) \supset \mathsf{B}_0(p \vee \neg\neg q)$, unlike in $\mathcal{LL}$.

In this section we prove various properties of our logic. Our first proposition clarifies the relationship between our logic and $\mathcal{LL}$:

Proposition 3. *Let S be a set of clauses, and let $\mathfrak{M} = \mathcal{H}(\{[\![c]\!] : c \in S\})$. If $S \models_{\mathcal{LL}} \mathsf{B}_k\alpha$ (where α contains only $\vee$ and $\neg$ operators), then $\mathfrak{M} \models \mathsf{B}_k\alpha$ (where the $\vee$ operators in α are expanded out in terms of $\wedge$).*

Proof sketch. The proof can be made by comparing the semantics of $\mathcal{LL}$ and our logic, which are rather similar. The only interesting point is this: Suppose that α is a clause, and there exists $c \in \mathcal{UP}(S)$ such that $c \subseteq \alpha$. By Proposition 2 we have that $[\![c]\!] \in \mathfrak{M}$. For each $v \in [\![c]\!]$, $v \models \alpha$, so $\mathfrak{M} \models [\Subset]\alpha$. □

Proposition 4. *Let $\Gamma \subseteq \mathscr{L}(\Phi)$ and suppose that $\mathfrak{M} = \mathcal{H}(\{[\![\gamma]\!] : \gamma \in \Gamma\})$. Then $\mathfrak{M} \models \mathsf{B}\alpha$ if and only if $\Gamma \models_{\mathbb{C}} \alpha$.*

Proof sketch. The "if" direction follows from logical omniscience and the fact that $\mathfrak{M} \models \mathsf{B}\gamma$ for each $\gamma \in \Gamma$. For the "only if" direction, note that each $w \in \Phi \to \mathbb{C}$ that makes every $\gamma \in \Gamma$ true is compatible with $\mathfrak{M}$, so if $\mathfrak{M} \models \mathsf{B}\alpha$, then each such w must satisfy α, so $\Gamma \models_{\mathbb{C}} \alpha$. □

Lemma 1. *If $\mathfrak{M} \models \mathsf{B}_k\alpha$ and $\mathfrak{M}'$ is a harmonized epistemic state such that $\mathfrak{M} \subseteq \mathfrak{M}'$, then $\mathfrak{M}' \models \mathsf{B}_k\alpha$.*

Proof idea. This can be shown by induction on k. □

Lemma 2 (monotonicity of expansions)**.** *Let $\alpha, \beta \in \mathscr{L}(\Phi)$. If $\mathfrak{M} \models \mathsf{B}_k\alpha$, then $\mathfrak{M}[\beta] \models \mathsf{B}_k\alpha$.*

Proof. $\mathfrak{M} \subseteq \mathfrak{M}[\beta]$. □

Proposition 5 (levels are cumulative)**.** $\models \mathsf{B}_k\alpha \supset \mathsf{B}_{k+1}\alpha$

Proof. Suppose $\mathfrak{M} \models \mathsf{B}_k\alpha$. Pick any $p \in \Phi$. By Lemma 2, $\mathfrak{M} \models [p]\mathsf{B}_k\alpha$ and $\mathfrak{M} \models [\neg p]\mathsf{B}_k\alpha$. □

Proposition 6 (level soundness)**.** $\models \mathsf{B}_k\alpha \supset \mathsf{B}\alpha$.

Proof idea. This can be shown by induction on k. □

Definition 13 (strictly finite)**.** An epistemic state $\mathfrak{M}$ is strictly finite if all of the following hold:

- $\mathfrak{M}$ is finite
- for every $V \in \mathfrak{M}$, V is finite
- for every $V \in \mathfrak{M}$, each $v \in V$ is finite.

Proposition 7 (eventual completeness)**.** *Suppose that $\mathfrak{M}$ is strictly finite. If $\mathfrak{M} \vDash \mathsf{B}\alpha$, then there is some k such that $\mathfrak{M} \vDash \mathsf{B}_k\alpha$.*

Proof. Suppose that $\mathfrak{M} \vDash \mathsf{B}\alpha$. Let $n = |\mathsf{at}(\mathfrak{M}) \cup \mathsf{at}(\alpha)|$ (since $\mathfrak{M}$ is strictly finite, n is finite). Let m be the number of atoms p such that either $[\![p]\!] \in \mathfrak{M}$ or $[\![\neg p]\!] \in \mathfrak{M}$. We will prove that $\mathfrak{M} \vDash \mathsf{B}_{n-m}\alpha$, by induction on $n - m$.

The base case, where $n - m = 0$, is straightforward. If there is some atom p such that both $[\![p]\!] \in \mathfrak{M}$ and $[\![\neg p]\!] \in \mathfrak{M}$, then because $\mathfrak{M}$ is harmonized, $\emptyset \in \mathfrak{M}$ and so $\mathfrak{M} \vDash \mathsf{B}_0\alpha$. Otherwise, let w be the truth assignment that makes p true iff $[\![p]\!] \in \mathfrak{M}$ and false iff $[\![\neg p]\!] \in \mathfrak{M}$. Note that if w is not compatible with $\mathfrak{M}$, then $\emptyset \in \mathfrak{M}$ (again, because $\mathfrak{M}$ is harmonized), and so $\mathfrak{M} \vDash \mathsf{B}_0\alpha$. Otherwise, $\mathfrak{M} \vDash \mathsf{B}\alpha$ iff $w \vDash_{\mathbb{C}} \alpha$. If $w \vDash_{\mathbb{C}} \alpha$, then using rules (6b-d) in the semantics it is possible to show that $\mathfrak{M} \vDash \mathsf{B}_0\alpha$.

For the inductive step, suppose that $n - m > 0$. Let $p \in \Phi$ be such that neither $[\![p]\!] \in \mathfrak{M}$ nor $[\![\neg p]\!] \in \mathfrak{M}$. Since $\mathfrak{M} \vDash \mathsf{B}\alpha$, it is also the case that $\mathfrak{M}[p] \vDash \mathsf{B}\alpha$ and $\mathfrak{M}[\neg p] \vDash \mathsf{B}\alpha$. Therefore, by the inductive hypothesis, $\mathfrak{M}[p] \vDash \mathsf{B}_{n-m-1}\alpha$ and $\mathfrak{M}[\neg p] \vDash \mathsf{B}_{n-m-1}\alpha$. Hence $\mathfrak{M} \vDash \mathsf{B}_{n-m}\alpha$ by rule (6e) in the semantics. □

Note that the proof of (Lakemeyer and Levesque, 2014, Theorem 3) is similar.

The next two observations hold for $\mathcal{LL}$ as well:

Observation 1. $\vDash \mathsf{B}_k(\alpha \wedge \beta) \equiv (\mathsf{B}_k\alpha \wedge \mathsf{B}_k\beta)$

Observation 2. $\vDash \mathsf{B}_k\neg\neg\alpha \equiv \mathsf{B}_k\alpha$

3.4.1 Associativity

We now will go about considering associativity within the scope of modal operators. Recall from our discussion of $\mathcal{LL}$ that $\mathsf{B}_k(\alpha \vee (\beta \vee \gamma)) \nvDash_{\mathcal{LL}} \mathsf{B}_k((\alpha \vee \beta) \vee \gamma)$. We will see that our logic works differently.

Lemma 3. $\vDash \mathsf{B}_k(\alpha \vee \beta) \supset \mathsf{B}_k(\alpha \vee (\beta \vee \gamma))$

Proof. Suppose that $\mathfrak{M} \vDash \mathsf{B}_k(\alpha \vee \beta)$, that is, that $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\alpha \wedge \neg\beta)$. We want to show that therefore $\mathfrak{M} \vDash \mathsf{B}_k(\alpha \vee (\beta \vee \gamma)$, that is, that $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\alpha \wedge \neg\neg(\neg\beta \wedge \neg\gamma))$. We prove this by induction on k.

If $k = 0$, there are two cases to consider.

- Rule (6a) applies, in that $\mathfrak{M} \vDash [\Subset](\alpha \vee \beta)$.

  Since $(\alpha \vee \beta) \vDash_{\mathbb{K}} (\alpha \vee (\beta \vee \gamma))$, $\mathfrak{M} \vDash [\Subset](\alpha \vee (\beta \vee \gamma))$, so $\mathfrak{M} \vDash \mathsf{B}_0(\alpha \vee (\beta \vee \gamma))$.

- Rule (6c) applies: either $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\alpha$ or $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\beta$.

  If $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\alpha$, then by rule (6c) $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\alpha \wedge \neg\neg(\neg\beta \wedge \neg\gamma))$.

  If $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\beta$, then by (6c) $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\beta \wedge \neg\gamma)$, by (6d) $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\neg(\neg\beta \wedge \neg\gamma)$, and by (6c) $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\alpha \wedge \neg\neg(\neg\beta \wedge \neg\gamma))$.

Now we want to show that $\vDash \mathsf{B}_{k+1}(\alpha \vee \beta) \supset \mathsf{B}_{k+1}(\alpha \vee (\beta \vee \gamma))$ given the inductive hypothesis that $\vDash \mathsf{B}_k(\alpha \vee \beta) \supset \mathsf{B}_k(\alpha \vee (\beta \vee \gamma))$. So, suppose $\mathfrak{M} \vDash \mathsf{B}_{k+1}(\alpha \vee \beta)$. There are again two cases. Again, rule (6c) might apply; this works as shown previously. The other case is that rule (6e) applies and there exists $p \in \Phi$ such that both $\mathfrak{M} \vDash [p]\mathsf{B}_k(\alpha \vee \beta)$ and $\mathfrak{M} \vDash [\neg p]\mathsf{B}_k(\alpha \vee \beta)$. By the inductive hypothesis, $\mathfrak{M} \vDash [p]\mathsf{B}_k(\alpha \vee (\beta \vee \gamma))$ and $\mathfrak{M} \vDash [\neg p]\mathsf{B}_k(\alpha \vee (\beta \vee \gamma))$, so $\mathfrak{M} \vDash \mathsf{B}_{k+1}(\alpha \vee (\beta \vee \gamma))$ by rule (6e). □

Note that Lemma 3 does not hold in $\mathcal{LL}$. Consider the setup $S = \{(p \vee q)\}$. Then

$$S \vDash_{\mathcal{LL}} B_0(p \vee q) \wedge \neg B_0(p \vee (q \vee \neg\neg r))).$$

Proposition 8. $\vDash \mathsf{B}_k((\alpha \vee \beta) \vee \gamma) \equiv \mathsf{B}_k(\alpha \vee (\beta \vee \gamma))$

Proof. We will show that $\vDash \mathsf{B}_k((\alpha \vee \beta) \vee \gamma) \supset \mathsf{B}_k(\alpha \vee (\beta \vee \gamma))$ (the other direction of implication works symmetrically). Note that $((\alpha \vee \beta) \vee \gamma) = \neg(\neg\neg(\neg\alpha \wedge \neg\beta) \wedge \neg\gamma)$ and $(\alpha \vee (\beta \vee \gamma)) = \neg(\neg\alpha \wedge \neg\neg(\neg\beta \wedge \neg\gamma))$. Suppose $\mathfrak{M}$ is such that $\mathfrak{M} \vDash \mathsf{B}_k((\alpha \vee \beta) \vee \gamma)$. The proof is by induction on k.

If $k = 0$ there are two cases to consider:

- Rule (6a) applies: $\mathfrak{M} \vDash [\Subset]((\alpha \vee \beta) \vee \gamma)$.

  Then, because $((\alpha \vee \beta) \vee \gamma) \vDash_{\mathbb{K}} (\alpha \vee (\beta \vee \gamma))$, $\mathfrak{M} \vDash [\Subset](\alpha \vee (\beta \vee \gamma))$ and so $\mathfrak{M} \vDash \mathsf{B}_0(\alpha \vee (\beta \vee \gamma))$.

- Rule (6c) applies: $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\neg(\neg\alpha \wedge \neg\beta)$ or $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\gamma$.

  If $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\neg(\neg\alpha\wedge\neg\beta)$, then by Observation 2, $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\alpha\wedge\neg\beta)$, i.e. $\mathfrak{M} \vDash \mathsf{B}_k(\alpha\vee\beta)$. Then by Lemma 3, $\mathfrak{M} \vDash \mathsf{B}_k(\alpha \vee (\beta \vee \gamma))$.

  If $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\gamma$, by (6c) we have that $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\beta \wedge \neg\gamma)$. Then by (6d) we have $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\neg(\neg\beta \wedge \neg\gamma)$, and finally by (6c) we have $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\alpha \wedge \neg\neg(\neg\beta \wedge \neg\gamma))$.

Now, for the inductive step, where $k > 0$. If rule (6c) applies this works as when $k = 0$. If rule (6e) applies then we can apply the inductive hypothesis to each of the split cases to get the desired result. □

That associativity also works for conjunctions can be demonstrated very simply by relying on Observation 1, as shown below (because Observation 1 is true for $\mathcal{LL}$, this result can be shown for $\mathcal{LL}$ as well).

Proposition 9. $\vDash \mathsf{B}_k((\alpha \wedge \beta) \wedge \gamma) \equiv \mathsf{B}_k(\alpha \wedge (\beta \wedge \gamma))$

Proof. We will show that $\vDash \mathsf{B}_k((\alpha \wedge \beta) \wedge \gamma) \supset \mathsf{B}_k(\alpha \wedge (\beta \wedge \gamma))$ (the other direction of implication works symmetrically). Suppose $\mathfrak{M} \vDash \mathsf{B}_k((\alpha \wedge \beta) \wedge \gamma)$. By applying Observation 1 (twice), we see that $\mathfrak{M} \vDash \mathsf{B}_k\gamma$, $\mathfrak{M} \vDash \mathsf{B}_k\alpha$, and $\mathfrak{M} \vDash \mathsf{B}_k\beta$. Then by using rule (6b) twice we get that $\mathfrak{M} \vDash \mathsf{B}_k(\alpha \wedge (\beta \wedge \gamma))$. □

3.4.2 Distributivity

Now we will consider whether we get distributivity of conjunction over disjunction and vice versa within belief modalities. As we will see, with our logic, each of the distribution rules work in only one direction. This is unsurprising, since that was also shown to be the case for the similar logic $\mathcal{SL}$ (see Liu, 2006, Section 5.3.3).

Proposition 10. $\vDash \mathsf{B}_k((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \supset \mathsf{B}_k(\alpha \wedge (\beta \vee \gamma))$

Proof. Suppose that $\mathfrak{M} \vDash \mathsf{B}_k((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$, that is, that $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg(\alpha \wedge \beta) \wedge \neg(\alpha \wedge \gamma))$. We will show that $\mathsf{B}_k(\alpha \wedge (\beta \vee \gamma))$, that is, that $\mathfrak{M} \vDash \mathsf{B}_k(\alpha \wedge \neg(\neg\beta \wedge \neg\gamma))$. The proof is by induction on k.

If $k = 0$, then there are two cases to consider:

- Rule (6a) applies: $\mathfrak{M} \vDash [\Subset]((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$.

  Since $((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \vDash_{\mathbb{K}} (\alpha \wedge (\beta \vee \gamma))$, $\mathfrak{M} \vDash [\Subset](\alpha \wedge (\beta \vee \gamma))$, so $\mathfrak{M} \vDash \mathsf{B}_0(\alpha \wedge (\beta \vee \gamma))$.

- Rule (6c) applies: $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg(\alpha \wedge \beta)$ or $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg(\alpha \wedge \gamma)$.

  If $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg(\alpha \wedge \beta)$, then by Observation 2 and Observation 1, $\mathfrak{M} \vDash \mathsf{B}_k\alpha$ and $\mathfrak{M} \vDash \mathsf{B}_k\beta$. Using rules (6b-d) it can be shown that $\mathfrak{M} \vDash \mathsf{B}_k(\alpha \wedge \neg(\neg\beta \wedge \neg\gamma))$. The case where $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg(\alpha \wedge \gamma)$ is similar.

For the inductive step, where $k > 0$, there are again two cases. If rule (6c) applies, the situation works as shown previously. If (6e) applies, by using the inductive hypothesis with each of the split cases and then applying rule (6e) we get the desired result. □

The converse of Proposition 10 does not hold in general. For example, if $\mathfrak{M} = \{[\![p]\!], [\![(q \vee r)]\!]\}$, then $\mathfrak{M} \vDash \mathsf{B}_0(p \wedge (q \vee r))$ but $\mathfrak{M} \nvDash \mathsf{B}_0((p \wedge q) \vee (p \wedge r))$.

Proposition 11. $\vDash \mathsf{B}_k(\alpha \vee (\beta \wedge \gamma)) \supset \mathsf{B}_k((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$

Proof. Suppose that $\mathfrak{M}$ is such that $\mathfrak{M} \vDash \mathsf{B}_k(\alpha \vee (\beta \wedge \gamma))$, that is, that $\mathfrak{M} \vDash \mathsf{B}_k\neg(\neg\alpha \wedge \neg(\beta \wedge \gamma))$. We will show that therefore $\mathfrak{M} \vDash \mathsf{B}_k((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$, i.e. that $\mathfrak{M} \vDash \mathsf{B}_k(\neg(\neg\alpha \wedge \neg\beta) \wedge \neg(\neg\alpha \wedge \neg\gamma))$. The proof is by induction on k.

If $k = 0$, then there are two cases to consider:

- Rule (6a) applies: $\mathfrak{M} \vDash [\Subset](\alpha \vee (\beta \wedge \gamma))$.

  Then, because $(\alpha \vee (\beta \wedge \gamma)) \vDash_{\mathbb{K}} ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$, $\mathfrak{M} \vDash \mathsf{B}_0((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$.

- Rule (6c) applies: $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg\alpha$ or $\mathfrak{M} \vDash \mathsf{B}_k\neg\neg(\beta \wedge \gamma)$.

  In either case, it is a straightforward task to show that $\mathfrak{M} \vDash \mathsf{B}_k((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$.

The inductive step is also straightforward and similar to in previous proofs that we have shown. □

Note that, as with the converse of Proposition 10, the converse of Proposition 11 does not hold. For example, if $\mathfrak{M} = \{[\![p \vee q]\!], [\![p \vee r]\!]\}$, then $\mathfrak{M} \models \mathsf{B}_0((p \vee q) \wedge (p \vee r))$ but $\mathfrak{M} \not\models \mathsf{B}_0(p \vee (q \wedge r))$.

3.5 A reasoning service

Our logic can be used to specify a reasoning service in the following way: after being told the sequence of sentences $\alpha_1, \alpha_2, \ldots, \alpha_n$ (and nothing else), the service will, given an input sentence β and integer k, return "Yes" if

$$\emptyset[\alpha_1][\alpha_2]\cdots[\alpha_n] \models \mathsf{B}_k\beta,$$

and "No" otherwise. Equivalently (because of Lemma 1), we could phrase the question as determining whether

$$\models [\alpha_1][\alpha_2]\cdots[\alpha_n]\mathsf{B}_k\beta.$$

Of course, it would be possible to define all sorts of other reasoning services, e.g. the service that, given $\alpha_1, \ldots, \alpha_n, \beta$, finds the least k (if it exists) for which $\models [\alpha_1][\alpha_2]\cdots[\alpha_n]\mathsf{B}_k\beta$. However, we will restrict our attention to the first one.

3.5.1 Complexity

We now want to consider the computational complexity of the reasoning service we have described, in other words, the complexity of deciding the validity of sentences that are of the form $[\alpha_1][\alpha_2]\cdots[\alpha_n]\mathsf{B}_k\beta$. Note that we will not be addressing the more general question of the complexity of deciding whether arbitrary sentences of the logic are valid.

Note that sometimes the harmonization of a set may be exponentially larger than the set itself. For example, consider the family of sets $\{\mathfrak{M}_i : i \in \{0, 1, 2, \ldots\}\}$ where

$$\mathfrak{M}_n = \{[\![(p_1 \vee p_2 \vee \cdots \vee p_n \vee q)]\!], [\![\neg p_1]\!], [\![\neg p_2]\!], \ldots [\![\neg p_n]\!]\}$$

Then $\mathcal{H}(\mathfrak{M}_n)$ is exponentially larger than $\mathfrak{M}_n$, since $[\![(\bigvee \Gamma) \vee q]\!] \in \mathcal{H}(\mathfrak{M}_n)$ for each $\Gamma \subseteq \{p_i : 1 \leq i \leq n\}$. Obviously, for our purposes there is not normally a need to explicitly compute all those elements, as the following observation shows:

Proposition 12. $\mathfrak{M} \models \mathsf{B}_k\alpha$ *iff* $\min_{\subset}(\mathfrak{M}) \models \mathsf{B}_k\alpha$.

Proof sketch. For the "if" direction, note that since $\min_{\subset}(\mathfrak{M}) \subseteq \mathfrak{M}$, if $\min_{\subset} \mathfrak{M} \models \mathsf{B}_k\alpha$ then $\mathfrak{M} \models \mathsf{B}_k\alpha$ (by Lemma 1). The "only if" direction can be shown by induction and noting that $\mathfrak{M} \models [\Subset]\alpha$, then $\min_{\subset}(\mathfrak{M}) \models [\Subset]\alpha$. □

Corollary 1. *If* $\mathfrak{M}'$ *is a harmonized epistemic state such that* $\min_{\subset}(\mathfrak{M}) \subseteq \mathfrak{M}' \subseteq \mathfrak{M}$*, then* $\mathfrak{M} \models \mathsf{B}_k\alpha$ *iff* $\mathfrak{M}' \models \mathsf{B}_k\alpha$

Proof. If $\mathfrak{M} \vDash \mathsf{B}_k\alpha$, then by Proposition 12 $\min_{\subset}(\mathfrak{M}) \vDash \mathsf{B}_k\alpha$, and then, by Lemma 1, $\mathfrak{M}' \vDash \mathsf{B}_k\alpha$. On the other hand, if $\mathfrak{M}' \vDash \mathsf{B}_k\alpha$, then by Lemma 1, $\mathfrak{M} \vDash \mathsf{B}_k\alpha$. □

Now, the following algorithm $\mathtt{h}$ is such that $\min_{\subset} \mathcal{H}(\mathfrak{M}) \subseteq \mathtt{h}(\mathfrak{M}) \subseteq \mathcal{H}(\mathfrak{M})$:

```
function h(𝔐):
  for each singleton {v} ∈ 𝔐:
    for U ∈ 𝔐:
      if U ≠ {u ∈ U : u ♢ v}:
        return h({{w ∈ W : w ♢ v} : W ∈ 𝔐})
  return 𝔐
```

That $\min_{\subset} \mathcal{H}(\mathfrak{M}) \subseteq \mathtt{h}(\mathfrak{M}) \subseteq \mathcal{H}(\mathfrak{M})$ can be seen from how the function $\mathtt{h}$ is basically just repeatedly applying the rule that harmonization is based on – if $W \in \mathfrak{M}$ and $\{v\} \in \mathfrak{M}$, then $\{u \in W : u \diamond v\}$ is added – to all elements $W \in \mathfrak{M}$, except that each time the rule is applied some elements that are supersets of others may be discarded. Note that $\mathtt{h}(\mathfrak{M})$ is always harmonized:

Lemma 4. $\mathtt{h}(\mathfrak{M}) = \mathcal{H}(\mathtt{h}(\mathfrak{M}))$

Proof. Let us name as $\mathfrak{M}_{final}$ the parameter passed to the final recursive call in $\mathtt{h}$. Since this is the last call, it must be the case that there are no $U \in \mathfrak{M}_{final}$ and $\{v\} \in \mathfrak{M}_{final}$ for which $U \neq \{u \in U : u \diamond v\}$. This means that $\mathfrak{M}_{final}$ is harmonized. □

Lemma 5. $\mathtt{h}$ *runs in polynomial time, on a strictly finite input.*

Proof sketch. In each new recursive call the input size has always been strictly reduced, so there will not be more recursive calls than the combined size of all elements of $\mathfrak{M}$. Furthermore, it is clear that within each call, only a polynomial amount of work is done. □

Now, the (rather boring) algorithm $\mathtt{Q}$ below computes whether $\mathfrak{M} \vDash \mathsf{B}_k\alpha$, given $\mathfrak{M}$, α, and k.

```
function Q(𝔐, α, k):
  if α = ¬¬α₁:
    return Q(𝔐, α₁, k)

  if k = 0:
    for V ∈ 𝔐:
      if v ⊨_𝕂 α for all v ∈ V:
        return ⊤
  if α = (α₁ ∧ α₂):
    if Q(𝔐, α₁, k) and Q(𝔐, α₂, k):
```

```
    return ⊤
if α = ¬(α₁ ∧ α₂):
  if Q(𝔐, ¬α₁, k) or Q(𝔐, ¬α₂, k):
    return ⊤
if k > 0:
  for p ∈ at(𝔐) ∪ at(α):
    if Q(h(𝔐 ∪ {⟦p⟧}), α, k − 1) and Q(h(𝔐 ∪ {⟦¬p⟧}), α, k − 1):
      return ⊤
return ⊥
```

Proposition 13 (correctness)**.** *Suppose $\mathfrak{M}$ is a strictly finite harmonized epistemic state. Then $\mathfrak{M} \models \mathsf{B}_k\alpha$ if and only if $\mathtt{Q}(\mathfrak{M}, \alpha, k) = \mathsf{T}$.*

Given how much the algorithm looks like the semantics, this proposition is not very surprising. The only point worth remarking upon is how the only atoms that need to be used in the splitting rule are those mentioned by either the epistemic state or α.

Proposition 14 (complexity)**.** *For constant k and strictly finite $\mathfrak{M}$, $\mathtt{Q}(\mathfrak{M}, \alpha, k)$ can be computed in polynomial time.*

Proof idea. This can be seen by inspection and induction. □

As we said previously, for our reasoning service we will want to determine whether $\models [\alpha_1][\alpha_2]\cdots[\alpha_n]\mathsf{B}_k\beta$. To do so, we can execute

$$\mathtt{decide}(\{\alpha_1, \alpha_2, \ldots, \alpha_n\}, \beta, k),$$

where the `decide` function is constructed as shown below:

```
function decide(Γ, k, β)
  return decide_helper({⟦γ⟧ : γ ∈ Γ}, β, k)

function decide_helper(𝔐, β, k)
  return Q(h(𝔐), β, k)
```

It follows from Lemma 5 and Proposition 14 that, when k is treated as a fixed constant, `decide_helper`$(\mathfrak{M}, \beta, k)$ can be computed in polynomial time. Therefore, for `decide`(Γ, β, k) to run in polynomial time for constant k it would suffice that, for each $\gamma \in \Gamma$, $[\![\gamma]\!]$ can be computed in polynomial time.

Unfortunately, this is not the case in general. For example, if a sentence γ is in conjunctive normal form (i.e. is a conjunction of clauses, where a clause is a disjunction of literals) then $[\![\gamma]\!]$ may be exponentially larger than γ. However, if γ is in disjunctive normal form (i.e. is

a disjunction of conjunction of literals), then $\|\gamma\|$ can be computed in polynomial time, as shown by the following observation:

Observation 3. *Suppose that*

$$\gamma = \bigvee_{1 \leq i \leq n} \bigwedge_{1 \leq j \leq m_i} \ell_{ij}$$

for some integers $n, m_1, \ldots, m_n$, where each ℓ_{ij} is some literal. Then

$$\|\gamma\| = \min_{\subset}(\{\ell_{ij} : 1 \leq j \leq m_i\} : 1 \leq i \leq n \text{ and } \ell_{ij} \neq \overline{\ell}_{ik} \text{ for any } j, k \in \{1, \ldots, m_i\}\}).$$

So, in conclusion, we have the following result:

Proposition 15. *For $\alpha_1, \ldots, \alpha_n$ in disjunctive normal form, any sentence β, and k a constant, whether $\vDash [\alpha_1][\alpha_2] \cdots [\alpha_n]\mathsf{B}_k\beta$ can be computed in polynomial time.*

Requiring $\alpha_1, \ldots, \alpha_n$ to be in disjunctive normal form may seem like a serious constraint, since converting a sentence into that form may take exponential time. However, for knowledge representation purposes, we may often be dealing with large collections of facts which are individually simple – i.e. the number n of sentences may grow to be very large, but each sentence typically remains small. In such a case it could be practical to convert each of the sentences into disjunctive normal form. If we have a knowledge base KB which is structured as a large conjunction of facts, then the idea would be to break apart the conjunction and convert each conjunct separately, rather than converting the entire conjunction.

4 Evaluating effort

Thus far, we have spoken of effort mostly without reference to concrete problems, but only as an abstraction. We will now take a small step towards rectifying this matter by considering the difficulty of solving Sudoku puzzles. A Sudoku board is a 9x9 grid of cells, where cells are grouped into nine 3x3 regions. Each cell can contain a number from $N = \{1, 2, 3, \ldots, 9\}$. In a starting configuration, some of the cells have numbers given in them, while the rest are empty. The goal is to write a number chosen from N into each of the initially empty cells in such a way that each row, column, and region contains all distinct numbers. The initial configuration is required to be such that there is a unique way of filling in the board that meets the goal. Figure 1 shows one possible starting configuration.

Sudoku has some disadvantages as an example problem. As a puzzle game it may be, even at its easiest, harder (for people) than a lot of commonsense reasoning. Unfortunately, for most commonsense reasoning it's very unclear how to delimit the inputs and background knowledge involved. So we will consider Sudoku; as it turns out, even from it, we will be able to get some ideas regarding possible issues with our logic.

			7					
1								
			4	3		2		
								6
			5		9			
						4	1	8
				8	1			
		2					5	
	4					3		

Figure 1: A Sudoku game (from Royle (2012)) with 17 given numbers

Lynce and Ouaknine (2006) considered Sudoku as a SAT problem. Using a large set of puzzles that each had 17 given numbers[1], they found that with their "extended" encoding (which contained some logically redundant clauses), about half of the puzzles could be solved using unit propagation alone, and all of them could be solved if closure under both unit propagation and the "failed literal rule" was used. (The failed literal rule allows a literal ℓ to be concluded if assuming $\overline{\ell}$ and doing unit propagation reveals a contradiction.) Henz and Truong (2009) showed, using the same extended encoding as Lynce and Ouaknine, that there were some puzzles (with more than 17 given numbers) which could not be solved using just unit propagation and the failed literal rule.

So, if we encode Sudoku puzzles as Lynce and Ouaknine did, then some (but not all) of them can be solved with unit propagation. By Proposition 2, if we create an epistemic state corresponding to the encoding of a puzzle that can be solved with unit propagation, then each move in the solution to the puzzle will be at level 0 in our logic. Perhaps that shows that we are not discriminating effort in a fine-grained enough way. An easy Sudoku puzzle is still a puzzle, which takes a notable amount of time for a person to complete. So even the lowest level of belief may require some "puzzle mode" (Levesque, 1988) reasoning.

There is a caveat here, in that how many puzzles can be solved with unit propagation depends on exactly how we encode a puzzle. Lynce and Ouaknine did also have a "minimal" encoding for Sudoku with which unit propagation was unable to solve any puzzles. However, Lynce and Ouaknine's extended encoding differs from the minimal encoding only by adding

[1] 17 is the minimal number of given numbers required for a puzzle to have a unique solutions, according to McGuire et al. (2012), though this had not yet been proven in 2006.

clauses saying things that any human Sudoku player would find obvious, like that each cell contains at most one number, or that each number appears at least once in a row (these facts are of course derivable from the minimal encoding, but not by unit propagation). So it is not clear how to encode puzzles in a way such that obvious properties of them can be derived, but not their solutions.

The problem for us with Sudoku may be that unit propagation is in some ways too powerful. In the next section, we suggest this is because it requires agents to remember too many things.

4.1 Memory

If you try to solve a Sudoku puzzle without writing *anything* down – i.e. without filling in the numbers you have worked out so far – it probably will seem much more difficult. Similarly, that the given numbers in the initial state remain before your eyes the entire time you're working on the puzzle is helpful. That Sudoku is solved on paper obscures some of the ways in which memory is required in reasoning.

Among measures of effort, proof length could be considered a baseline. We would like for anything fancier to show some advantage. But it seems reasonable to predict that the moves that could easily be derived in Sudoku *without writing anything down* would be those that had a short proof. To get closure under unit propagation, or closure under most anything else, requires you to remember what you have concluded so far so that you can build upon it.

Another important thing about memory and Sudoku is that it is easy to remember the rules, which can be described in a single paragraph. However, expressing the rules as propositional clauses in an obvious way seems to obscure their structure – we would never present the rules to a human by listing thousands of constraints on individual cells. Given such a list, a human would have to sift through it to discover the patterns that a normal description would make obvious. This would not be easy at all, unless perhaps we cheated by arranging the list in a systematic way, so that the ordering itself provides information beyond the contents. If the order is random, e.g. like

> Cell $\langle 4, 7\rangle$ can either have a 2 or a 3 or a 9 or a 6 or a 5 or a 4 or a 7 or an 8 or a 1, and cell $\langle 3, 9\rangle$ can either have a 4 or a 6 or a ...

then it's very hard to make sense of them. Without discovering the features of the usual descriptions, but just having a jumble of thousands of constraints, a person (even a logician!) would probably not be able to play very well at all. The person might have to consult a written copy of the rules constantly to remember which numbers are allowed to go where. Even if the human can memorize all the rules, will those memories be appropriately indexed so that relevant rules can be efficiently retrieved when needed?

For illustration, it may help to consider taking a set of propositional clauses expressing the constraints of the game, and perturbing them in idiosyncratic ways. For example, you might make it so that row three is allowed to have two 7's, column six may or may not have a 9, and various other things of that sort, carefully chosen so as to ensure the constraints overall are still satisfiable. Let us call the resulting game Corrupted Sudoku. The rules of Corrupted Sudoku probably could not be summarized as concisely as Sudoku's, even though their propositional representation could be about the same size. Consequently, a human might find it hard to play Corrupted Sudoku merely because it's hard to remember the rules. Even if our system of levels of belief accurately described how difficult people find Sudoku, the system would predict that Corrupted Sudoku should be the same.

Most if not all popular puzzle games have very simple first-order descriptions, probably much smaller than any propositional representation of them. Therefore, looking at such games to determine how our notion of effort in propositional logic correlates with human difficulty may be systematically misleading.

Another thing that human players know about Sukoku is that a puzzle can be solved by just considering the puzzle itself – most world knowledge is irrelevant. A player may split cases to solve a puzzle, but they won't bother with splitting completely irrelevant cases, like "either the conservatives will win the next election, or they won't". The way our levels of belief are designed, how difficult it is to compute what is at a given level depends on the size of the whole knowledge base. So, having a large amount of political knowledge, for example, could make an agent much slower at Sudoku, which is undesirable.

5 Extensions

We now consider various, rather speculative, ways in which our logic from section 3 might be extended or modified.

5.1 A parameterized logic

The logic we have described in section 3 incorporates a number of features that may seem somewhat arbitrary. We might wonder about alternatives to harmonization (especially given our criticism of unit propagation in the last section), or if the rather syntactic rules (6b-d) in the semantics should be replaced by other syntactic rules. To aid reflection on this, we will construct a family of logics defined by three parameters $\langle \mathsf{hyp}, \mathsf{d}, \mathsf{R} \rangle$, so as to highlight possible ways our logic could be modified.

The parameters are explained below:

- hyp is a function from epistemic states and literals to epistemic states – the idea is that $\mathsf{hyp}(\mathfrak{M}, \ell)$ is the result of an agent with state $\mathfrak{M}$ hypothesizing that ℓ is true, as with the $\mathfrak{M}[\ell]$ function from our logic in section 3.

- d is some function from trees (in the graph theory sense) to some measure of their size.
- R is a set of proof rules.

Given a set of proof rules R and set of sentences Γ, let us write $\mathsf{C}_\mathsf{R}(\Gamma)$ to denote the closure of Γ under R, i.e the superset of Γ that includes all sentences that can be derived from Γ using the rules.

Definition 14. Λ_{hyp} is the set of all rooted trees whose nodes are epistemic states and are such that for any node $\mathfrak{M}$ in the tree, if $\mathfrak{M}'$ is a child of $\mathfrak{M}$ then $\mathfrak{M}' = \mathsf{hyp}(\mathfrak{M}, \ell)$ for some literal ℓ.

Definition 15. The relation $\models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} \subseteq \Lambda_{\mathsf{hyp}} \times \mathscr{L}(\Phi)$ is the least relation such that all of the following hold:

- $T \models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} \alpha$, where T has root $\mathfrak{M}$, if there is some $V \in \mathfrak{M}$ such that $v \vDash_{\mathbb{K}} \alpha$ for all $v \in V$
- $T \models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} \alpha$, where T has root $\mathfrak{M}$, if $\mathfrak{M}$ has children $\mathfrak{M}_1 = \mathsf{hyp}(\mathfrak{M}, \ell)$ and $\mathfrak{M}_2 = \mathsf{hyp}(\mathfrak{M}, \overline{\ell})$, which are the roots of trees T_1 and T_2 respectively, and $T_1 \models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} \alpha$ and $T_2 \models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} \alpha$
- $\models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} = \Big\{\langle T, \alpha\rangle : T \in \Lambda_{\mathsf{hyp}} \text{ and } \alpha \in \mathsf{C}_\mathsf{R}\Big(\Big\{\beta \in \mathscr{L}(\Phi) : T \models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} \beta\Big\}\Big)\Big\}$

Note that if two relations satisfy these three properties, then so does their intersection, so there does always exist a least relation satisfying the properties.

Definition 16. The relation $\models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}}$ between epistemic states and modal formulas is defined by

- $\mathfrak{M} \models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}} \mathsf{B}_k\alpha$ if there exists $T \in \Lambda_{\mathsf{hyp}}$ with root $\mathfrak{M}$ such that $\mathsf{d}(T) = k$ and $T \models\!\!\!\models_{\mathsf{R},\mathsf{hyp}} \alpha$
- $\mathfrak{M} \models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}} (\varphi \wedge \psi)$ if $\mathfrak{M} \models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}} \varphi$ and $\mathfrak{M} \models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}} \psi$
- $\mathfrak{M} \models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}} \neg\varphi$ if not $\mathfrak{M} \models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}} \varphi$

With the right choice of parameters in $\models\!\!\!\models^{\mathsf{d}}_{\mathsf{R},\mathsf{hyp}}$, we can more or less emulate our logic from section 3: let $\mathsf{hyp}(\mathfrak{M}, \ell) = \mathfrak{M}[\ell] = \mathcal{H}(\mathfrak{M} \cup \{[\![\ell]\!]\})$, let d be depth, and let R consist of proof rules corresponding to rules (6b-d) in our semantics, i.e. introduction rules for conjunctions, negated conjunction, and double negation. (Note that in section 3 we also restricted the satisfaction relation to harmonized epistemic states only.)

Clearly, we could use other measures of the size of a tree, other proof rules, and could replace the use of harmonization in hyp with something else. The choices made have historical basis in Lakemeyer and Levesque (2014), but have not been subject to much scrutiny. We

could, for instance, replace harmonization with this strictly stronger closure condition: if $U \in \mathfrak{M}$ and $V \in \mathfrak{M}$, then $\{u \in U : u \circledcirc \bigcap V\} \in \mathfrak{M}$. Of course, $\bigcap V$ is the intersection of the truth assignments in V, and if $V = \{v\}$, then $\bigcap V = v$.

However, given that we have argued that even unit propagation is too strong, a better direction of change might be to make hyp capture *fewer* inferences. We could set $\mathsf{hyp}(\mathfrak{M}, \ell)$ to simply be the function

$$z(\mathfrak{M}, \ell) := \{\{v \in V : v \nVdash_{\mathbb{K}} \neg\ell\} : V \in \mathfrak{M}\} \cup \{[\![\ell]\!]\}.$$

The informal interpretation of this is that hypothesizing ℓ just involves adding a belief in ℓ and removing beliefs that contradict ℓ. Note that if we had used $z(\mathfrak{M}, \ell)$ instead of $\mathfrak{M}[\ell]$ we would still have been able to prove eventual completeness. Also, using z might make the depth of case-splitting more correlated with human memory use (because the memory requirements of harmonization would be eliminated).

5.1.1 On the bushiness of trees

Let us turn to thinking of alternatives to depth. Sometimes, when cases are split, one of the cases is very easy to refute. Unit propagation can be thought of as case-splitting in which one of the cases can be immediately thrown away. Only considering the depth of case-splitting may not give a very accurate idea of how much effort is really being spent, because among trees of a given depth, some have many more nodes than others.

An alternative measure to depth is the Horton-Strahler number, which has been described as "a measure of 'bushiness' for trees" (Helmert et al., 2014, p. 39), and which has been used in analyzing the difficulty of SAT instances (see Ansótegui et al. (2008)).

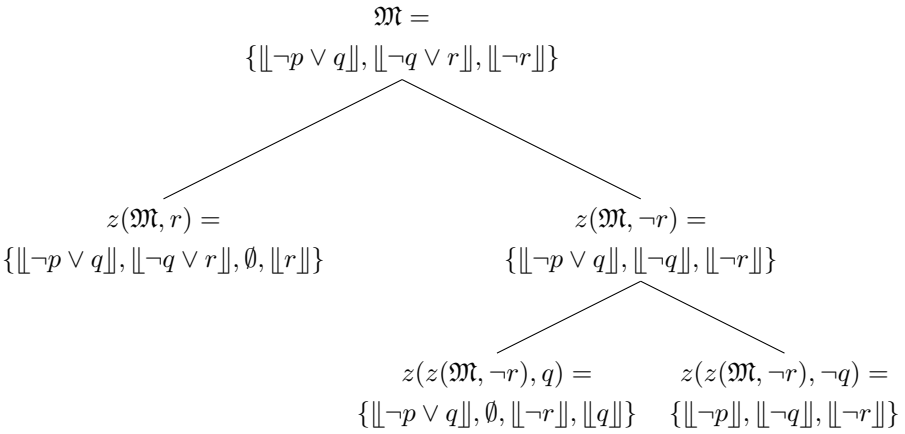
Definition 17 (Horton-Strahler number, see (Ansótegui et al., 2008, Definition 1))**.** The Horton-Strahler number of a case-splitting tree T, written $\mathsf{hs}(T)$, is define recursively as follows:

- If T has only one node, $\mathsf{hs}(T) = 0$.
- If T's root node has two children, let T_1 and T_2 be the subtrees rooted by those children. If $\mathsf{hs}(T_1) = \mathsf{hs}(T_2)$, then $\mathsf{hs}(T) = \mathsf{hs}(T_1) + 1$. Otherwise, $\mathsf{hs}(T) = \max(\mathsf{hs}(T_1), \mathsf{hs}(T_2))$.

There is an interesting relation between Horton-Strahler numbers and unit propagation (see (Ansótegui et al., 2008, Lemma 4)). It can be seen that if $\mathfrak{M} = \{[\![c]\!] : c \in \Gamma\}$ for some set of clauses Γ, then for any $c \in \mathcal{UP}(\Gamma)$ it is the case that $\mathfrak{M} \overset{\mathsf{hs}}{\underset{\emptyset, z}{\models}} \mathsf{B}_1 c$. So using Horton-Strahler numbers are an elegant way to incorporate some ability to do unit propagation without including in the semantics any reference to harmonization. An example of this in action follows:

Example 1. Let $\Gamma = \{(\neg p \vee q), (\neg q \vee r), \neg r\}$ and $\mathfrak{M} = \{[\![c]\!] : c \in \Gamma\}$. Note that $\neg p$ and $\neg q$ can be derived from Γ by unit propagation.

The tree from Λ_z shown below has a Horton-Strahler number of 1, and its root is $\mathfrak{M}$.



It can be seen from this tree that $\mathfrak{M} \mid\!\overset{\mathsf{hs}}{\underset{\emptyset,z}{=\!=}} \mathsf{B}_1 \neg p$ and $\mathfrak{M} \mid\!\overset{\mathsf{hs}}{\underset{\emptyset,z}{=\!=}} \mathsf{B}_1 \neg q$.

5.2 Introspection

Our logic does not feature any introspection, but as we said in the introduction, we expect that incorporating that would have interesting applications. For now, we will just note a problem with the introspection featured in the logic $\mathcal{LB}$ from (Lakemeyer and Levesque, 2013) (aside from introspection, $\mathcal{LB}$ was otherwise similar to $\mathcal{SL}$). According to (Lakemeyer and Levesque, 2013, Proposition 2), the following is valid in $\mathcal{LB}$, for any nonnegative integers k and j:

$$\mathsf{B}_k \alpha \supset \mathsf{B}_j \mathsf{B}_k \alpha$$

Note that this means that determining what sentences are in any level, even level 0, cannot be easier than determining what sentences are in level k, for arbitrarily high k. If, as in (Lakemeyer and Levesque, 2013), mere decidability rather than tractability is the main concern, this may not cause a problem, but clearly for our purposes we would have to do something different.

5.3 Other possibilities

Parikh (1987) defined a *knowledge algorithm* as consisting of a database and a procedure that, given an input question and a resource bound, works up to the bound, and then either answers the question or says "I don't know". Also, in a feature that might be interesting to extend our logic with, the database may be updated as a result of the query. This could be used to model Socratic questioning, where a series of well-chosen questions make the agent

realize what it (implicitly) knew all along (see Crawford and Kuipers (1989) for an existing approach to formalizing this).

Parikh also suggested that, in some cases, the agent may know that an implicit belief does not exist. McCarthy (1977) gave the example of being sure that you will not be able to, by reasoning alone, determine whether the president is currently standing. Our levels of belief can be thought of as approximations of implicit belief from below; it would be interesting to have approximations from above, that would identify sentences that were obviously neither believed nor disbelieved. See (Schaerf and Cadoli, 1995) and (Finger and Wassermann, 2007) for existing approaches at this.

Finally, let us note that one of the advantages some authors have found with neighborhood semantics is that agents can have conflicting beliefs without believing everything. Unfortunately, our eventual completeness result means that in our logic agents can ultimately derive anything from contradictory beliefs. Some further mechanism would be needed to prevent this.

6 Conclusion

In the introduction, we discussed (Reiter, 2000)'s idealized model of narratives. It's interesting to note that Reiter implemented a query evaluator for narratives that used an incomplete theorem prover using unit propagation and a form of case-splitting. However, for Reiter these seemed to be merely ad hoc implementation details. In contrast, the position of this paper is that the limitations of reasoning should be incorporated into the *theory* of intelligent reasoning.

As we have shown, the approach we have taken, in incorporating neighborhood semantics, has some advantages over the preceding work (Liu et al., 2004; Liu, 2006; Lakemeyer and Levesque, 2013, 2014) it is based on. By not relying on sets of clauses as semantic objects, our logic avoids being so sensitive to minor syntactic variations. We retain the hierarchy of levels of increasing inferential power, which can be used to define a (sometimes) tractable reasoning service.

Ultimately, though, our approach remains rather disconnected from human reasoning. We still measure effort in the same way as preceding work, and a reflection on the difficulty of Sudoku puzzles suggests that this way does not capture all the constraints that people operate under. We have made some suggestions for ways in which our logic might be modified, but how to address this issue effectively remains unclear. For future work, we suggest that a more psychological turn should be taken, and that evaluations be made based on comparisons with human performance.

A direction in which our work could be extended, which may appeal also to AI researchers who have less interest in cognitive science, would be to develop notions of effort that could

model various sorts of abstract resource bounds (e.g. time or certain types of memory). Artificial agents may have resource bounds that are quantitatively very different from people's – for example, an artificial agent may not have such a drastically limited working memory capacity – but they still cannot act with logical omniscience, except in very simple circumstances. By specifying precisely how reasoning is limited, possibilities may arise to make proofs and give guarantees about behavior in more realistic circumstances.

References

C. Ansótegui, M. L. Bonet, J. Levy, and F. Manyà. Measuring the Hardness of SAT Instances. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, pages 222–228, 2008.

C. Areces and D. Figueira. Which Semantics for Neighbourhood Semantics? In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 671–676, 2009.

R. A. Brooks. Elephants Don't Play Chess. *Robotics and Autonomous Systems*, 6(1–2): 3–15, 1990.

H. Castañeda. Review: Jaakko Hintikka, Knowledge and Belief. An Introduction to the Logic of the Two Notions. *Journal of Symbolic Logic*, 29(3):132–134, 1964.

J. M. Crawford and D. W. Etherington. A Non-Deterministic Semantics for Tractable Inference. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI 1998)*, pages 286–291, 1998.

J. M. Crawford and B. Kuipers. Towards a Theory of Access-limited Logic for Knowledge Representation. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR 1989)*, pages 67–78, 1989.

J. M. Crawford and B. Kuipers. Negation and Proof by Contradiction in Access-Limited Logic. In *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI 1991)*, pages 897–903, 1991.

M. D'Agostino and L. Floridi. The enduring scandal of deduction. *Synthese*, 167(2):271–315, 2009.

M. D'Agostino, M. Finger, and D. Gabbay. Semantics and proof-theory of depth bounded Boolean logics. *Theoretical Computer Science*, 480(0):43–68, 2013.

M. Dalal. Semantics of an Anytime Family of Reasoners. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI 1996)*, pages 360–364, 1996.

H. N. Duc. *Resource-Bounded Reasoning about Knowledge*. PhD thesis, Faculty of Mathematics and Informatics, University of Leipzig, 2001.

J. J. Elgot-Drapkin. *Step-Logic: Reasoning Situated in Time*. PhD thesis, University of Maryland, 1988.

J. J. Elgot-Drapkin and D. Perlis. Reasoning situated in time I: basic concepts. *Journal of Experimental & Theoretical Artificial Intelligence*, 2(1):75–98, 1990.

R. Fagin and J. Y. Halpern. Belief, Awareness, and Limited Reasoning. *Artificial Intelligence*, 34:39–76, 1988.

R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

M. Finger. Polynomial Approximations of Full Propositional Logic via Limited Bivalence. In *Logics in Artificial Intelligence, 9th European Conference, JELIA 2004*, volume 3229 of *Lecture Notes in Computer Science*, pages 526–538. Springer Berlin Heidelberg, 2004.

M. Finger and R. Wassermann. Anytime Approximations of Classical Logic from Above. *Journal of Logic and Computation*, 17(1):53–82, 2007.

M. Helmert, P. Haslum, J. Hoffmann, and R. Nissim. Merge-and-Shrink Abstraction: A Method for Generating Lower Bounds in Factored State Spaces. *Journal of the ACM*, 61 (3):16:1–16:63, 2014.

M. Henz and H.-M. Truong. SudokuSat—A Tool for Analyzing Difficult Sudoku Puzzles. In *Tools and Applications with Artificial Intelligence*, volume 166 of *Studies in Computational Intelligence*, pages 25–35. Springer Berlin Heidelberg, 2009.

J. Hintikka. *Knowledge and Belief. An Introduction to the Logic of the Two Notions*. Cornell University Press, Ithaca, New York, 1962.

M. O. Hocutt. Is epistemic logic possible? *Notre Dame Journal of Formal Logic*, 13(4): 433–453, 10 1972.

M. Jago. *Logics for Resource-Bounded Agents*. PhD thesis, University of Nottingham, 2006.

T. Q. Klassen, S. A. McIlraith, and H. J. Levesque. Towards Tractable Inference for Resource-Bounded Agents. In *Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 2015)*, forthcoming.

S. C. Kleene. On Notation for Ordinal Numbers. *The Journal of Symbolic Logic*, 3(4): 150–155, 1938.

K. Konolige. A Deduction Model of Belief and its Logics. Technical Note 326, SRI International, 1984.

K. Konolige. What Awareness Isn't: A Sentential View of Implicit and Explicit Belief. In *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge (TARK 1986)*, pages 241–250, 1986.

R. Kowalski. Using Meta-Logic to Reconcile Reactive With Rational Agents. In K. R. Apt and F. Turini, editors, *Meta-logics and Logic Programming*, pages 227–242. MIT Press, 1995. Updated version online at `http://www.doc.ic.ac.uk/%7Erak/recon-abst.pdf`.

G. Lakemeyer and H. J. Levesque. Evaluation-Based Reasoning with Disjunctive Information in First-Order Knowledge Bases. In *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR 2002)*, 2002.

G. Lakemeyer and H. J. Levesque. Decidable Reasoning in a Logic of Limited Belief with Introspection and Unknown Individuals. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 969–975, 2013.

G. Lakemeyer and H. J. Levesque. Decidable Reasoning in a Fragment of the Epistemic Situation Calculus. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference (KR 2014)*, 2014.

H. J. Levesque. A Logic of Implicit and Explicit Belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI 1984)*, pages 198–202, 1984.

H. J. Levesque. Logic and the complexity of reasoning. *Journal of Philosophical Logic*, 17 (4):355–389, 1988.

H. J. Levesque. A knowledge-level account of abduction. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI 1989)*, pages 1061–1067, 1989.

Y. Liu. *Tractable Reasoning in Incomplete First-order Knowledge Bases*. PhD thesis, University of Toronto, 2006.

Y. Liu, G. Lakemeyer, and H. J. Levesque. A Logic of Limited Belief for Reasoning with Disjunctive Information. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR 2004)*, pages 587–597, 2004.

I. Lynce and J. Ouaknine. Sudoku as a SAT Problem. In *Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006. Proceedings online at `http://anytime.cs.umass.edu/aimath06/`.

G. L. McArthur. Reasoning about knowledge and belief: a survey. *Computational Intelligence*, 4(3):223–243, 1988.

J. McCarthy. Epistemological Problems of Artificial Intelligence. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI 1977)*, pages 1038–1044, 1977.

G. McGuire, B. Tugemann, and G. Civario. There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem via Hitting Set Enumeration. *CoRR*, abs/1201.0749, 2012.

R. Montague. Pragmatics. In R. Klibansky, editor, *Contemporary Philosophy*, pages 102–122. La Nuova Italia Editrice, Firenze, 1968.

R. C. Moore. Semantical Considerations on Nonmonotonic Logic. *Artificial Intelligence*, 25 (1):75–94, 1985.

A. Moreno. Avoiding logical omniscience and perfect reasoning: a survey. *AI Communications*, 11(2):101, 1998.

E. T. Mueller. Prospects for in-depth story understanding by computer, 1999. Online at `http://cogprints.org/554/`.

M. Nirkhe, S. Kraus, and D. Perlis. Thinking takes time: A modal active-logic for reasoning in time. In *Proceedings of the Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence*, 1995.

R. Parikh. Knowledge and the problem of logical omniscience. In *Methodologies for Intelligent Systems, Proceedings of the Second International Symposium*, pages 432–439, 1987.

R. Reiter. Narratives as Programs. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR 2000)*, pages 99–108, 2000.

G. Royle. Good at Sudoku? Here's some you'll never complete. *The Conversation*, February 12 2012. Online at `http://theconversation.com/good-at-sudoku-heres-some-youll-never-complete-5234`.

M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249 – 310, 1995.

D. Scott. Advice on Modal Logic. In K. Lambert, editor, *Philosophical Problems in Logic*. D. Reidel, Dordrecht, Holland, 1970.

K. M. Sim. Epistemic Logic and Logical Omniscience: A Survey. *International Journal of Intelligent Systems*, 12(1):57–81, 1997.

R. Stalnaker. The Problem of Logical Omniscience, I. *Synthese*, 89(3):425–440, 1991.

H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, Dordrecht, The Netherlands, 2007.

M. Y. Vardi. On Epistemic Logic and Logical Omniscience. In *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge (TARK 1986)*, pages 293–305, 1986.

M. Whitsey. Logical Omniscience: A Survey. Technical Report NOTTCS-WP-2003-2, School of Computer Science and Information Technology, University of Nottingham, 2003.