Studies in Logic 97

The Logic of Knowledge Bases

Second Edition

Hector Levesque
Gerhard Lakemeyer

Studies in Logic

Volume 97

The Logic of Knowledge Bases

Second Edition

Volume 90
Model Theory for Beginners. 15 Lectures
Roman Kossak

Volume 91
A View of Connexive Logics
Nissim Francez

Volume 92
Aristotle's Syllogistic Underlying Logic: His Model with his Proofs of Soundness and Completeness
George Boger

Volume 93
Truth and Knowledge
Karl Schlechta

Volume 94
A Lambda Calculus Satellite
Henk Barendregt and Giulio Manzonetto

Volume 95
Transparent Intensional Logic. Selected Recent Essays
Marie Duží, Daniela Glaviničová, Bjørn Jespersen and Miloš Kosterec, eds

Volume 96
BCK Algebras versus m-BCK Algebras. Foundations
Afrodita Iorgulescu

Volume 97
The Logic of Knowledge Bases, Second Edition
Hector Levesque and Gerhard Lakemeyer

Studies in Logic Series Editor
Dov Gabbay dov.gabbay@kcl.ac.uk

The Logic of Knowledge Bases

Second Edition

Hector Levesque

Gerhard Lakemeyer

ISBN 978-1-84890-420-0

College Publications
Scientific Director: Dov Gabbay
Managing Director: Jane Spurr

Cover image “Stationary High” by Christopher Pratt.
Reproduced with permission of the National Gallery of Canada.

http://www.collegepublications.co.uk

To Marc and Michelle,
Jan and Leilani.

Contents

Foreword to the Second Edition

This book is the second edition of one with the same name and authors published by MIT Press in 2000. Both editions have exactly the same focus: the concept of *knowledge* as it applies to an agent that represents what it knows in a symbolic knowledge base consisting of formulas of the first-order predicate calculus, and that reasons from this knowledge base using logic and introspection.

The book is divided into two parts: Part I, Chapters 1 to 8, presents the basics of a dialect of the predicate calculus augmented with epistemic operators for knowing and (what is called) only-knowing; Part II, Chapters 9 to 14, is a collection of independent chapters that consider various research topics relating to Part I.

In this new edition of the book, Part I is the same except for the correction of a few small errors. Part II, however, has been rewritten to take into account the research on these topics by us and our colleagues over the last twenty years. The order of chapters 9 and 10 has been switched, but the main changes have to do with chapters 11 to 14:

- Chapter 11 is concerned with default reasoning. A knowledge base is now considered to contain a predicate calculus component as before, but also a separate collection of default rules. Three forms of default reasoning are considered, due to Robert Moore, Kurt Konolige, and Ray Reiter (whose account is the most studied of the three). These modes of default reasoning are characterized semantically in terms of only-knowing.
- Chapter 12 is the first of two chapters that explore what can be done to make the reasoning required of an agent computationally tractable. In this chapter, this is done by restricting the knowledge base to predicate calculus formulas of a certain form. The emphasis in the chapter is on keeping the representation language as expressive as possible while preserving the tractability of the required logical reasoning.
- Chapter 13 is perhaps the most demanding chapter in the book. Again the goal is to keep the reasoning computationally tractable, but here the knowledge base is once again allowed to use arbitrary predicate calculus formulas. Tractability is ensured by defining a new, more complex model of knowledge that requires calculating some but not all the logical consequences of the knowledge base.
- Chapter 14 deals with reasoning about a dynamic world. A knowledge base now contains a predicate calculus component as before to characterize initial knowledge of the world, but also a separate collection of formulas characterizing how the world changes as the result of actions available to the agent, as well as how knowledge of the world changes as the result of sensing operations available to the agent.

As in the previous edition, each of the chapters of Part II end with bibliographic notes and suggestions for future research.

Hector Levesque and Gerhard Lakemeyer
March 2022

Preface

The idea that defines the very heart of "traditional" Artificial Intelligence (AI) is due to John McCarthy: his imagined ADVICE-TAKER was a system that would decide how to act (in part) by running formal reasoning procedures over a body of explicitly represented knowledge, a *knowledge base*. The system would not so much be programmed for specific tasks as told what it needed to know, and expected to infer the rest somehow. Knowledge and advice would be given declaratively, allowing the system to operate in an undirected manner, choosing what pieces of knowledge to apply when they appeared situationally appropriate. This vision contrasts sharply with that of the traditional programmed computer system, where what information is needed and when is anticipated in advance, and embedded directly into the control structure of the program.

This is a book about the *logic* of such knowledge bases, in two distinct but related senses. On the one hand, a knowledge base is a collection of sentences in a representation language that entails a certain picture of the world represented. On the other hand, *having* a knowledge base entails being in a certain state of knowledge where a number of other epistemic properties hold. One of the principal aims of this book is to develop a detailed account of the relationship between symbolic representations of knowledge and abstract states of knowledge.

This book is intended for graduate students and researchers in AI, database management, logic, or philosophy interested in exploring in depth the foundations of knowledge, knowledge bases, knowledge-based systems, and knowledge representation and reasoning. The exploration here is a mathematical one, and we assume some familiarity with first-order predicate logic (and for motivation at least, some experience in AI).

The book presents a new mathematical model of knowledge that is not only quite general and expressive (including but going well beyond full first-order logic), but that is much more workable in practice than other models that have been proposed in the past. A reader can expect to learn from this book a style of semantic argument and formal analysis that would have been quite cumbersome, or even outside the practical reach of other approaches.

From a computer science point of view, the book also develops a new way of specifying what a knowledge representation system is supposed to do in a way that does not make assumptions about how it should do it. The reader will learn how to treat a knowledge base like an *abstract data type*, completely specified in an abstract way by the knowledge-level operations defined over it.

The book is divided into two sections: Part I, consisting of Chapters 1 to 8, covers the basics; Part II, consisting of Chapters 9 to 14, considers a number of more-or-less independent research topics and directions. (The contents of these chapters are described at the end of Chapter 1.) The material in the book has been used in graduate level courses at the authors' institutions in Canada and Germany. In one semester, it should be possible

to cover all of Part I and at least some of the advanced chapters of Part II. Exercises and bibliographic notes are included at the end of each chapter. Suggestions for further research are made at the end of the chapters of Part II. An index of the important technical terms, whose first use is underlined in the text, appears at the end of the book. Comments and corrections are most welcome and can be sent to the authors at

hector@cs.toronto.edu
gerhard@cs.rwth-aachen.de.

Although every effort has been made to keep the number of errors small, this book is offered as is, with no warranty expressed or implied.

Acknowledgments to the First Edition

This book has been in the works for about twenty years. It began in the late seventies, when John Mylopoulos suggested that we consider a small extension to the PSN representation system then under development at the University of Toronto to allow for knowledge bases with incomplete knowledge. He realized that merely extending a classical true/false semantics to include a third value for "unknown" did not work properly. Among other things, tautologies could come out unknown. What sort of semantic account would assign unknown to just those formulas whose truth values really were unknown?

In a sense this book is an attempt to answer this question in a clean and general way. It incorporates the doctoral theses of both authors at the University of Toronto, as well as a number of related conference and journal papers.

Along the way, many people contributed directly and indirectly to the research reported here. We wish to thank John Mylopoulos, of course, and other thesis committee members, Faith Fich, Graeme Hirst, Alberto Mendelzon, Ray Perrault, Charles Rackoff, Ray Reiter, John Tsotsos, Alasdair Urquhart, as well as Gerhard's external examiner, Joe Halpern, and Hector's external examiner, longtime friend, and co-conspirator, Ron Brachman. A special thanks to Alex Borgida and Jim des Rivières for the many discussions at the early stages of this work.

As the work began to progress and further develop in the eighties, we began a long-term dialogue with Joe Halpern which greatly influenced the work in very many ways, first at the Knowledge Seminar held at IBM Almaden, and then at the TARK Conferences at Asilomar. We are very grateful for the help and profound insight of Joe and his colleagues Yoram Moses, Ron Fagin, and Moshe Vardi. We also acknowledge the many illuminating discussions at these meetings with Yoav Shoham, Kurt Konolige, and Bob Moore.

Three further developments shifted the work in fruitful directions. First, in the mid eighties, we began to consider the problem of logical omniscience and a solution in terms of a computationally limited notion of belief. We again thank Joe for his contributions here, as well as Peter Patel-Schneider and Greg McArthur, both at the University of Toronto. Second, at the start of the nineties, we began to see how Bob Moore's syntactic notion of autoepistemic logic could be given a semantic characterization in terms of only-knowing. We are grateful to Vladimir Lifschitz, Victor Marek, and Grigori Schvarts, for ideas there. Third, in the late nineties, we began to explore the connection between knowing, only-knowing and action. This was greatly influenced by the ongoing work on the situation calculus and Golog at Toronto's Cognitive Robotics Group. We are indebted to Ray Reiter, Yves Lespérance, Fangzhen Lin, Richard Scherl, and the other members of the group for their insights and encouragement.

Earlier versions of this book were presented as part of a graduate course at the University of Toronto, the University of Bonn, and Aachen University of Technology. We are grateful to Bruno Errico, Koen Hindricks, Gero Iwan, Eric Joanis, Daniel Marcu, Richard

Scherl, Steven Shapiro, and Mikhail Soutchansky for comments on drafts of chapters, and to all the other students who helped debug many of the ideas.

Over the years, many other friends and colleagues contributed, in one way or another, to this project. Gerhard would like to extend a special thanks to Diane Horton and Tom Fairgrieve for their sustained friendship and hospitality during his many visits to Toronto, and to Armin B. Cremers for providing a stimulating research environment while Gerhard was at the University of Bonn. Hector would like to acknowledge Jim Delgrande, Jim des Rivières, Pat Dymond, and Patrick Feehan for their continued friendship and support.

We would also like to thank Bob Prior, Katherine Innis, and the other staff members at MIT Press, who helped in the production of this book.

Financial support for this research was gratefully received from the Natural Sciences and Engineering Research Council of Canada, the Canadian Institute for Advanced Research, the World University Service of Canada, and the German Science Foundation.

Last but nowhere near least we would like to thank our families, Pat, Margot, Marc, Michelle, Jan, and Leilani, who stood by us over all these years while this book was in the making.

Hector Levesque and Gerhard Lakemeyer
Toronto and Aachen, September 2000

Addendum to the second edition: We would like to thank Yongmei Liu for her help with an early version of a logic of limited belief. Gerhard would like to acknowledge Sheila McIlraith for hosting him during his many visits to the University of Toronto. Lastly, we would also like to thank Jane Spurr of College Publications who saw us through the publication of this second edition of the book.

Hector Levesque and Gerhard Lakemeyer
Toronto and Aachen, September 2022

Part I

1 Introduction

In a book about the logical foundations of knowledge bases, it is probably a good idea to review if only briefly how concepts like knowledge, representation, reasoning, knowledge bases, and so on are understood informally within Artificial Intelligence (AI), and why so many researchers feel that these notions are important to the AI enterprise.

1.1 Knowledge

Much of AI does indeed seem to be concerned with *knowledge*. There is knowledge representation, knowledge acquisition, knowledge engineering, knowledge bases and knowledge-based systems of various sorts. In the early eighties, during the heyday of commercial AI, there was even a slogan "Knowledge is power" used in advertisements. So what exactly is knowledge that people in AI should care so much about it? This is surely a philosophical issue, and the purpose of this chapter is not to cover in any detail what philosophers, logicians, and computer scientists have said about knowledge over the years, but only to glance at some of the issues involved and especially their bearings on AI.

1.1.1 Propositions

To get a rough sense of what knowledge is supposed to be, at least outside of AI, it is useful to look at how we talk about it informally. First, observe that when we say something like "John knows that ...," we fill in the blank with a simple *declarative sentence*. So we might say that "John knows that Mary will come to the party" or that "John knows that dinosaurs were warm blooded." This suggests that, among other things, knowledge is a relation between a knower (like John) and a *proposition*, that is, the idea expressed by a simple declarative sentence (like "Mary will come to the party").

Part of the mystery surrounding knowledge is due to the abstract nature of propositions. What can we say about them? As far as we are concerned, what matters about propositions is that they are abstract entities that can be *true* or *false*, right or wrong.[1] When we say that "John knows that p," we can just as well say that "John knows that it is true that p." Either way, to say that somebody knows something is to say that somebody has formed a judgement of some sort, and has come to realize that the world is one way and not another. In talking about this judgement, we use propositions to classify the two cases.

1 Strictly speaking, we might want to say that the *sentences* expressing the proposition are true or false, and that the propositions themselves are either factual or non-factual. Further, because of linguistic features such as indexicals (that is, words like "me" and "yesterday"), we more accurately say that it is actual tokens of sentences or their uses in contexts that are true or false, not the sentences themselves.

A similar story can be told about a sentence like "John hopes that Mary will come to the party." The same proposition is involved, but the relationship John has to it is different. Verbs like "knows," "hopes," "regrets," "fears," and "doubts" all denote *propositional attitudes*, relationships between agents and propositions. In all cases, what matters about the proposition is its truth: if John hopes that Mary will come to the party, then John is hoping that the world is one way and not another, as classified by the proposition.

Of course, there are sentences involving knowledge that do not mention a proposition. When we say "John knows who Mary is taking to the party," or "John knows how to get there," we can at least imagine the implicit propositions: "John knows that Mary is taking so-and-so to the party," or "John knows that to get to the party, you go two blocks past Main Street, turn left," and so on. On the other hand, when we say that John has a skill as in "John knows how to play piano," or a deep understanding of someone or something as in "John knows Bill well," it is not so clear that any useful proposition is involved. We will have nothing further to say about this latter form of knowledge in the book.

1.1.2 Belief

A related notion that we are concerned about, however, is the concept of *belief*. The sentence "John believes that p" is clearly related to "John knows that p." We use the former when we do not wish to claim that John's judgement about the world is necessarily accurate or held for appropriate reasons. We sometimes use it when we feel that John might not be completely convinced. In fact, we have a full range of propositional attitudes, expressed by sentences like "John is absolutely certain that p," "John is confident that p," "John is of the opinion that p," "John suspects that p," and so on, that differ only in the level of conviction they attribute. For now, we will not distinguish among *any* of them.[2] What matters is that they all share with knowledge a very basic idea: John takes the world to be one way and not another.

So when we talk about knowledge or any other propositional attitude, we are implicitly imagining a number of different ways the world could be. In some of these, Mary comes to the party; in others, she does not. When we say that John knows or believes or suspects that Mary will come to the party, we are saying that John takes it (with varying degrees of conviction) that those where Mary does not come to the party are fantasy only; they do not correspond to reality.

In this very abstract and informal picture, we can already see emerging two very different but related views of knowledge or belief. First, we can think of knowledge (or belief) as a collection of propositions held by an agent to be true. Second, we can think in terms

2 One way to understand (subjective) probability theory is as an attempt to deal in a principled way with these levels of conviction as numeric degrees of belief. This is the last we will say on this subject in the book.

different possible ways the world could be, and knowledge (or belief) as a classification of these into two groups, those that are considered incorrect, and those that are candidates for the way the world really is.

1.1.3 Representation

The interest of AI in knowledge is obviously that we want to design and build systems that know a lot about their world, enough, in fact, that they do not act unintelligently.[3] But there is more to it. Any system, AI-based or not, can be said to have knowledge about its world. Any Java compiler, for example, knows a lot about the details of the Java language. There's even the joke about a thermos "knowing" whether the liquid it contains is hot or cold, and making sure it preserves the correct one. This idea of attributing knowledge to a more-or-less complex system (or person) is what the philosopher Dennett calls "taking the intentional stance." But when people in AI talk about knowledge bases, knowledge engineering and so on, they mean more than this. They have in mind a system that not only knows a lot in the above sense, but also a system that does what it does using a representation of that knowledge.

The concept of representation is no doubt as philosophically problematic as that of knowledge. Very roughly speaking, *representation* is a relationship between two domains where the first is meant to "stand for" or take the place of the second. Usually, the first domain, the representer, is more concrete, immediate, or accessible in some way than the second. The type of representer that we will be most concerned with here is that of a formal *symbol*, that is, a character or group of them taken from some predetermined alphabet. The digit "7," for example, stands for the number 7, as does the group of letters "VII," and in other contexts, the words "sept," "sieben," and "shichi." As with all representation, it is assumed to be easier to deal with symbols (recognize them, distinguish them from each other, display them *etc.*) than with what the symbols represent. In some cases, a word like "John" might stand for something quite concrete; but many words, like "love" or "truth," stand for abstractions.

Of special concern to us is when a group of formal symbols stands for a proposition: "John loves Mary" stands for the proposition that John loves Mary. Again, the symbolic English sentence is concrete: it has distinguishable parts involving the 3 words, for example, and a recognizable syntax. The proposition, on the other hand, is abstract: it is something like a classification of the ways the world can be into two groups: those where John loves Mary, and those where he does not.

Knowledge Representation, then, is this: it is the field of study within AI concerned

3 What we call "commonsense" clearly involves considerable knowledge of a variety of sorts, at least in the sense of being able to form a judgement about different ways the world could be.

with using formal symbols to represent a collection of propositions believed by some putative agent. As we will see however, we would not want to insist that there be symbols to represent *each* of the propositions believed by the agent. There may very well be an infinite number of propositions believed, only a finite number of which are ever represented. It will be the role of reasoning to bridge the gap between what is represented and the full set of propositions believed.

1.1.4 Reasoning

So what is *reasoning*? In general, it is the formal manipulation of the symbols representing a collection of believed propositions to produce representations of new ones. It is here that we use the fact that symbols are more accessible than the propositions they represent: they must be concrete enough that we can manipulate them (move them around, take them apart, copy them, string them together) in such a way as to construct representations of new propositions.

The analogy here is with arithmetic. We can think of binary addition as being a certain formal manipulation: we start with symbols like "1011" and "10," for instance, and end up with "1101." The manipulation here is addition since the final symbol represents the sum of the numbers represented by the initial ones. Reasoning is similar: we might start with the sentences "John loves Mary" and "Mary is coming to the party," and after a certain amount of manipulation produce the sentence "Someone John loves is coming to the party." We would call this form of reasoning logical inference because the final sentence represents a logical entailment of the propositions represented by the initial ones. According to this view (first put forward, incidentally, by the philosopher Leibniz in the 17th century), reasoning is a form of calculation, not unlike arithmetic, but over symbols standing for propositions rather than numbers.

1.2 Why knowledge representation and reasoning?

Let's talk motivation: why do people in AI who want their systems to know a lot, also want their systems to represent that knowledge symbolically? The intentional stance above says nothing about what is or is not represented within a system. We can say that a system knows that p without claiming that there is anything represented within the system corresponding to that proposition. The hypothesis underlying much (but not all) of the work in AI, however, is that we want to construct systems that do contain symbolic representations with two important properties. First is that we (from the outside) can understand them as standing for propositions. Second is that the system is designed to behave the way that it does *because* of these symbolic representations. This is what Brian Smith has called the

Knowledge Representation Hypothesis:

> Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantic attribution, play a formal but causal and essential role in engendering the behaviour that manifests that knowledge.

In other words, the Knowledge Representation Hypothesis is that we will want to construct systems for which the intentional stance is grounded by design in symbolic representations. A system of this sort is called a *knowledge-based system* and the symbolic representation involved its *knowledge base* (or KB).

1.2.1 Knowledge-based systems

To see what a knowledge-based system amounts to, it is helpful to look at two very simple Prolog programs with identical behaviour.[4] The first is:

```
printColour(snow) :- !, write("It's white.").
printColour(grass) :- !, write("It's green.").
printColour(sky) :- !, write("It's yellow.").
printColour(X) :- write("Beats me.").
```

The second is:

```
printColour(X) :- colour(X,Y), !,
       write("It's "), write(Y), write(".").
printColour(X) :- write("Beats me.").
colour(snow,white).
colour(sky,yellow).
colour(X,Y) :- madeof(X,Z), colour(Z,Y).
madeof(grass,vegetation).
colour(vegetation,green).
```

Observe that both programs are able to print out the colour of various items (getting the sky wrong, as it turns out). Taking an intentional stance, both might be said to "know" that the colour of snow is white. The crucial point, however, is that only the second program is designed according to the Knowledge Representation Hypothesis.

Consider the clause `colour(snow,white)`, for example. This is a symbolic structure that we can understand as representing the proposition that snow is white, and moreover,

4 No further knowledge of Prolog is assumed beyond this motivating example.

we know, by virtue of knowing how the Prolog interpreter works, that the system prints out the appropriate colour of snow precisely *because* it bumps into this clause at just the right time. Remove the clause and the system would no longer do so.

There is no such clause in the first program. The one that comes closest is the first clause of the program which says what to print when asked about snow. But we would be hard-pressed to say that this clause literally represents a belief, except perhaps a belief about what ought to be written.

So what makes a system knowledge-based, as far as we are concerned, is not the use of a logical formalism (like Prolog), or the fact that it is complex enough to merit an intentional description involving knowledge, or the fact that what it believes is true; rather it is the presence of a KB, a collection of symbolic structures representing what it believes and reasons with during the operation of the system.

1.2.2 Why knowledge representation?

So an obvious question arises when we start thinking about the two Prolog programs of the previous section: what advantage, if any, does the knowledge-based one have? Would it not be better to "compile out" the KB and distribute this knowledge to the procedures that need it, as we did in the first program? The performance of the system would certainly be better. It can only slow a system down to have to look up facts in a KB and reason with them at runtime in order to decide what actions to take. Indeed advocates within AI of so-called "procedural knowledge" take pretty much this point of view.

When we think about the various skills we have, such as riding a bicycle or playing a piano, it certainly *feels* like we do not reason about the various actions to take (shifting our weight or moving our fingers); it seems much more like we just know what to do, and do it. In fact, if we try to think about what we are doing, we end up making a mess of it. Perhaps (the argument goes) this applies to most of our activities, making a meal, getting a job, staying alive, and so on.

Of course, when we first learn these skills, the case is not so clear: it seems like we need to think deliberately about what we are doing, even riding a bicycle. The philosopher Hubert Dreyfus first observed this paradox of "expert systems." These systems are claimed to be superior precisely because they are knowledge-based, that is, they reason over explicitly represented knowledge. But novices are the ones who think and reason, claims Dreyfus. Experts do not; they learn to recognize and to react. The difference between a chess master and a chess novice is that the novice needs to figure out what is happening and what to do, but the master just "sees" it. For this reason (among others), Dreyfus believes that the development of knowledge-based systems is completely wrong-headed, if it is attempting to duplicate human-level expertise.

So why even consider knowledge-based systems? Unfortunately, no definitive answer

can yet be given. We suspect, however, that the answer will emerge in our desire to build systems that deal with a set of tasks that is *open-ended*. For any fixed set of tasks, it might work to "compile out" what the system needs to know; but if the set of tasks is not determined in advance, the strategy will not work. The ability to make behaviour depend on explicitly represented knowledge seems to pay off when we cannot specify in advance how that knowledge will ever be used.

The best example of this, perhaps, is what happens when we read a book. Suppose we are reading about South American geography. When we find out for the first time that approximately half of the population of Peru lives in the Andes, we are in no position to distribute this piece of knowledge to the various routines that might eventually require it. Instead, it seems pretty clear that we are able to assimilate the fact in declarative form for a very wide variety of potential uses. This is the prototypical case of a knowledge-based system.

From a system design point of view, the knowledge-based approach seems to have a number of desirable features:

- We can add new tasks and easily make them depend on previous knowledge. In our Prolog program example, we can add the task of enumerating all objects of a given colour, or even of painting a picture, by making use of the KB to determine the colours.
- We can extend the existing behaviour by adding new beliefs. For example, by adding a clause saying that canaries are yellow, we automatically propagate this information to any routine that needs it.
- We can debug faulty behaviour by locating the erroneous belief of the system. In the Prolog example, by changing the clause for the colour of the sky, we automatically correct any routine that uses colour information.
- We can concisely explain and justify the behaviour of the system. Why did the program say that grass was green? It was because it believed that grass is a form of vegetation and that vegetation is green. Moreover, we are justified in saying "because" here since if we removed either of the two relevant clauses, the behaviour would indeed change.

Overall, then, the hallmark of a knowledge-based system is that by design it has the ability to be *told* facts about its world and adjust its behaviour correspondingly. We will take this up again below.

This ability to have some of our actions depend on what we believe is what the cognitive scientist Zenon Pylyshyn has called *cognitive penetrability*. Consider, for example, responding to a fire alarm. The normal response is to get up and leave the building. But we would not do so if we happened to believe that the alarm was being tested, say. There are any number of ways we might come to this belief, but they all lead to the same effect. So our response to a fire alarm is cognitively penetrable since it is conditioned on what we

can be made to believe. On the other hand, something like a blinking reflex as an object approaches your eye does not appear to be cognitively penetrable: even if you strongly believe the object will not touch you, you still blink.

1.2.3 Why reasoning?

To see the motivation behind reasoning in a knowledge-based system, it suffices to observe that we would like action to depend on what the system believes about the world, as opposed to *just* what the system has explicitly represented. In the Prolog example, there was no clause representing the belief that the colour of grass was green, but we still wanted the system to know this. In general, much of what we expect to put in a KB will involve quite general facts, which will then need to be applied to particular situations.

For example, we might represent the following two facts explicitly:

1. Patient x is allergic to medication m.
2. Anyone allergic to medication m is also allergic to medication m'.

In trying to decide if it is appropriate to prescribe medication m' for patient x, neither represented fact answers the question. Together, however, they paint a picture of a world where x is allergic to m', and this, together with other represented facts about allergies, might be sufficient to rule out the medication. So we do not want to condition behaviour only on the represented facts that we are able to *retrieve*, like in a database system. The beliefs of the system must go beyond these.

But beyond them to where? There is, as it turns out, a simple answer to this question, but one that we will argue in later chapters is too simplistic. The simple answer: the system should believe p if, according to the beliefs it has represented, the world it is imagining is one where p is true. In the above example, facts (1) and (2) are both represented. If we now imagine what the world would be like if (1) and (2) were both true, then this is a world where

3. Patient x is allergic to medication m'

is also true, even though this fact is only implicitly represented.

This is the concept of *entailment*: we say that the propositions represented by a set of sentences S entail the proposition represented by a sentence p when the truth of p is implicit in the truth of the sentences in S. In other words, if the world is such that every element of S comes out true, then p does as well. All that we require to get some notion of entailment is a language with an account of what it means for a sentence to be true or false. As we argued, if our representation language is to represent knowledge at all, it must come with such an account (again: to know p is to take p to be true). So any knowledge representation language, whatever other features it may have, whatever syntactic form it may take, whatever reasoning procedures we may define over it, ought to have a well-

defined notion of entailment.

The simple answer to what beliefs a knowledge-based system should exhibit, then, is that it should believe all and only the entailments of what it has explicitly represented. The job of reasoning, then, according to this account, is to compute the entailments of the KB.

What makes this account simplistic is that there are often quite good reasons not to calculate entailments. For one thing, it is too *difficult* computationally to decide if a sentence is entailed by the kind of KB we will want to use. Any procedure that gives us answers in a reasonable amount of time will occasionally either miss some entailments or return too many. In the former case, the reasoning process is said to be *incomplete*; in the latter case, the reasoning is said to be *unsound.*

But there are also conceptual reasons why we might consider unsound or incomplete reasoning. For example, suppose p is not entailed by a KB, but is a reasonable guess, given what is represented. We might still want to believe that p is true. To use a classic example, suppose all I know about an individual Tweety is that she is a bird. I might have a number of facts about birds in the KB, but likely they would not *entail* that Tweety flies. After all, Tweety might turn out to be an ostrich. Nonetheless, it is a reasonable assumption that Tweety flies. This is unsound reasoning since we can imagine a world where everything in the KB is true but where Tweety does not fly.

As another example, a knowledge-based system might come to believe a collection of facts from various sources which, taken together, cannot all be true. In this case, it would be inappropriate to do logically complete reasoning, since *every* sentence would then be believed. This is because for any sentence p, any world where all the sentences in the set are true is one where p is also true, since there are no such worlds. An incomplete form of reasoning would clearly be more useful here until the contradictions are dealt with, if ever.

But despite all this, it remains the case that the simplistic answer is by far the best starting point for thinking about reasoning, even if we intend to diverge from it. So while it would be a mistake to *identify* reasoning in a knowledge-based system with logically sound and complete inference, it is the right place to begin.

1.3 Knowledge representation systems

The picture of a knowledge-based system that emerges from the above discussion is one where a system performs some problem-solving activity such as deciding what medicine to prescribe, and does so intelligently by appealing at various points to what it knows: is patient x allergic to medication m? what else is x allergic to? The mechanism used by the system to answer such questions involves reasoning from a stored KB of facts about the world. It makes sense in this scenario to separate the management of the KB from the rest

of the system. The data structures within a KB and the reasoning algorithms used are not really of concern to the problem-solving system. Ultimately, what a medical system needs to find out is whether or not x is allergic to m (and perhaps how certain we are of that fact), not whether or not a certain symbolic structure occurs somewhere or can be processed in a certain way.

We take the view that it is the role of a *knowledge representation system* to manage the KB within a larger knowledge-based system. Its job is to make various sorts of information about the world available to the rest of the system based on what information it has obtained perhaps from other parts of the system and whatever reasoning it can perform.[5] So its job is smaller than that of a full knowledge-based problem solver, but larger than that of a database management system which would merely retrieve the contents of the KB. According to this view, the contents of the KB and the reasoning algorithms used by the knowledge representation system are its own business; what the rest of knowledge-based problem solver gets to find out is just what is and is not known about the world.

1.3.1 The knowledge and symbol levels

Allen Newell suggested that we can look at the knowledge in a knowledge-based system in at least two ways. At the *knowledge level*, we imagine a knowledge-based system as being in some sort of abstract epistemic state. It acquires knowledge over time, moving from state to state, and uses what it knows to carry out its activities and achieve its goals. At the *symbol level*, we also imagine that within the system somewhere there is a symbolic KB representing what the system knows, as well as reasoning procedures that make what is known available to the rest of the system. In our terms, the symbol level looks at knowledge from *within* a knowledge representation system where we deal with symbolic representations explicitly; the knowledge level looks at knowledge from *outside* the knowledge representation system, and is only concerned with what is true in the world according to this knowledge. So at the knowledge level, we are concerned with the logic of *what* a system knows; at the symbol level, within a knowledge representation system, we are concerned with *how* a system does it.

There are clearly issues of adequacy at each level. At the knowledge level, we deal with the expressive adequacy of a representation language, the characteristics of its entailment relation, including its computational complexity; at the symbol level, we ask questions about the computational architecture, the properties of the data structures and algorithms, including their algorithmic complexity.

This is similar in many ways to the specification/implementation distinction within

5 In this most general picture, we include the possibility of a knowledge representation system *learning* from what it has observed, as well as it having various levels of confidence in what it believes.

traditional computer science. The symbol level provides an implementation for the more abstract knowledge level specification. But what exactly does a knowledge level specify? What would a symbol level need to implement? In a sense, being precise about these is the topic of this book.

1.3.2 A functional view: TELL and ASK

We said that the role of a knowledge representation system was to make information available to the rest of the system based on what it had acquired and what reasoning it could perform. In other words, we imagine that there are two main classes of operations that a knowledge representation system needs to implement for the rest of the system: operations that absorb new information as it becomes available, and operations that provide information to the rest of the system as needed. In its simplest form, a knowledge representation system is passive: it is up to the rest of the system to *tell* it when there is something that should be remembered, and to *ask* it when it needs to know something. It is up to the knowledge representation system to decide what to do with what it is told, and in particular, how and when to reason so as to provide answers to questions as requested.

For a large section of the book, we will be concerned with a very simple instance of each of these operations: a **TELL** operation and an **ASK** operation each of which take as argument a sentence about the world. The idea is that the **TELL** operation informs the knowledge representation system that the sentence in question is true; the **ASK** operation asks the system whether the sentence in question is true. We can see immediately that any realistic knowledge representation system would need to do much more. At the very least, it should be possible to ask *who* or *what* satisfies a certain property (according to what is known). We will examine operations like these later; for now, we stick to the simple version.

So the idea at the knowledge level is that starting in some state of knowledge, the system can be told certain sentences and move through a sequence of states; at any point, the system believes the world is in a certain state, and can be asked if a certain sentence is true. In subsequent chapters, we will show how these two operations can be defined precisely but in a way that leaves open how they might be implemented. We will also discuss simple implementation techniques at the symbol level based on automated theorem-proving, and be able to prove that such implementations are correct with respect to the specification.

1.3.3 The interaction language

With this functional view of knowledge representation, we can see immediately that there is a difference at least conceptually between the *interaction language*, that is, the language used to tell the system or to ask it questions about the world, and the *representation lan-*

guage, the collection of symbolic structures used at the symbol level to represent what is known. There is no reason to suppose the two languages are identical, or even that what is stored constitutes a declarative language of any sort. Moreover, there are clear intuitive cases where simply storing what you have been told would be a bad thing to do.

Consider indexicals, for example, that is, terms like "I," "you," "here," and "now," that might appear in an interaction language. If a fact about the world you are told is that "*There is a treasure buried here*" for instance, it would be a bad idea to absorb this information by storing the sentence verbatim. Two weeks from now, when you decide to go looking for the treasure, it is likely no longer true that it is located "here." You need to resolve the "here" at the time you are told the fact into a description of a location that can be used in different contexts. If later the question "*Is there a treasure here?*" is asked, we would want to resolve the "here" differently. We need to distinguish between how information is communicated to or retrieved from the system and how it is represented for long-term storage.

In this book, we will not emphasize indexicals like those above (although they are mentioned in an exercise). There is an important type of indexical that we *will* want to examine in considerable detail, however, and that is one that refers to the current state of knowledge.

Suppose, for example, we have a system that is attempting to solve a murder mystery, and that all it knows so far is that Tom and Dick were at the scene of the crime (and perhaps others). If the system is told that "*The murder was not committed by anyone you currently know to have been present*," the system learns that the murderer was neither Tom nor Dick. It is this "currently" that makes the expression indexical. As the knowledge of the system changes, so will what this expression refers to, just as "here" did. Suppose the system later finds out that Harry was also at the scene and was in fact the murderer. If it is now asked "*Was the murder committed by someone you currently know to have been present?*" the correct answer is *yes*, despite what it was told before. As we will see in Chapter 3, it is extremely useful to imagine an interaction language that can use indexicals like these to request information or provide new information. But it will require us to distinguish clearly between an interaction language and any language used at the symbol level to represent facts for later use.

1.4 The rest of the book

The remaining chapters of the book are divided into two broad sections: Chapters 2 to 8 cover the basics in sequential order; Chapters 9 to 14 cover advanced research-oriented topics.

- In Chapter 2, we start with a simple interaction language, a dialect of the language of first-order logic (with which we assume familiarity). However, there are good reasons to insist on some specific representational features, such as standard names and a special treatment of equality. With these, the semantic specification of the language ends up being clearer and more manageable than classical accounts. This will be especially significant when we incorporate epistemic features.
- In Chapter 3, we extend the first-order interaction language to include an epistemic operator, resulting in a language we call $\mathcal{KL}$. This involves being clear about what we mean by an epistemic state, distinct from a world state. This will allow us, among other things, to distinguish between questions about the world (e.g. the birds that do not fly) and questions about what is known (e.g. the birds that are known not to fly).
- Since what we mean by "knowledge" is so crucial to the enterprise here, in Chapter 4, we examine properties of knowledge in detail as reflected in the semantics of the language $\mathcal{KL}$. Among other things, we examine the interplay between quantifiers and knowledge, as well as the status of knowledge about knowledge.
- In Chapter 5, we define the **TELL** and **ASK** operations for the interaction language $\mathcal{KL}$. This provides a clear knowledge-level specification of the service to be provided by a knowledge representation system. We also include in this chapter a detailed example of the kind of questions and assertions that can be handled by our definition.
- In Chapter 6, we examine the relationship between the two views of knowledge mentioned above: knowledge in a KB, and knowledge in an abstract epistemic state. In other words, we look at the relationship between the symbol-level and knowledge-level views of knowledge. As it turns out, the correspondence between the two, as required by the semantics of the language $\mathcal{KL}$, is not exact.
- In Chapter 7, we prove that, despite the results of Chapter 6, it is possible to produce a symbol-level implementation of the interaction operations **TELL** and **ASK**, based on ordinary first-order reasoning. In particular, we show that the result of a **TELL** operation on a finitely represented state can itself always be properly represented, even if the sentence contains (indexical) references to what is currently known.
- In Chapter 8, we introduce a new concept called only-knowing that captures in a purely logical setting what is behind the **TELL** and **ASK** operations. The idea is to formalize using a new epistemic operator the assertion that a sentence is not only known, but all that is known.
- In Chapter 9, we consider a proof theory for the logic of only-knowing. We show soundness and completeness in the propositional case and discuss why it is incomplete in the first-order case. Nevertheless, the axiom system allows us to obtain nontrivial derivations involving quantifiers and epistemic operators.

- In Chapter 10, we relate only-knowing to what is called Autoepistemic Logic, a special brand of so-called nonmonotonic logic which has been studied extensively in the literature. We are able to fully reconstruct Autoepistemic Logic using only-knowing and, in addition, extend it since we are using a more expressive language.
- Chapter 11 is a continuation of the previous chapter but with a focus on default reasoning. The chapter considers three forms of default reasoning, due respectively to Robert Moore, Kurt Konolige, and Ray Reiter (whose account is the most studied of the three). What is significant here is that these modes of default reasoning can now be understood not only in proof-theoretic terms, but in semantic terms via three forms of only-knowing.
- Chapter 12 is the first of two chapters that explore what can be done to make the reasoning required of an agent computationally tractable. In this chapter, this is done by restricting the form of the knowledge base. Instead of allowing arbitrary formulas of the first-order predicate calculus, only certain formulas are allowed to be used. The emphasis in the chapter is on keeping the representation language as expressive as possible while still being able to prove that the reasoning required is tractable.
- In Chapter 13, the goal once again is to keep the reasoning computationally tractable, but this time the knowledge base is allowed to contain arbitrary first-order formulas. Tractability is obtained by defining a new, more complex model of knowledge that requires calculating some but not all the logical consequences of the knowledge base.
- Chapter 14 deals with the issue of reasoning about a dynamic changing world. A knowledge base is now considered to contain a first-order component as before to characterize knowledge of the world before any actions take place, but also a separate collection of formulas characterizing how the world changes as the result of actions available to the agent, as well as how knowledge of the world changes as the result of sensing operations available to the agent. Again the characterization is done semantically in terms of only-knowing.

There are different ways of approaching this book. The first eight chapters are the core, but the remaining ones can be read more or less independently of each other. Those interested in proof systems and questions involving the logic of only-knowing should read Chapter 9; those interested in default reasoning or nonmonotonic reasoning should read Chapters 10 and 11; those interested in tractable logical reasoning and the problem of logical omniscience should read Chapters 12 or 13; finally, those interested in how knowledge relates to action (including perceptual action), for robotic applications, for example, should read Chapter 14.

1.5 Bibliographic notes

Much of the material in this chapter is shared with a textbook on knowledge representation [11]. For a collection of readings on knowledge representation, see [10]. For a more philosophical discussion on knowledge and belief see [57, 13, 49], and [53] on the difference between the two. The notes at the end of Chapters 3 and 4 discuss attempts to formalize these notions. A general discussion of propositions, declarative sentences, and sentence tokens, as bearers of truth values, including the role played by indexicals, can be found in [5]. The connection between knowledge and commonsense is discussed in [138], one of the first papers on AI. The intentional stance is presented in [33], and critically examined in [34]. On Leibniz' views about thinking as a form of calculation see, for example, [38], vol. 3, p. 422. The Knowledge Representation Hypothesis is from Brian Smith's doctoral thesis [175], the Prologue of which appears in [10]. Procedural representations of knowledge are discussed in [186], and the criticism of AI by Hubert Dreyfus can be found in [36]. Zenon Pylyshyn discusses cognitive penetrability in [156], making a strong case for propositional representations to account for human-level competence. For Newell's knowledge and symbol levels, as well as the **TELL** and **ASK** functional interface, see the notes in Chapters 5 and 6. For general references on logic and entailment, see the notes in Chapter 2. Why reasoning needs to diverge from logic is discussed in [19] and [115]. For a review of the research in knowledge representation and reasoning in terms of this divergence, see [114]. For references on default (and logically unsound) reasoning, see the notes in Chapter 11.

1.6 Exercises

1. Consider a task requiring knowledge like baking a cake. Examine a recipe and state what needs to be known to follow the recipe.
2. In considering the distinction between knowledge and belief in this book, we take the view that belief is fundamental, and that knowledge is simply belief where the outside world happens to be cooperating. Describe an interpretation of the terms where knowledge is taken to be basic, and belief is understood in terms of it.
3. Explain in what sense reacting to a loud noise is and is not cognitively penetrable.
4. It has become fashionable to attempt to achieve intelligent behaviour in AI systems without using propositional representations. Speculate on what such a system should do when reading a book on South American geography.
5. Describe some ways in which the first-hand knowledge we have of some topic goes beyond what we are able to write down in a language. What accounts for our inability to express this knowledge?

2 A First-Order Logical Language

In this chapter, we will examine the properties of a first-order logical language that is suitable as a starting point at least for communicating with a KB about some application domain. As discussed in the previous chapter, we assume some familiarity with classical logical languages, propositional and quantificational, as discussed in any number of introductory logic texts (see the bibliographic notes). Here we concentrate mainly on the differences between our dialect of first-order logic and a standard one.

2.1 Why first-order logic?

We said in the previous chapter that the only feature of an interaction language that really mattered is that we had a clear and unambiguous account of what it meant for expressions in the language to be *true* or *false*. So why use a dialect of first-order logic for knowledge representation? It seems at first glance that this language is more suitable for expressing facts about *mathematical domains* such as the domain of numbers, sets, groups, and so on. This is why, after all, the language was invented by Frege at the turn of the last century, and continues to be its main application in logical circles. These mathematical concepts, it might be thought, have very little in common with the typically vague and imprecise concepts underlying commonsense reasoning. Furthermore, quantification appears to be necessary only for stating facts about infinite domains, whereas many of the applications of knowledge representation concentrate on finite collections of objects.

The answer to these objections is best seen by considering how one might use a first-order language to express commonsense knowledge.

Each of the expressions of a first-order language in Figure 2.1 is accompanied by a gloss in English, interpreting the predicate, constant, and function symbols in the obvious way. Following these in each case is a question about what is being said.

Even though the intended domain here involves only a small collection of very simple objects, all of the facilities of full first-order logic with equality are being used. Consider the quantification in Example 4, for instance. If we are willing to assume that there are only finitely many blocks, can we not do without this universal quantifier? In one sense the answer is *yes*: we could simply state *of the blocks in the box* that they are light, as in:

$$Light(block\_b) \wedge Light(block\_e).$$

The disjunction of Example 1 can be eliminated analogously by stating which of the two disjuncts is true

$$In(block\_b, box),$$

1. $In(block\_a, box) \vee In(block\_b, box)$
   Either block A or B is in the box.
   But which one?
2. $\neg In(block\_c, box)$
   Block C is not in the box.
   But where is it?
3. $\exists x.In(x, box)$
   Something is in the box.
   But what is it?
4. $\forall x.In(x, box) \supset Light(x)$
   Everything in the box is light (in weight).
   But what are the things in the box?
5. $heaviest\_block \neq block\_a$
   The heaviest block is not block A.
   But which block is the heaviest block?
6. $heaviest\_block = favourite(john)$
   The heaviest block is also John's favourite.
   But what block is this?

Figure 2.1: Expressing knowledge in first-order logic

and similar considerations apply to the other examples.

The problem with this approach is simply that it requires us to know more than we may know in order to express what we want to express. We would need to know, for example, what blocks are in the box before we could say that they are all light. In other words, we would need to know (among other things) the answers to the questions listed after each example in order to express what the example expresses. In some applications, this knowledge will be available and it will be possible to *list directly* the properties of the objects in question without appeal to much in the way of logical notation:

$$\begin{array}{ll} On(block\_a, table) & Heavy(block\_a) \\ In(block\_b, box) & Light(block\_b) \\ \ldots & \\ heaviest\_block = block\_d & \\ favourite(john) = block\_d. & \end{array}$$

In this case, a simple language of something like

$$< object,attribute,value >$$

triples would be sufficient.

But in cases where knowledge arrives incrementally, first-order logic (or English, or German for that matter) gives us much more: it allows us to say what we want to say without having to say more than we know. That is, from the point of view of knowledge representation, the expressiveness of first-order logic lies in what it allows us to leave

unsaid.[1] Stated differently, the logical facilities of first-order logic with equality allow us to express knowledge that is *incomplete* in the sense that it does not completely pin down the facts about the situation being represented.[2] Thus what we are allowing by using a first-order language as our interaction language, is a system that can have, from a functional standpoint, incomplete knowledge of its application domain: it will be possible for the system to know that one of a set things must hold without also knowing which. This power is one of the hallmarks of knowledge representation formalisms and perhaps the major source of their complexity (conceptual and computational).

2.2 Why standard names?

Given the desirability of using disjunction, negation, equality and the rest of the baggage of first-order logic, the next question is: why not stop there? This is, after all, the place where "classical" logic (as described in text books) ends. What is the point of what we will call standard names?

Consider the expression

$$Teaches(cs\_100, best\_friend(george)).$$

This can be interpreted as saying that the best friend of George teaches a course called CS100. But if we asked "Who teaches CS100?" and were told only that it was the best friend of George, we would probably feel cheated. This *describes* the individual, but not necessarily in enough detail to *identify* him. The same could be said of an even more vague description like "it's a person with brown hair."

Given that we have the capability using first-order logic of expressing knowledge about the best friend of George without necessarily identifying him, a natural question we should consider is what it would mean to identify him. An obvious place to look is at assertions of equality. We might use something like

$$best\_friend(george) = father(bill),$$

but this doesn't seem to say who he is unless we already know who the father of Bill is. If we have

$$father(bill) = mister\_smith,$$

then again the question arises as to who is Mr. Smith.

So it seems that we have two options when it comes to knowing who somebody is. The first is simply to say that we cannot *identify* individuals directly using expressions of

1 This is not strictly true, since there are cases in knowledge representation where we also want to deal with mathematical or infinite domains. For example, we may want to state general facts that apply to all points in time, or to all situations, all events, or whatever.

2 This notion of incompleteness will be defined precisely later.

a first-order logic with equality. In this case, we would say that although we might know a collection of sentences of the form

$$Teaches(cs\_100, t), \quad (t_i = t_j), \quad (t_j \neq t_k),$$

none of these terms would be considered "special" in any way. Whether or not the system knows who any of the teachers are is something that cannot be determined by examining the known sentences. In fact, if this (admittedly nebulous) property should exist at all, it would be as a result of *non-linguistic* information that the system has acquired. From the point of view of a purely linguistic functional interface, we would never talk about the system knowing who or what (or when or where) something is, but only about the character of the terms t such that the system knows something expressed using t.

The second option is to say that it is a very useful concept to be able to distinguish between knowing that somebody or something must have a certain property and knowing who that individual is, *even for a system whose information is limited to a linguistic interface.* To do so, we need to introduce conventions into our language that go beyond those of standard first-order logic.

Perhaps the simplest mechanism for doing this is to imagine the space of all terms as partitioned into equivalence classes, where terms t_1 and t_2 are considered equivalent if $(t_1 = t_2)$ is true. Now imagine naming all potential equivalence classes using $^{\#}1, ^{\#}2, ^{\#}3$, and so on. We call these terms *standard names*. Then, we can simply *decide* as a linguistic convention that we will consider a term to be identified just in case we can name which equivalence class it belongs to.

So, for example, if we know that $best\_friend(george) = {}^{\#}27$ then we will say that we know who the best friend of George is. Similarly, if we do not know something of this form for say *mister_smith*, then regardless of what else we may know about him, and specifically what other terms we know to be equal to it, we will say that we do not know who Mr. Smith is.

By convention, the question of who $^{\#}27$ is does not arise. The term $^{\#}27$, unlike say $best\_friend(george)$ is intended to carry absolutely *no useful domain-dependent information* except that it is distinct from $^{\#}26$, and all the others. So while the ordinary equalities partition the terms into equivalence classes, the standard names anchor these classes and distinguish them from each other. As such they play a role like the *unique identifiers* of database formalisms (such as object identification numbers, like social security numbers).

If there is any doubt about the identity of an individual, we should not assign it a standard name. Fortunately, the language of first-order logic allows us to express knowledge without committing ourselves to the identity of the individuals involved. In fact, we may decide never to use standard names at all and stick to statements of equality and inequality between ordinary terms. In general, however, a term can be assigned to a standard name

when we wish to express that it is distinct from all other terms that have been assigned standard names and when we do not wish to pursue further its identity.

2.3 The syntax of the language $\mathcal{L}$

We are now ready to describe the dialect of first-order logic called $\mathcal{L}$ that we will be using and the terminology and notation that goes with it.

The expressions of $\mathcal{L}$ are built up as sequences of symbols taken from the following two distinct sets:

1. the *logical symbols* consist of the following distinct sets:
   - a countably infinite supply of (individual) *variables*, written as x, y, or z, possibly with subscripts or superscripts.
   - a countably infinite supply of *standard names*, $^{\#}1$, $^{\#}2$, and so on, written schematically as n possibly with subscripts or superscripts.
   - the *equality* symbol, written $=$.
   - the usual logical connectives $\exists$, $\vee$, $\neg$ and punctuation (,), ,.
2. the *non-logical symbols* consist of two distinct sets:
   - the *predicate* symbols, written schematically as P, Q or R possibly with subscripts or superscripts, and intended to be domain specific properties and relations like *Person*, *Heavy*, *Teaches*.
   - the *function* symbols, written schematically as f, g or h possibly with subscripts or superscripts, and intended to denote mappings from individuals to individuals like *best_friend* or *father*. In case the mapping has no arguments, the function symbol is called a *constant* and is written schematically as b or c possibly with subscripts or superscripts; these are intended to denote individuals in the domain like *george*, *block_a*, or *heaviest_block*.

   Each predicate or function symbol is assumed to have an *arity*, that is, a number indicating how many arguments it takes.

The logical symbols are the part of the alphabet of $\mathcal{L}$ that will have a fixed interpretation and use; the non-logical symbols, on the other hand, are the domain-specific elements of the vocabulary. Note that the equality symbol is taken to be domain-independent and is not considered to be a predicate symbol.

We are now ready to describe the expressions of $\mathcal{L}$. There are two types: *terms*, which are used to describe individuals in the application domain, and well-formed formulas or *wffs* which describe relations, properties or conditions in the application domain. We use

schema variables t and u possibly with subscripts or superscripts to range over terms, and use Greek schema variables like α, β or γ possibly with subscripts or superscripts to range over wffs. We will use capitalized variables to range over *sets* of syntactic expressions. For example, G would be a set of function symbols, and Γ, a set of wffs.

Terms fall into three syntactic categories

1. variables,
2. standard names,
3. *function applications*, written as $f(t_1, \ldots, t_k)$ where the t_i are terms, and k is the arity of f. If the function symbol is a constant, it can be written without parentheses, as c instead of $c()$.

A term that contains no variables is called a *ground term*, and a ground term containing only a single function symbol is called a *primitive term*. In other words, a primitive term is of the form $f(n_1, \ldots, n_k)$, where $k \geq 0$, that is, a function application all of whose arguments are standard names.

The wffs of $\mathcal{L}$ are divided syntactically into atomic and non-atomic cases. The atomic formulas or *atoms* of $\mathcal{L}$ are of the form $P(t_1, \ldots, t_k)$, where P is a predicate symbol, the t_i are terms, and k is the arity of P. As with function applications, a *ground atom* is one without variables, and a *primitive atom* is a ground one where every t_i is a standard name. Thus, primitive expressions, both terms and atoms, contain a single non-logical symbol.

In general, a wff is one of the following:

1. an atom,
2. $(t_1 = t_2)$,
3. $\neg\alpha$,
4. $(\alpha \vee \beta)$,
5. $\exists x\alpha$.

As is the custom, we omit parentheses when the context is clear, use square parentheses or periods to increase readability, and freely write the usual abbreviations:

$$\forall x\alpha,\ (\alpha \wedge \beta),\ (\alpha \supset \beta),\ (\alpha \equiv \beta).$$

We also need the usual notion of a free or bound occurrence of a variable in a wff. The rigorous specification of these notions can be found in logic books, but informally, an occurrence of x in a wff is *bound* if it is located within a subwff of the form $\exists x.\alpha$ (in which case we say that it is within the *scope* of that quantifier $\exists$), and *free* otherwise. We use the notation α_t^x to mean the wff that results from textually replacing all free occurrences of the variable x in the wff α by the term t. Typically, the term t here will be a standard name. If $\vec{x}$ and $\vec{t}$ are sequences of variables and terms respectively and of the same size, by $\alpha_{\vec{t}}^{\vec{x}}$ we mean the wff that results by simultaneously replacing each free x_i by its corresponding t_i.

One last syntactic notion: a *sentence* of $\mathcal{L}$ is a wff with no free variables. This is the most important syntactic category. Sentences and sentences alone will receive a truth value, can be believed, and so represent knowledge. In fact, we can think of $\mathcal{L}$ just as its set of sentences, with the rest of the syntactic machinery merely playing a supporting role.

2.4 Domain of quantification

Before discussing in detail how we will interpret the sentences of $\mathcal{L}$, we need to discuss an assumption that we make regarding the domain of quantification. This assumption, although not necessary, greatly simplifies the semantic specification of $\mathcal{L}$ and the technical analysis to follow.

The assumption is this: the application domain is considered to be isomorphic to the set of standard names. In other words, we assume that there is always a distinct object in the domain for each standard name (and so the domain must be infinite) and that each object in the domain has a distinct name (and so the domain must be countable).

So what does this rule out? First of all, unlike classical logic, we rule out domains with only finitely many individuals. This is not to say that predicates are required to be infinite; every predicate will be allowed to have a finite extension. But the sum total of *all* individuals in the domain must be infinite. If we wish to deal with finite domains, we use predicates (like *Object* or some such) and relativize what needs to be said to instances of these predicates. Another way of thinking about this is to say that the set of integers (or strings or some such) is always included in the domain of discourse even in otherwise "small" situations.

Conversely, we rule out domains where there are more objects than standard names. We do not want inaccessible individuals, that is individuals that have properties according to the predicate and function symbols, but cannot be referred to by name. If the domain is countable, this is not a real problem since we could have assigned the names differently to cover all the domain elements.

But if we had imagined a domain containing (say) the set of real numbers, this may seem to present a more serious difficulty. It is somewhat illusory, however: it is a well known result of ordinary classical logic that any satisfiable set of sentences is satisfiable in a *countable* domain.[3] What this means is that although we may be thinking of an uncountable domain like the reals, any collection of sentences of ordinary first-order logic that rules out the countable domains is guaranteed to be inconsistent! The real numbers might indeed be *compatible* with what we are talking about, but countable domains must always be too. So in the logic of $\mathcal{L}$, we simply take this one step further and imagine the

3 This is true for first-order logic, but not for higher-order logics.

domain as always being countable.

Note that although we assume the domain to be isomorphic to the set of standard names, we do not make this assumption for *constants*. As in ordinary classical logic, two constants may indeed refer to the same individual, and there may be individuals that are not named by any constant.[4] So, as we will see below, for the part of $\mathcal{L}$ that does not use standard names or equality, everything will be the same as in ordinary classical logic.

The main consequences of this assumption are:

- although it is sometimes desirable to talk about what individual in the domain a term refers to under some interpretation, it will never be *necessary* to do; instead we can talk about finding a *co-referring* standard name, since every individual has a unique name.
- it will similarly be possible to understand quantification *substitutionally*. For example, $\exists x.P(x)$ will be true just in case $P(n)$ is true for some standard name n, since every individual in the domain has a name.

It is these two assumptions that greatly simplify the semantic specification of $\mathcal{L}$.

2.5 World state

In the classical interpretation of sentences of first-order logic due to Tarski, one specifies a domain of discourse and appropriate functions and relations for the function and predicate symbols. Using these, a set of rules specify the truth value of every sentence. This is done by first considering the more general case where terms and wffs can have free variables, and using an assignment of domain elements to these variables.

In our case, the rules are much simpler. All the variability in the interpretation of sentences reduces to the understanding of the function and predicate symbols. We will still want to know the truth value of every sentence, of course, but this will be completely determined in a straightforward way, more like the way it is done in classical *propositional* logic.

Recall that in propositional logic, one specifies an interpretation by fixing an assignment to the atomic sentences, after which the truth value of the non-atomic sentences (disjunctions and negations) is recursively defined. In our case, we require two things to specify an interpretation: a truth value for each of the primitive atoms, and a standard name for each of the primitive terms. We call such an assignment a *world state* (or world, for short) and let W name the set of all world states. We use this term since it is these assignments that tell us the way the world is, relative to the language $\mathcal{L}$.

For example, suppose we only care about one function symbol *best_friend* and one

4 In this sense, what has been called the domain closure and unique name assumptions do not apply.

predicate symbol *Person*. What we need to completely describe the way things are (that is, a world state) in this language is to say who the people are, and who is the best friend of whom. To say who the people are, we need only specify which sentences of the form $Person(n)$ are true, since every individual is assumed to have a unique standard name. Similarly, we can handle best friends by specifying for each term of the form $best\_friend(n)$, the standard name of the best friend of the individual named n.[5] From this specification, as we will see below, the truth value of every sentence will be determined.

So what we have is that for any $w \in W$,

- $w[f(n_1, \ldots, n_k)]$ is a standard name, taken to be a specification of the primitive's unique co-referring standard name;
- $w[P(n_1, \ldots, n_k)]$ is either 0 or 1, taken to be a specification of the primitive's truth value, where 1 indicates truth.

In each case, we will say that w provides the *value* of the primitive expression.

2.6 Term and formula semantic evaluation

The generalization of term evaluation to non-primitive terms is straightforward. Suppose we want the value of the term $f(g(n), c)$ with respect to some w. If the value of the primitive terms $g(n)$ and c are n_1 and n_2 respectively, then the value of $f(g(n), c)$ is the value of the primitive term $f(n_1, n_2)$. In other words, to determine a co-referring standard name for a term, we recursively substitute co-referring standard names for the arguments, stopping at primitives.

More formally, the value of a ground term t at world state w, which we write as $w(t)$, is defined by

1. $w(n) = n$;
2. $w(f(t_1, \ldots, t_k)) = w[f(n_1, \ldots, n_k)]$, where $n_i = w(t_i)$.

We will never need to evaluate terms with variables.

We are now ready to state precisely what it means for a sentence α be true in a world state w, which we write as $w \models \alpha$. Informally, we proceed as follows: for atomic sentences and equalities, we evaluate the arguments and check the answers; for disjunctions and negations, we proceed as with ordinary propositional logic; for existential quantification, we use the fact that every individual has a name and consider each substitution instances of the wff in question. More precisely, we have:

1. $w \models P(t_1, \ldots, t_k)$ iff $w[P(n_1, \ldots, n_k)] = 1$, where $n_i = w(t_i)$;

5 In case n happens not to be a person, we can assign it any other name that is also not a person, for example n itself, since we only care about people.

2. $w \models (t_1 = t_2)$ iff $w(t_1)$ is the same name as $w(t_2)$;
3. $w \models \neg\alpha$ iff it is not the case that $w \models \alpha$;
4. $w \models \alpha \vee \beta$ iff $w \models \alpha$ or $w \models \beta$;
5. $w \models \exists x\alpha$ iff for some name n, $w \models \alpha^x_n$.

Again there is no need to talk about the truth or falsity of wffs with free variables, even to deal with existential quantifiers.

2.7 Satisfiability, implication and validity

The semantic specification above completely determines which sentences of $\mathcal{L}$ are true and which are false given values for the primitives. Clearly, it is the non-logical symbols that carry the semantic burden here. The standard names, for example, are the fixed reference points in terms of which the predicate and function symbols are characterized. Each primitive expression deals with a single non-logical symbol and specifies one independent aspect of its meaning, namely its value for the given arguments. There is a world state for each possible value for each possible sequence of arguments for each primitive expression.

Although many or most of these world states will not be of interest to us, they determine the complete range of what can be true according to $\mathcal{L}$. We say that a set of sentences Γ is *satisfiable* just in case there is some world state w such that $w \models \alpha$ for every α in Γ. In other words, Γ is satisfiable if there is some way the primitives could be assigned to make the sentences in Γ true. In this case, we will say that w satisfies Γ.

Although all primitives can be assigned freely and independently to their values, the semantic rules make certain sets unsatisfiable, even simple ones such as,

$$\{P(c),\ \neg P(n),\ (n = c)\}.$$

The semantic rules of $\mathcal{L}$ are such that if any two elements of this set are true, then the last one *must* be false.

To focus on what *must* hold according to the semantic rules of $\mathcal{L}$, we say that a sentence α is *logically implied* by a set of sentences Γ, which we write $\Gamma \models \alpha$ just in case the set $\Gamma \cup \{\neg\alpha\}$ is unsatisfiable. Stated differently, Γ logically implies α if the truth of Γ forces α to be true also.

To preview what is to come, the reason logical implication is so important is this. We imagine a knowledge-based system acquiring information about some world state w in an incremental fashion. At any given point, it will have at its disposal not w itself, but only information about w in the form of a collection of sentences Γ, corresponding to what it has been told. If it now has to make a decision about what holds in w, for example to answer a question, it must use Γ since this alone represents what it knows. The trouble

is that although w will satisfy Γ, in general many very different world states will as well. However, if $\Gamma \models \alpha$, then it is perfectly safe to conclude that α is also true in w since, according to the definition of entailment, any world state satisfying Γ (such as w) also satisfies α. So the information represented by Γ includes the fact that all of its implications are true in the intended world state.

Of particular concern to us is when the information represented by Γ is *finite,* that is, when Γ consists of the sentences $\{\alpha_1, \ldots, \alpha_k\}$. In this case, we can capture the notion of logical implication using a sentence of the language: it is easy to see that $\Gamma \models \alpha$ iff the sentence

$$(\neg\alpha_1 \vee \neg\alpha_2 \vee \ldots \vee \neg\alpha_k \vee \alpha)$$

comes out true at every world state. We call a sentence α *valid*, which we write as $\models \alpha$, if it is satisfied by every world state. With respect to finitely specifiable conditions, it is therefore the valid sentences of a logical language that determine what is *required* by the language. They also determine what is *allowed* by the language, since $\{\alpha_1, \ldots, \alpha_k\}$ is satisfiable iff the sentence $(\neg\alpha_1 \vee \ldots \vee \neg\alpha_k)$ is not valid. If the notion of a sentence is the culmination of the syntax of a language, the notion of validity is the culmination of its semantics. It is indeed often the case that a logic is simply identified with its set of valid sentences.

2.8 Properties of logic $\mathcal{L}$

We now examine some of the properties of the language $\mathcal{L}$ in preparation for the generalization to $\mathcal{KL}$ to follow. Mostly what we will do is to compare and contrast the properties of validity and satisfiability in $\mathcal{L}$ with their counterparts in ordinary classical first-order logic. As noted, we assume some familiarity with ordinary first-order logic, and use the terms *first-order valid*, *first-order satisfiable*, and *first-order implies* to refer to validity, satisfiablity, and implication there. When we consider a formula of $\mathcal{L}$ in classical first-order terms, what we have in mind is to interpret standard names as ordinary constants and equality as an ordinary binary predicate.

The first thing to show is that one direction of the correspondence is clear:

Theorem 2.8.1: *Let Γ be any set of sentences. If Γ is satisfiable in $\mathcal{L}$, then it is first-order satisfiable.*

Proof: Suppose w satisfies every sentence of Γ. Construct a Tarskian interpretation $\langle D, \Phi\rangle$ as follows: D is the set of standard names; $\Phi(p)$ is the set of tuples $\langle n_1, \ldots, n_k\rangle$ such that $w[P(n_1, \ldots, n_k)] = 1$; $\Phi(=)$ is the identity relation; $\Phi(f)(n_1, \ldots, n_k)$ is the

name n such that $w[f(n_1, \ldots, n_k)] = n$; and $\Phi(n)$ is n. It is not hard to show for any sentence β that $w \models \beta$ iff $\langle D, \Phi \rangle$ satisfies β. (We leave the proof as an exercise.) So the interpretation $\langle D, \Phi \rangle$ satisfies Γ. ∎

All the complications come in for the other direction. For example, the sentence (#3 = #5) is clearly first-order satisfiable, but is not satisfiable in $\mathcal{L}$. However, for sentences without standard names and equality, the correspondence is exact.

Theorem 2.8.2: *Let Γ be any set of sentences without equality or standard names. If Γ is first-order satisfiable, then it is satisfiable in $\mathcal{L}$.*

Proof: Suppose there is a Tarskian interpretation $\langle D, \Phi \rangle$ that satisfies Γ. By the Skolem-Löwenheim theorem, we may assume without loss of generality that the domain D is countably infinite. Let π be any bijection from standard names to D. Define a world state w as follows:

- $w[P(n_1, \ldots, n_k)] = 1$ iff the tuple $\langle d_1 \ldots, d_n \rangle$ is in $\Phi(p)$, where d_i is $\pi(n_i)$;
- $w[f(n_1, \ldots, n_k] = n$ iff $\Phi(f)(d_1, \ldots, d_k) = d$, where d_i is $\pi(n_i)$ and n is the standard name such that that $\pi(n) = d$.

We leave it as an exercise to show for any sentence β without standard names or equality that $w \models \beta$ iff the interpretation $\langle D, \Phi \rangle$ satisfies β. Consequently, Γ is satisfiable in $\mathcal{L}$. ∎

Corollary 2.8.3: *Suppose α does not contain standard names or equality. Then $\models \alpha$ iff α is first-order valid.*

This shows that although the specification of $\mathcal{L}$ was much simpler than the traditional Tarskian account, when it comes to sentences without equality or standard names, the two accounts give exactly the same logic.

What can we say about equality in $\mathcal{L}$, that is, how is it different from standard first-order theories of equality (assuming we first restrict our attention to wffs without standard names)? The main difference is that the following wffs are all valid in $\mathcal{L}$:

1. $\neg\exists x_1 \forall y (y = x_1)$,
2. $\neg\exists x_1 \exists x_2 \forall y (y = x_1) \vee (y = x_2)$,
3. $\neg\exists x_1 \exists x_2 \exists x_3 \forall y (y = x_1) \vee (y = x_2) \vee (y = x_3)$,

and so on. The i-th sentence in this enumeration says that there are not i individuals such that every individual is one of them; in other words, there are more than i individuals. If

we call this set of sentences Δ, then Δ as a whole guarantees that there are *infinitely* many individuals, since no (finite) i will be sufficient.

This is clearly not a property of ordinary first-order logic where finite domains are allowed. It suggests that in $\mathcal{L}$ we should avoid talking about the domain as a whole, and always restrict our attention to individuals of a certain type. In other words, we should rarely write $\forall x.\alpha$, since after all, very little of interest will be true of *everything* (including numbers, people, bits of rock, events, and so on.) Instead, we should use a form of *typed* quantification and write something like

$$\forall x(P(x) \supset \alpha),$$

relativizing what we want to say to some predicate P. In fact, sentences like those of Δ that do not contain function or predicate symbols have a special property:

Theorem 2.8.4: *If α contains only logical symbols, then either it or its negation is valid.*

Proof: Let w_1 and w_2 be any world states. Since the sentence α does not contain function or predicate symbols, by induction, $w_1 \models \alpha$ iff $w_2 \models \alpha$. Thus if α is satisfiable, it must be valid, and the theorem follows. ∎

If we think in terms of the information carried by a sentence, sentences with only logical symbols do not really express information about the world at all; they are either logically true or logically false.

To capture the precise relationship between our treatment of equality and its treatment in classical first-order logic, we need to quickly review the latter. In ordinary first-order logic, equality is treated as a regular binary predicate, but is specified to be an equivalence relation that allows substitution in arguments. So let *EQ* be the following sentences:

- reflexivity: $\forall x(x = x)$;
- symmetry: $\forall x \forall y(x = y) \supset (y = x)$;
- transitivity: $\forall x \forall y \forall z((x = y) \wedge (y = z)) \supset (x = z)$;
- substitution of equals for functions: for any function symbol f,
  $\forall x_1 \ldots \forall x_k \forall y_1 \ldots \forall y_k((x_1 = y_1) \wedge \ldots \wedge (x_k = y_k)) \supset$
  $f(x_1, \ldots, x_k) = f(y_1, \ldots, y_k)$;
- substitution of equals for predicates: for any predicate symbol P,
  $\forall x_1 \ldots \forall x_k \forall y_1 \ldots \forall y_k((x_1 = y_1) \wedge \ldots \wedge (x_k = y_k)) \supset$
  $P(x_1, \ldots, x_k) \equiv P(y_1, \ldots, y_k)$.

Then, what we have is the following:

Theorem 2.8.5: *A sentence α without standard names is valid iff $\Delta \cup EQ$ first-order*

implies α.

Proof: Left as an exercise. ∎

This theorem ensures that in $\mathcal{L}$ we get all the standard properties of equality, and moreover, that what we get beyond the standard properties are the sentences of Δ, making the domain of quantification infinite.

Finally, we turn our attention to standard names. What can we say about them? Clearly, for each name n, the wff $(n = n)$ is valid, as we would expect from any well-behaved theory of equality. Less conventionally, perhaps, is that for any pair of distinct names n_1 and n_2, the wff $(n_1 \neq n_2)$ is also valid. So it is built into the logic that each standard name is equal to exactly one name, itself.

We can see this by letting *UNA* be set of all sentences of the form $(n \neq n')$, where n and n' are distinct standard names. Then we have the following variant of Theorem 2.8.5 which now completely characterizes validity in $\mathcal{L}$ in classical first-order terms:

Theorem 2.8.6: *A sentence* α *of* $\mathcal{L}$ *is valid iff UNA* $\cup$ *EQ first-order implies* α.

Proof: Left as an exercise. ∎

As it turns out, apart from equality, standard names behave just like constants in terms of validity. In particular, we can strengthen Corollary 2.8.3 above:

Theorem 2.8.7: *Suppose* α *does not mention equality. Then* $\models \alpha$ *iff* α *is first-order valid.*

Proof: The if direction follows immediately from Theorem 2.8.1. For the only-if direction, suppose $\models \alpha$ but there is a Tarskian interpretation $\langle D, \Phi \rangle$ that satisfies $\neg\alpha$. By the Skolem-Löwenheim theorem, we may again assume without loss of generality that the domain D is countably infinite. Now let Z be any infinite set of standard names that do not appear in α, and let π be any bijection from Z to D. Define a world state w as follows:

- $w[P(n_1, \ldots, n_k)] = 1$ iff the tuple $\langle d_1 \ldots, d_n \rangle$ is in $\Phi(p)$, where d_i is $\pi(n_i)$ when $n_i \in Z$, and d_i is $\Phi(n_i)$ otherwise;
- $w[f(n_1, \ldots, n_k] = n$ iff $\Phi(f)(d_1, \ldots, d_k) = d$, where d_i is $\pi(n_i)$ when $n_i \in Z$, and d_i is $\Phi(n_i)$ otherwise, and where n is the element of Z such that that $\pi(n) = d$.

Again, we leave it as an exercise to show for any sentence β that does not mention equality or any element of Z that $w \models \beta$ iff the interpretation $\langle D, \Phi \rangle$ satisfies β. Consequently, $w \models \neg\alpha$, contradicting the assumption that $\models \alpha$. ∎

There is one more interesting property of standard names and that is, roughly, that they have no other special logical properties! In other words, the *only* feature that characterizes standard names is that they are different from each other. As a first approximation, we might want to say that if $\alpha^x_{\#1}$ is valid, then so must be $\alpha^x_{\#2}$, since there is nothing special about $^{\#}1$. But this is not quite right: imagine α is the wff $(x \neq {}^{\#}2)$; then $\alpha^x_{\#1}$ is valid, but $\alpha^x_{\#2}$ is not.

However, we can capture precisely what we want to say by talking about a consistent renaming of all the standard names in a sentence. First some notation: let * be a bijection from standard names to standard names. For any term t or wff α, we let t^* or α^* indicate the expression resulting from simultaneously replacing in t or α every name by its mapping under *. Then we get the following:

Theorem 2.8.8: *Let * be a bijection from standard names to standard names. Then α is valid iff α^* is valid.*

Proof: Here we prove the theorem only for the special case where α contains no function symbols. The more general case is left as an exercise.

To begin, let us define for any world state w, w^* to be the world state that is like w except that for any primitive wff α, $w^*[\alpha] = w[\alpha^*]$. We now show that for any α without function symbols, $w^* \models \alpha$ iff $w \models \alpha^*$. The proof is by induction over the depth of the formation tree of α.[6] The property clearly holds for atomic formulas $P(n_1, \ldots, n_k)$ by definition of w^*. For equalities, we have that $w^* \models (n_1 = n_2)$ iff n_1 and n_2 are the same iff (since * is a bijection) ${n_1}^*$ and ${n_2}^*$ are the same iff $w \models ({n_1}^* = {n_2}^*)$. For negations, we have that $w^* \models \neg\alpha$ iff it is not the case that $w^* \models \alpha$ iff (by induction) it is not the case that $w \models \alpha^*$ iff $w \models \neg\alpha^*$. For disjunctions, we have that $w^* \models (\alpha \vee \beta)$ iff $w^* \models \alpha$ or $w^* \models \beta$ iff (by induction) $w \models \alpha^*$ or $w \models \beta^*$ iff $w \models (\alpha \vee \beta)^*$. Finally, for existentials, we have that $w^* \models \exists x.\alpha$ iff for some name n, $w^* \models \alpha^x_n$ iff (by induction) for some name n, $w \models (\alpha^x_n)^*$ iff for some name n, $w \models (\alpha)^{*x}_n$ iff $w \models (\exists x.\alpha)^*$.

Now using this property, if for some w we have that $w \models \neg\alpha^*$, then for w^* we have that $w^* \models \neg\alpha$, and if for some w we have that $w \models \neg\alpha$, then for w^* we have that $w^* \models \neg\alpha^*$. So α must be valid iff α^* is. ∎

As a corollary to this we get:

Corollary 2.8.9: *Let α be a formula with free variables $x_1, \ldots, x_k$ and let * be a bijection*

6 Since this type of induction proof is used so often throughout the book, we include this first one in full detail.

that leaves the names in α unchanged. Then for any standard names $n_1, \ldots, n_k$,

$$\models \alpha^{x_1}_{n_1} \cdots {}^{x_k}_{n_k} \quad \textit{iff} \quad \models \alpha^{x_1}_{n_1{}^*} \cdots {}^{x_k}_{n_k{}^*}.$$

As a special case, we get:

Corollary 2.8.10: *Let α have a single free variable x and let n_1 and n_2 be names not appearing in α. Then, $\alpha^x_{n_1}$ is valid iff $\alpha^x_{n_2}$ is valid.*

Proof: Consider the bijection that swaps names n_1 and n_2 and leaves all other names unchanged. Then $(\alpha^x_{n_1})^*$ is $\alpha^x_{n_2}$ and the result follows from the theorem. ∎

This establishes that names that do not appear in a wff can be used interchangeably. As a consequence we get:

Corollary 2.8.11: *Let α have a single free variable x and let n be a standard name not appearing in α. Let $n_1, \ldots, n_k$ be the all the standard names appearing in α. If α^x_n is valid and all the $\alpha^x_{n_i}$ are valid, then so is $\forall x \alpha$.*

Proof: To show a universal is valid, we need only show that all of its substitution instances are valid. For standard names appearing in α, the validity is assumed; for names not appearing in α, the same argument as the previous corollary can be used with the name n. ∎

Among other things, this shows under what conditions we are entitled to infer the validity of a universal from the validity of a finite collection of substitution instances. This will play a role in the axiomatization of $\mathcal{L}$ below.

To conclude this section on the logical properties of $\mathcal{L}$, we mention one additional minor difference between $\mathcal{L}$ and ordinary first-order logic. This difference would not be apparent looking only at validity, since it concerns infinite sets of sentences. Ordinary first-order logic is *compact*, that is, a set of sentences is satisfiable iff all of its finite subsets are. But this is not true of $\mathcal{L}$: there is a set of sentences of $\mathcal{L}$ that is unsatisfiable, but all of whose proper subsets are satisfiable:

$$\{\exists x\, P(x),\ \neg P(^{\#}1),\ \neg P(^{\#}2),\ \neg P(^{\#}3),\ \ldots\}.$$

The reason for the difference is that in $\mathcal{L}$ we can *name* every domain element (using an infinite collection of sentences). In ordinary logic, the set would be satisfiable since there would be the possibility of domain elements that are not named by any term. This difference is indeed minor since it requires an infinite set of sentences to exhibit it; the finite case is completely characterized by the above theorems.

2.9 Why a proof theory for $\mathcal{L}$?

We now turn to the development of a proof theory for $\mathcal{L}$. Before doing so, however, it is worth considering *why* we care about proof theories at all in this context, since the motivation here is perhaps non-standard.

First of all, we do *not* use a proof theory because we care about the structure of sound arguments, from premises to conclusions. Nor do we use a proof theory as the basis for a computational procedure for knowledge representation purposes. In fact, there is absolutely no reason to believe that the proof theory we will present is closer in any way to a realistic computational realization of a decision procedure for $\mathcal{L}$.

Rather, we examine a proof theory for one reason only: it gives us another revealing look at the valid sentences of $\mathcal{L}$, from a very different perspective. So far, we have defined the valid sentences in terms of truth: the valid sentences are those that come out true in all world states. With a proof theory, the picture we have is of a class of sentences defined by a closure operation: we start with a basic set (the axioms), then apply operations (the rules of inferences) on elements of the set until no new members are introduced.

A good analogy here is with the idea of a formal language, that is, a set of strings taken from some alphabet. We might describe a language as being all strings of the form

$$\{a^n b \mid n \geq 0\}.$$

We might also choose to describe the language as that which is produced by the following grammar:

$$\begin{array}{lcl} S & \rightarrow & aS \\ S & \rightarrow & b \end{array}$$

The two descriptions of this simple language are complementary, and each has its utility in certain contexts. The grammatical description is most like a proof theory since it describes the language by a closure operation: the language is the least set of strings such that b is in the language, and if S is in the language, then so is S with a prepended.

With this analogy we see clearly why this proof theory should not be understood procedurally. There is a clear separation between having a grammar for a language and having a *recognizer*. In fact, an efficient recognizer may or may not use the grammar explicitly. Similarly, having a proof theory is distinct from having a program that can prove (or even recognize) theorems. Such a theorem-prover may or may not use the proof theory, since it is the valid sentences that count, not the particular axioms and rules of inference.

2.10 Universal generalization

Most of the proof theory described below is standard. The main difference involves standard names. This shows up clearly in the treatment of the rule of universal generalization

Universal generalization is the rule that allows universals to be concluded from arguments involving "arbitrary values." The standard way to phrase this rule is using open wffs, as in

From α, infer $\forall x.\alpha$.

The reason that this rule is *sound*, that is, that the conclusion is valid given that the premise is valid, involves two cases: if α does not have x free, then clearly $\forall x.\alpha$ is valid; if it does have x free, then α is talking about some particular value of x. So the argument goes: if α is valid, then there is nothing special about x, and so the universal must be valid also.

In the case of $\mathcal{L}$, we do not need to appeal to wffs with free variables since we can use standard names. A first step might be to have a rule with an infinite set of premises like this:

From $\alpha^x_{\#_1}$, $\alpha^x_{\#_2}$, $\alpha^x_{\#_3}$, ..., infer $\forall x.\alpha$.

According to our semantics, this rule is clearly sound: if α is valid for all standard names replacing x, then the universal must be valid also.

However, there is a finitary version of this rule that does the trick because of Corollary 2.8.11 discussed earlier: From $\alpha^x_{n_1}, \ldots, \alpha^x_{n_k}$ where the n_i range over all the standard names in α and at least one standard name not in α, infer $\forall x.\alpha$. Thus, to infer a universal, we need only look at a finite number of arguments, one for each name in the wff, and one extra name. The soundness of this strategy is immediate from the corollary.

2.11 The proof theory

Except for the rule of universal generalization, the proof theory for $\mathcal{L}$ is not very surprising. First, we have the following axioms, for any formula α, β, or γ, any variable x, and any closed term t:

1. $\alpha \supset (\beta \supset \alpha)$
2. $(\alpha \supset (\beta \supset \gamma)) \supset ((\alpha \supset \beta) \supset (\alpha \supset \gamma))$
3. $(\neg\beta \supset \neg\alpha) \supset ((\neg\beta \supset \alpha) \supset \beta)$
4. $\forall x(\alpha \supset \beta) \supset (\alpha \supset \forall x\beta)$, provided that x does not occur free in α
5. $\forall x\alpha \supset \alpha^x_t$
6. $(n = n) \wedge (n \neq m)$, for any distinct n, m

The rules of inference are *modus ponens* and *universal generalization*, as discussed in the previous section:

1. From α and $(\alpha \supset \beta)$, infer β.
2. From $\alpha^x_{n_1}, \ldots, \alpha^x_{n_k}$, infer $\forall x \alpha$, provided the n_i range over all names in α and at least one not in α.

As usual, we say that α is a *theorem* of $\mathcal{L}$, which we write as $\vdash \alpha$, iff there is a sequence of wffs $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $\alpha_k = \alpha$ and each α_i in the sequence is either an instance of an axiom, or follows from earlier sentences in the sequence by one of the two rules of inference. If Γ is any set of sentences, we say that Γ *derives* α, written as $\Gamma \vdash \alpha$ iff Γ contains sentences $\gamma_1, \ldots, \gamma_k$, where $k \geq 0$ and such that $\vdash ((\gamma_1 \wedge \ldots \wedge \gamma_k) \supset \alpha)$. Finally, we say that Γ is *inconsistent* if it contains sentences $\gamma_1, \gamma_2, \ldots, \gamma_k$, where $k > 0$ and such that $\{\gamma_1, \gamma_2, \ldots, \gamma_{k-1}\} \vdash \neg\gamma_k$.

The first three axioms above are typical ones that are used (with the rule of modus ponens) to characterize propositional logic. In fact, they could be replaced by any combination of axioms and rules that correctly captures ordinary propositional logic. We therefore simply state the following without proof:

Theorem 2.11.1: *A sentence α is a theorem of ordinary propositional logic iff it can be derived using just the first three axioms and the rule of modus ponens.*

The next two axioms (and the rule of universal generalization) are the typical way quantifiers are formalized in a proof theory (although as noted above, universal generalization is handled differently here). Finally, the last axiom is the one and only addition that is necessary to handle equality; the usual formalization is much more complex. Note that all that is needed to capture the properties of standard names (as distinct from other terms) are the axiom of equality and the rule of generalization.

The most important property of this proof theory (following our discussion above of its role) is that it correctly matches the semantic characterization given earlier:

Theorem 2.11.2: $\models \alpha$ *iff* $\vdash \alpha$.

Proof: The proof has two parts: *soundness* involves establishing that everything derivable is valid; *completeness* involves showing that any valid sentence is derivable.

The proof of the former is easy, and proceeds by induction on the length of the derivation of α: establish (case by case) that all instances of axioms are valid, and then show that each rule of inference preserves validity (using Corollary 2.8.11). The details are left as an exercise.

The proof of the latter is more challenging. The usual way to show that a sentence that is not derivable is not valid, is to show that any finite consistent set of sentences is satisfiable. This is sufficient since if a sentence α is not derivable, then $\{\neg\alpha\}$ must be consistent, and so $\{\neg\alpha\}$ would be satisfiable, in which case α would not be valid. We will not prove here that finite consistent sets are indeed satisfiable, since the details of the proof can be reconstructed from the proof for the more general language $\mathcal{KL}$ to follow. The basic structure of the argument, however, is to show how the set can be extended to an infinite superset that remains consistent and that contains for every sentence, either the sentence or its negation. From this set (that also has other properties), a satisfying world state w is constructed directly: for any primitive term t and primitive atom α, $w[t] = n$ iff $t = n$ is an element of the set, and $w[\alpha] = 1$ iff α is an element of the set. This style of completeness proof is called a *Henkin proof*. ∎

Thus the valid sentences are the same as those that are derivable. We obtain as an easy corollary:

Corollary 2.11.3: *If* Γ *is a finite set of sentences, then* $\Gamma \models \alpha$ *iff* $\Gamma \vdash \alpha$.

We leave it as an exercise to show that this corollary need not hold when Γ is infinite.

2.12 Example derivation

The whole point of introducing a proof theory is to provide a different perspective on the working of the semantics of $\mathcal{L}$. Consider, for example, the fact that equals can be substituted for equals as in

$$\forall y \forall x.(x = y) \supset (f(x) = f(y)).$$

We can prove this sentence valid as follows: Let w be any world state, and n and m be any standard names. If $w \models (n = m)$, then they must be the same standard name, and $f(n)$ and $f(m)$ must be the same terms. Thus $w \models (f(n) = f(m))$ also. Since this works for any pair of names and any world state, the universal must be valid. Using the proof theory of $\mathcal{L}$, we will show that the above sentence is derivable, which gives a different argument for its validity.

To show a derivation, we will list a sequence of sentences, one per line, followed by a justification. If the justification is of the form **Ax**, this means that the current line is an axiom; if it is of the form **UG**, this means that the current line is derivable from the preceding line and perhaps some earlier ones by universal generalization; if it is of the form **MP**, this means that the current line is some β and that there is a previous α such

1.	$^{\#}1 = {}^{\#}1$	**Ax**
2.	$\forall x(x = x)$	**UG**
3.	$f(^{\#}1) = f(^{\#}1)$	**MP**
4.	$(^{\#}1 = {}^{\#}1 \supset f(^{\#}1) = f(^{\#}1))$	**MP**
5.	$^{\#}2 \neq {}^{\#}1$	**Ax**
6.	$(^{\#}2 = {}^{\#}1 \supset f(^{\#}2) = f(^{\#}1))$	**MP**
7.	$\forall x(x = {}^{\#}1 \supset f(x) = f(^{\#}1))$	**UG**
8.	$\forall y \forall x(x = y \supset f(x) = f(y))$	**UG**

Figure 2.2: A sample derivation in $\mathcal{L}$

that $(\alpha \supset \beta)$ is either an axiom or on an earlier line. The derivation for the substitutivity property is in Figure 2.2.

Note how at the end, universal generalization is used twice, once for each universally quantified variable. The first application on line 7, has α being the formula

$$(x = {}^{\#}1) \supset (f(x) = f(^{\#}1)).$$

This uses one standard name, $^{\#}1$, and so to apply generalization, we need to prove two instances of α, $\alpha^x_{\#1}$ and for some n not in α, α^x_n. The former is line 4, and the latter is line 6, where n is $^{\#}2$.

The final application of universal generalization is on line 8 for the formula

$$\forall x(x = y \supset f(x) = f(y)).$$

This uses no standard names, and so all we need is an instance with the variable y replaced by any standard name. This occurs on line 7 with y replaced by $^{\#}1$.

A similar strategy would be used to prove a sentence of the form $\forall x \forall y \forall z.\beta$, where β has no standard names. In this case, 3 new standard names would be used, call them $^{\#}1$, $^{\#}2$, $^{\#}3$. Then, to conclude the universal for z, it would be necessary to prove 3 formulas: $\beta^{x\ y\ z}_{\#1\#2\#1}$, $\beta^{x\ y\ z}_{\#1\#2\#2}$, and $\beta^{x\ y\ z}_{\#1\#2\#3}$. To conclude the universal for y, 2 previous formulas are required: $\forall z.\beta^{x\ y}_{\#1\#1}$ and $\forall z.\beta^{x\ y}_{\#1\#2}$. To conclude the final sentence, only 1 previous sentence is used: $\forall y \forall z.\beta^x_{\#1}$. In general, to prove a sentence with no standard names but k universal variables, k new standard names must be introduced, and a total of $k!$ previous sentences must be established.

As a final example, we show that a standard property of first-order logic holds for $\mathcal{L}$:

Theorem 2.12.1:

$$\vdash \forall x(\alpha \supset \beta) \supset ((\forall x\alpha) \supset (\forall x\beta)).$$

Proof: The proof proceeds by deriving a universal:

$$\vdash \forall x[\forall x(\alpha \supset \beta) \supset ((\forall x\alpha) \supset \beta)].$$

To derive this universal we need to get,

$$\vdash[\forall x(\alpha \supset \beta) \supset ((\forall x\alpha) \supset \beta^x_n)],$$

for all the names n appearing in α or β, and an additional one. To derive this, we use the fact that

$$\vdash \forall x(\alpha \supset \beta) \supset (\alpha^x_n \supset \beta^x_n)$$

and

$$\vdash (\forall x\alpha \supset \alpha^x_n),$$

which are axioms, and put these two together using ordinary properties of propositional logic. Now with the above universal in hand, we get by *modus ponens* by distributing over the universal:

$$\vdash \forall x(\alpha \supset \beta) \supset \forall x((\forall x\alpha) \supset \beta).$$

Then distributing once more over the universal, we get:

$$\vdash \forall x(\alpha \supset \beta) \supset ((\forall x\alpha) \supset (\forall x\beta)),$$

which completes the proof. ∎

Thus the proof here again depends on the number of standard names in the sentence, unlike the case in ordinary logics.

2.13 Bibliographic notes

There are many excellent introductions to classical first-order logic among which [144] and [41]. The non-modal parts of [69] also offer a very clear and succinct presentation. Our use of standard names was inspired by a similar construct in a textbook by Smullyan [177]. There they were called "parameters," and this was also the name used in the first presentation of $\mathcal{L}$ in [111]. Standard names also owe much to the idea of unique identifiers (sometimes called object identifiers) in database management, for which, see [3], for example. The presentation of first-order logic here is non-standard in that it concentrates on the truth of sentences, not on the denotation of terms. This approach has been called a truth-value semantics by Leblanc [109]. Denotation and reference has been a major preoccupation of logicians, especially in modal contexts, attempting to capture the meaning of natural language noun phrases. See the references at the end of chapters 3 and 4 regarding terms and their denotations. On the substitutional interpretation of quantification, see Leblanc's paper above as well as [110], and for a more critical discussion, [82]. Logic,

and first-order logic especially, has acquired a position of prominence in Knowledge Representation. For why all of first-order logic with equality is needed, see [146]; for why we should stop there, see [140].

2.14 Exercises

1. Complete the proof of Theorem 2.8.1
2. Complete the proof of Theorem 2.8.2. (Hint: to prove the correspondence between the world and the Tarskian model it will be necessary to deal with formulas with free variables and variable substitutions. Prove by induction that for any formula β without equality or standard names, β is satisfied by the model $\langle D, \Phi\rangle$ and variable substitution μ iff w satisfies β_μ, where β_μ is β but with any free variable x replaced by the standard name n such that $\pi(n) = \mu(x)$.)
3. Prove Theorem 2.8.5. (Hint: Use the construction from the proof of Theorem 2.8.2.)
4. Prove Theorem 2.8.6. (Hint: Adapt the proof of Theorem 2.8.7.)
5. Complete the proof of Theorem 2.8.7. (Hint: adapt the hint for the proof of Theorem 2.8.2 to the set Z.)
6. In discussing the failure of compactness for $\mathcal{L}$, it was noted that there is a set of sentences without equality that is first-order satisfiable but not satisfiable. Prove that if Γ is a set of sentences without equality but where there is an infinite set of standard names that Γ does not mention, then Γ will be satisfiable iff it is first-order satiafiable. (Hint: see the proof of Theorem 2.8.7.)
7. Let * be a bijection from standard names to standard names as in Theorem 2.8.8. Suppose that w_1 and w_2 are world states that satisfy $(w_1[t])^* = w_2[t^*]$ for every primitive term t. Prove by induction that for every closed term t, $(w_1(t))^* = w_2(t^*)$.
8. Prove Theorem 2.8.8 for the case where α may contain function symbols. Hint: define w^* so that on primitive terms t, $w^*[t]$ equals $(w[t^*])^{*-1}$ (and therefore, that $(w^*[t])^* = w[t^*]$), and redo the induction using the result of the previous exercise.
9. Show that $\vdash \forall x \forall y(x = y \supset y = x)$.
10. Show that $\vdash \forall x(\alpha \wedge \beta) \equiv (\forall x\alpha) \wedge (\forall x\beta)$.
11. Show that $\vdash \exists x(t = x)$. Hint: Use the fact that $\forall x(^{\#}1 \neq x) \supset (^{\#}1 \neq {}^{\#}1)$ is an axiom. Then apply contra-positives, generalization, and specialization.
12. Show that $\vdash \exists x((\exists x\alpha) \supset \alpha)$. Hint: Show that
$$\vdash \forall x((\exists x\alpha) \wedge \neg\alpha) \supset (\exists x\alpha) \wedge (\forall x\neg\alpha),$$
then apply contrapositives.

13. Prove that the logic of $\mathcal{L}$ is sound.
14. When discussing rules of inference, some logic textbooks use the term "truth-preserving": a rule is truth-preserving if whenever the premises of the rule are true, the conclusion is also true. For example, *modus ponens* is truth-preserving. Show that our version of universal generalization is *not* truth-preserving, but is "validity-preserving." Explain why truth-preserving rules are not needed for a logic to be sound.
15. Prove that Corollary 2.11.3 fails when Γ is infinite. Hint: consider what is implied by the set of sentences $\{P(\#1), P(\#2), P(\#3), \ldots\}$. Show a consistent set of sentences that is unsatisfiable.
16. In some logic textbooks, derivability is defined directly by something like: $\Gamma \vdash \alpha$ iff there is a sequence of wffs $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $\alpha_k = \alpha$ and each α_i in the sequence is either an instance of a logical axiom, a member of Γ, or follows from earlier sentences in the sequence by one of the two rules of inference. (In this account, the theorems of the language would then be defined as the sentences derivable from the empty set of premises.) Give an example of a Γ and an α where this definition and ours diverge. Comment on why our definition of derivability is more suitable for our semantics.
17. Extend the semantic description of the language to incorporate complex predicates:
    (a) Every predicate symbol P is a predicate.
    (b) If α is a wff then $\lambda(x_1, \ldots, x_k)\alpha$ is also a predicate (of arity k).

3 An Epistemic Logical Language

In this chapter, we introduce a new logical language called $\mathcal{KL}$ that goes beyond the first-order language considered in the previous chapter. Like $\mathcal{L}$, $\mathcal{KL}$ is intended as a language for communicating with a KB, but unlike $\mathcal{L}$, in $\mathcal{KL}$ we can talk about what is or is not *known*, in addition to what is or is not true in the world. We begin by considering why a simple first-order language like $\mathcal{L}$ is insufficient by itself. It turns out that it is precisely the *incomplete knowledge* expressible using $\mathcal{L}$ that compels us to go beyond $\mathcal{L}$. We briefly consider two other strategies for dealing with this incomplete knowledge, before settling on $\mathcal{KL}$ as the cleanest and most general approach. We then discuss, first informally, and then formally, the semantics of $\mathcal{KL}$.

3.1 Why not just use $\mathcal{L}$?

Given that we imagine a KB as representing knowledge about the world as expressed in a language like $\mathcal{L}$, why would we ever want to go beyond $\mathcal{L}$? To see the reason most clearly, we will put aside temporarily the idea of a functional interface and imagine that a KB consists simply of a finite set of sentences from $\mathcal{L}$. In our examples, we will mostly use a single two-place predicate *Teach*, where the sentence $Teach(t_1, t_2)$ is intended to be true if the person referred to by t_1 teaches the person referred to by t_2 in some course. Instead of writing standard names like $^{\#}17$ for arguments, we will adopt the following convention: for this chapter, proper names starting with a "*t*" like *tina* or *tom* are to be understood not as constants, but as standard names which will be used as the *t*eacher argument; proper names starting with an "*s*" like *sara* or *sam* are to be understood as standard names which will be used as the *s*tudent argument. This is only for readability.

So, for example, we could have a KB consisting of the two sentences

$$\{Teach(ted, sue), \quad (Teach(tina, sue) \vee Teach(tara, sue))\}.$$

Note that this KB has incomplete knowledge in that it knows that one of Tina or Tara teaches Sue, but does not know which. We cannot simply ask the KB to produce a list of Sue's teachers, for instance, since it does not know who they all are. On the other hand, the system *should* know that Sue has a teacher other than Ted. In addition, it should realize that it does not know who this other teacher is. Consequently, we should be able to ask

Does Sue have a teacher who is not yet known to be her teacher?

and expect to get the answer *yes*. In other words, the system should realize that its list of Sue's teachers is currently incomplete. The reason we need to go beyond $\mathcal{L}$ is that there is no way to express this question as a sentence of $\mathcal{L}$.

3.2 Known vs. potential instances

Before going into ways of dealing with this issue, let us be clear about what we mean by saying that somebody is a known teacher. For a standard name n and a predicate P, we say that n is a *known instance* of P if the sentence $P(n)$ is known to be true; we say that n is a *potential instance* of P if the sentence $\neg P(n)$ is not known to be true. This can obviously be generalized to predicates with additional arguments or even to arbitrary open formulas.

The main point is that saying that somebody is a known teacher is not just talking about the way the world is, like saying that somebody is a teacher; it is a property of the KB in that it says that a certain sentence is known to be true. Thus we can distinguish between the following three sets of individuals: the known teachers, the actual teachers, and the potential teachers, where the first and last category depend on the state of the KB, and the middle category depends on the state of the world. For a KB whose knowledge is *accurate*, we would expect

Known instances $\subseteq$ Actual instances
Actual instances $\subseteq$ Potential instances.

For a KB whose knowledge was also *complete*, we would expect the reverse as well, so that all three sets would be the same.

Thus, the known and potential instances bound from below and above respectively the actual instances of a predicate. As more knowledge is acquired using $\mathcal{L}$, these bounds can become tighter. For example, the sentence

$$\forall x[Teacher(x) \supset (x = {}^{\#}1) \vee (x = {}^{\#}7) \vee (x = {}^{\#}9)]$$

serves to narrow the set of potential instances of the predicate to the three individuals named. It does not provide any new known instances, but rules out all but the three named. To tighten the bounds from below, the obvious way is to name an instance, as in $Teacher({}^{\#}1)$. But more generally, we can describe the set of instances using wffs like

$$\exists x[Teacher(x) \wedge x \neq {}^{\#}1],\ \ Teacher({}^{\#}3) \vee Teacher({}^{\#}7),\ \ Teacher(best\_friend({}^{\#}8)).$$

None of these directly result in more known instances, but they serve to augment what is known about the actual ones.

3.3 Three approaches to incomplete knowledge

We said above that we needed to go beyond $\mathcal{L}$ because it was impossible to express in $\mathcal{L}$ the question about whether there were any teachers that were not yet known teachers of Sue. But perhaps we were too quick in making that assessment. What about

$$\exists x Teach(x, sue) \wedge\ \neg Known\_teach(x, sue),$$

that is, why not use *Known_teach* as a predicate. In the above example, Ted would be both a teacher and a known teacher of Sue, but although one of Tina and Tara is a teacher of Sue, *neither* would be a known teacher.

The trouble with this approach concerns the relation between the two predicates *Teach* and *Known_teach*. The two predicates are clearly not independent. Observe that if we found out that Tom was a teacher of Sue, we would immediately want to conclude that Tom was a known teacher of Sue. Since this holds for any individual, it appears that the sentence

$$\forall x.\mathit{Teach}(x, \mathit{sue}) \supset \mathit{Known\_teach}(x, \mathit{sue})$$

must be true. Unfortunately it is not, since we would then get that

$$\mathit{Known\_teach}(\mathit{tina}, \mathit{sue}) \vee \mathit{Known\_teach}(\mathit{tara}, \mathit{sue}),$$

which is false since neither is a known teacher.

In a nutshell, the reason we should not have a predicate *Known_teach* is that it is not a property of the world of teachers, but of the knowledge about that world. To find out if somebody is a known teacher, it is not sufficient to look carefully at the set of teachers in the world; it depends crucially on everything else that is known.

A second approach to dealing with this issue involves using $\mathcal{L}$ but with a *3-valued logic* instead of the current 2-valued one. Instead of sentences being merely true or false, we would allow them to take on the value *unknown.* For example, the sentence *Teach*(*ted*, *sue*) would be true, but the sentence *Teach*(*tina*, *sue*) would be unknown.

The problem with this approach is how to specify the semantics of $\mathcal{L}$. In particular, the *unknown* truth value does not seem to behave like the other two. For example, in the 2-valued logic $\mathcal{L}$, the truth value of a sentence $(\alpha \vee \beta)$ is a direct function of the truth values of α and β: if either is true then the disjunction is true, and otherwise it is false. But what would the truth table be for a 3-valued logic? Suppose α and β are both unknown; the only reasonable conclusion is that the disjunction $(\alpha \vee \beta)$ should be unknown as well. For example, if *Teach*(*tom*, *sam*) and *Teach*(*tom*, *sara*) are both unknown, then so must be (*Teach*(*tom*, *sam*) $\vee$ *Teach*(*tom*, *sara*)).

Unfortunately, this does not work. In the above example, the truth value for both *Teach*(*tina*, *sue*) and *Teach*(*tara*, *sue*) is unknown, yet their disjunction is clearly known to be true (since it is one of the sentences in the KB). As with *Known_teach* above, the problem is that we cannot assign an appropriate truth value to a sentence without taking into account the totality of what is known. Again, whether or not a sentence is considered unknown is not a property of the world of teachers, but of what is known about that world.

In summary, to talk about the known teachers, it appears that we need to be able to use sentences in two distinct ways: we need to be able to say that a sentence is true or false

(in the world), and we need to be able to say that a sentence is known or unknown (by the KB). To handle the former, we use a sentence of $\mathcal{L}$ directly; this then precludes using it for the latter.

The third approach, and the one we will be using throughout, is to augment the language $\mathcal{L}$ so that for every sentence α, there is another sentence that can be read as "α is known." Then, instead of saying that α is known to be true, we would say that the sentence "α is known" is true; instead of saying α is not known to be true, we would say that "α is known" is false. By extending the language in this way, we only have to talk about which sentences are true or false as before, even though we care about which are known or unknown. It is this extension to the language that constitutes $\mathcal{KL}$.

3.4 The language $\mathcal{KL}$

Syntactically, the language $\mathcal{KL}$ is the same as $\mathcal{L}$ except that it has one extra logical symbol, $\boldsymbol{K}$, and one extra formation rule for wffs:

> If α is a formula, then $\boldsymbol{K}\alpha$ is a formula too.

Informally, $\boldsymbol{K}\alpha$ should be read as "α is currently known to be true."

Before looking at the semantics of $\mathcal{KL}$, it is worth examining these sentences informally. We can distinguish between two main types of sentences in $\mathcal{KL}$. First, the *objective* sentences of $\mathcal{KL}$ are those that are also sentences of $\mathcal{L}$. These are sentences whose truth value depends only on the state of the world; they say nothing about what is or is not known. The second category of sentence are the *subjective* sentences, which are those where every function or predicate symbol appears within the scope of a $\boldsymbol{K}$ operator. These are sentences whose truth value depends only on what is known; they say nothing about the state of the world.[1] Of course, there are also mixed sentences that are neither purely subjective nor objective, as in

$$P(^{\#}1) \wedge \neg \boldsymbol{K} Q(^{\#}1).$$

The truth value here depends on both the state of the world and the epistemic state.

For example, the objective sentence $\neg \mathit{Teach}(\mathit{tina}, \mathit{sue})$ is true or false depending on whether Tina teaches Sue; this fact may or may not be known. Similarly, the subjective sentence $\neg \boldsymbol{K}\mathit{Teach}(\mathit{tina}, \mathit{sue})$ says that it is not known that Tina teaches Sue. This says nothing about the world of teachers, in that Tina may or may not actually teach Sue; it is purely an assertion about the KB. Finally a mixed sentence like

$$\mathit{Teach}(\mathit{tara}, \mathit{sue}) \wedge \neg \boldsymbol{K}\mathit{Teach}(\mathit{tara}, \mathit{sue})$$

1 Recall that we do not assume that something known is necessarily true in the world.

talks both about the world state and the epistemic state: it says that Tara actually teaches Sue even though this is not currently known by the system.

It is worth noting that sentences of $\mathcal{L}$ that contain no predicate or function symbols like $\forall x.(x = x)$ are strictly speaking both objective and subjective, according to the above definition. As shown in Theorem 2.8.4, these special sentences do not depend on either the state of the world or on what is known, and so are either logically true (valid) or logically false (unsatisfiable).

One very important distinction that can be made by $\mathcal{KL}$ and that will come up repeatedly is that between the following two subjective sentences:

$$\boldsymbol{K}\exists x.Teach(x, sam) \quad \text{and} \quad \exists x.\boldsymbol{K}Teach(x, sam)$$

The first says of a particular sentence of $\mathcal{L}$, that it is known to be true. The second sentence says that for some value of x, a certain sentence involving x is known to be true. The first says that it is known that Sam has a teacher; the second says that there is an x for which it is known that x teaches Sam, that is, Sam has a *known* teacher.

The difference between the two would show up, for example, when all that was known was the sentence $(Teach(tom, sam) \vee Teach(tara, sam))$. In this epistemic state, the first sentence would be true since it is known that somebody teaches Sam. But the second sentence would be false since nobody is known to teach Sam. The first sentence merely requires the existence of a teacher to be known, but the second requires the KB to know *who* the teacher is. So, for example, if what was known was $Teach(tom, sam)$, then both sentences would be true.

A final feature of $\mathcal{KL}$ worth noting is that a $\boldsymbol{K}$ operator may appear within the scope of other $\boldsymbol{K}$ operators. For example, the sentence $\boldsymbol{K}\neg\boldsymbol{K}Teach(tom, sam)$ says that it is known that Tom is not known to teach Sam. The knowledge that is expressed here is not objective since it uses a $\boldsymbol{K}$ operator. This type of subjective knowledge is usually called *meta-knowledge*. The most useful application of meta-knowledge is when the object of belief is neither objective nor subjective. Consider, for example, the sentence

$$\boldsymbol{K}[\exists x.Teach(x, sue) \wedge \boldsymbol{K}Teach(x, sam)].$$

This sentence expresses knowledge that is both about the world and the epistemic state: what is known is that Sue has a teacher (world) who is among the known teachers of Sam (knowledge). This is a much stronger claim than

$$\boldsymbol{K}[\exists x.Teach(x, sue) \wedge Teach(x, sam)],$$

where what is known is that Sue has a teacher among the teachers of Sam, since the known teachers of Sam are usually a much smaller set than the teachers of Sam. One of the most powerful and useful features of $\mathcal{KL}$ is that it allows us to express this variety of meta-knowledge, knowledge about the relationship between the world and the epistemic state.

3.5 Possible worlds

In the previous section, we saw informally that the truth value of sentences of $\mathcal{KL}$ depended on both a world state and an epistemic state. Before defining the latter precisely in Section 3.8, it is worth spending some time reviewing the general idea that will be used, which is that of *possible worlds*.

The notion of a possible world goes back to the philosopher Leibniz in the 18th century, but its technical development is mainly due to Kripke beginning in 1959. Its application to knowledge is primarily due to Hintikka starting in 1962. The main idea is actually already implicit in the way we have treated the semantics of $\mathcal{L}$: although there is only one world (about which we are interested in making assertions, having knowledge and so on), there are many different *ways* the world can be, only one of which is the way it actually is. Each of these different ways is what is called a possible world. So to say there are two possible worlds is *not* to say that there are two realities (with two individuals corresponding to Socrates and so on), but that *the* world can be two different ways. A possible world is *actual* if that is the way the world really is.

To see why this notion is useful, consider the following two sentences:

1. If I put my hand into a fire, it will feel hot.
2. If I put my hand into a fire, it will feel cold.

Intuitively, we would like to claim that the first sentence is true and the second one is false. But why should that be? Let us assume (for the sake of argument) that I will never put my hand into a fire. In this case, the antecedent of the conditional is false, and so both sentences would be equally true. So although these sentences use future tenses, they cannot be understood as simple claims about the future.

The usual explanation for how we should understand these sentences and why the truth values are different is that we have to consider a possible world where I do put my hand into fire. In other words, we imagine a (so called counter-factual) possible world that is exactly like the actual one *except* that at some point, I put my hand into a fire. The claim in the first sentence is that *in this possible world* it feels hot (correct), and in the second, that it feels cold (incorrect).

The difficulty in the previous example is being precise about what it means for a possible world to be exactly like reality except for a few changes. This is a partial description of a new way things could be, and it is often not clear exactly what is being described. For example, in the possible world where I put my hand into a fire, we may assume that the laws of physics continue to apply (otherwise they would not be laws). But clearly there is more to it than just a difference in hand motions. If I burn my hand, this will be a start of a chain of consequences with potentially far-reaching implications. Moreover, in a possible

world where I am *willing* to put my hand into a fire, I obviously am very different from the way I really am. And what are the consequences of those differences? Would I live in the same city, have the same job, friends, family, and so on?

For our purposes, rather than describing possible worlds as being minimal changes to other possible worlds, we can describe them directly in terms of what sentences are true, treating any two possible worlds that satisfy the same sentences as equivalent. For the part of the world that is purely objective, we can think of a possible world as modeled by what we have called a *world state*: a specification of the values for every primitive expression.[2]

3.6 Objective knowledge in possible worlds

The relationship between knowledge and possible worlds is this: We imagine that what a knower cares about is the way the world is, that is, the possible world that is actual. At any given point, the knower will not have determined this in full detail, but perhaps some possibilities will have been ruled out. As more information is acquired, more and more possible worlds can be eliminated. Eventually, the knower may have eliminated all but a single possible world, which would then be taken to be *the* way things really are. But this final state of complete knowledge may never be achieved, and in general, incomplete knowledge will force the agent to deal with a set of possibilities.

An epistemic state, then, can be modeled by the set of possible worlds that have not been ruled out by the knower as being the actual one. For purely objective knowledge, we can think of an epistemic state as a set of world states. Consider Figure 3.1, for example.

This picture illustrates an epistemic state called e_5 where all but three possible worlds have been eliminated, w_1, w_2, and w_3. We assume that these three worlds assign a truth value to the primitive atoms as illustrated.[3] So we are imagining in this case that the knower has decided that the real world must be in one of the three world states illustrated.

What does the knower believe in this situation? The idea of the possible-world understanding of knowledge (due to Hintikka) is that what is known for sure is what would be true regardless of which possible world turns out to be the correct one. That is, we take a conservative view and say that what is known is exactly what is true in *all* the world states that make up the epistemic state. This has the effect of guaranteeing that as long as the real world is among these alternatives, what is known will be true in reality. In the figure, *Teach*(*ted*, *sue*) would be known, since it comes out true in all three worlds. The

2 The situation will be complicated by the fact that we also want to treat knowledge as part of a possible world; but for the moment, we limit ourselves to purely objective knowledge.

3 For concreteness, we may assume that they assign all other primitive atoms to false, and all primitive terms to the standard name $^{\#}1$. Nothing hinges on this assumption.

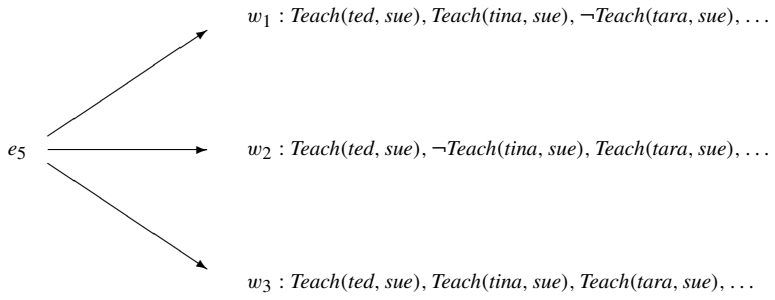

Figure 3.1: An epistemic state modeled as three world states

disjunction

$$(Teach(tina, sue) \vee Teach(tara, sue))$$

would also be known since at least one disjunct is true in each world state. On the other hand, neither $Teach(tina, sue)$ nor $Teach(tara, sue)$ is known since in either case, there is a world state where it comes out false.

With this possible-world understanding of knowledge, and unlike the 3-valued approach discussed earlier, we can see how two sentences can be unknown while their disjunction is known. Moreover, we can see what it would mean to have *complete* knowledge of the world: this corresponds to a case where the epistemic state can be modeled by a single world state. With complete knowledge, everything not known to be true is known to be false.

It need not be the case that this knowledge is *accurate*, however. To show whether or not the real world is among those in the epistemic state, we need to augment our diagrams to show which world state is actual. Thus, we will introduce a new label for a world state immediately beside the epistemic state as in Figure 3.2 as a way of indicating the real state of the world. In this figure, the knowledge of the world is indeed accurate, so everything known is true. When knowledge is both accurate and complete, the epistemic state would be modeled by the set consisting of just the real world state.

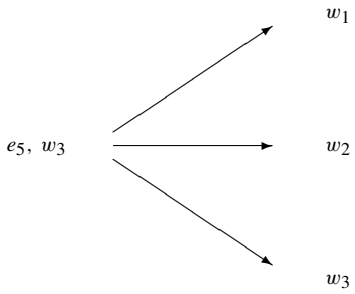

Figure 3.2: Including the actual world state

3.7 Meta-knowledge and some simplifications

So far, our possible-world account applies only to objective knowledge: each epistemic state is characterized by the (objective) world states that have not been ruled out. But as we said earlier, $\mathcal{KL}$ allows for the possibility of knowledge about the epistemic state as well.

To handle this, the simplest way is to imagine that we must deal with an enlarged notion of possible world, let us call it a *possible universe*, that consists of both a world state and an epistemic state. At any given point, only one world state and only one epistemic state will be actual. These correspond to the way the world really is and to what is really known (which is the left side of the diagrams). However, there are other possible ways the universe could be: other sentences could be true, and other sentences could be known (which is the right side of the diagrams).

To handle meta-knowledge, we assume that the agent is interested in determining both the real state of the world *and* the real state of knowledge. As before, at any point, only some of these possible universes will have been ruled out. Thus we now imagine an epistemic state as involving a set of possible universes, each consisting of both a world state and an epistemic state. Ignoring the circularity in this for a moment, the picture we have is more like that of Figure 3.3. The difference is that on the right of the diagram, instead of a list of world states, we have a list of pairs consisting of an epistemic state and a world state.

The interpretation of this diagram is this: we imagine the actual universe as being in epistemic state e_5 and world state w_3. Moreover, e_5 is an epistemic state that rules out all

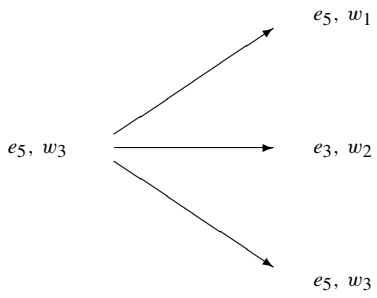

Figure 3.3: Knowledge about epistemic states

but three possibilities: in the first, the world is in w_1 and the knower is in state e_5; in the second, the world is in state w_2 and the knower in state e_3; in the final, the world is in state w_3 and the knower is in state e_5. We can already see that this knowledge is accurate since the real universe is one of the three possibilities. This does not yet complete the specification, however, since we have not yet described epistemic state e_3. It could, for example, introduce new world states and still further epistemic states requiring additional elaboration.

But without this extra complication, we can already see how meta-knowledge will be handled in simple cases. An objective sentence ϕ is considered known if it comes out true in each alternative possible universe. So for epistemic state e_5, this involves world states w_1, w_2, and w_3. Now a subjective sentence like $\boldsymbol{K}\psi$ is analogously considered known in epistemic state e_5 if it comes out true in each alternative possible universe. Thus, we would require $\boldsymbol{K}\psi$ to be true in both epistemic state e_5 and e_3, since as far as the knower is concerned, either could be the real epistemic state. So the principle is the same in both cases: to find out if an arbitrary sentence α is known, test if α is true in all of the alternative possible universes, by using the world state for the objective parts, or recursively, the epistemic state for the subjective part.

But what exactly is an epistemic state in this enlarged view? It cannot simply be a *set* of possible universes since that would require in the above example epistemic state e_5 to contain itself, among other things.

A general and very elegant way of handling this circularity was first proposed by Kripke. Instead of thinking of universes as pairs of world states and epistemic states, we can think of them as atomic indices and use two additional relations:

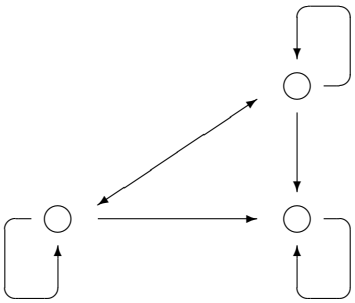

Figure 3.4: Epistemic state understood as an accessibility graph

- a relation that tells us for each index, the value of the primitive expressions at that index; this is the world state part;
- a relation (called the *accessibility* relation) that tells us for each index, what other indices are considered to be possible; this is the epistemic state part.

Ignoring the world state part, then, what we have is a set of points and a binary relation over them, which can be most clearly illustrated using a graph as in Figure 3.4. In this graph, we have three indices (corresponding to three possible universes). The arrows indicate the accessibility relation. For example, from the leftmost universe (which corresponds to $\langle e_5, w_3 \rangle$ from before), all three universes are possible; from the topmost universe (which corresponds to $\langle e_5, w_1 \rangle$ from before), again all three are possible, so the epistemic state is the same; in the bottom one, (which corresponds to $\langle e_3, w_2 \rangle$ from before), the epistemic state is different, and only a single universe is considered possible. Thus, to find out what is known with respect to any of these indices, we need only find out what is true at all the accessible indices.

While this mechanism of accessibility relations is powerful and elegant, it is too general for our needs. This is because we are willing to make a simplifying assumption about subjective knowledge:

Assumption *Purely subjective meta-knowledge is both complete and accurate.*

One way of thinking about this is that we assume that a knower, by introspection, can determine his/her true internal subjective state. To say that this subjective knowledge is complete is to say that there is no doubt in the knower about what is or is not known; to say that this subjective knowledge is accurate is to say that what the knower believes about

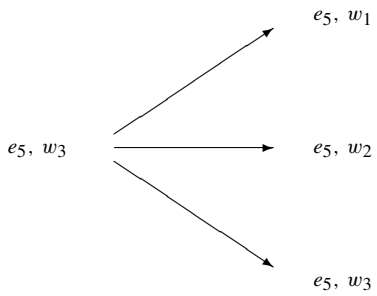

Figure 3.5: Complete and accurate meta-knowledge

this internal state is always correct.[4]

We can represent this diagrammatically as in Figure 3.5. The key observation about this picture is that while there is uncertainty about the real world state in e_5, there is no uncertainty about the epistemic state: there is a single epistemic state in the list of alternatives (this is the completeness part of the assumption) and it is indeed the correct epistemic state e_5 (this is the accuracy part).

While this simplification is not necessary and we could have continued under the more general setting of accessibility relations, it does allow us to avoid much of the complexity since there is always exactly one epistemic state accessible from another. Thus, we can treat an epistemic state simply as a *set* of world states as before, keeping the fixed epistemic part in the background. As we will see, this still leaves room for interesting meta-knowledge that is not purely subjective, and certainly sufficient richness to keep us occupied.

3.8 The semantics of $\mathcal{KL}$

With the preliminaries out of the way, we now turn to formally specifying the conditions under which sentences of $\mathcal{KL}$ are considered to be true or false (and thus, indirectly, known or unknown). As discussed above, a sentence is considered to be true or false in a universe

4 Actually, the assumption as stated here is not quite right. As discussed in the next chapter, we assume that subjective meta-knowledge is accurate only as long as the epistemic state is consistent. But this is a detail that need not concern us yet.

consisting of both a world state and an epistemic state. World states are modeled exactly as they were in $\mathcal{L}$, namely as functions from primitive expressions to their values, and epistemic states are modeled as sets of world states. We write $e, w \models \alpha$ to indicate that α is true in world state w and epistemic state e. We proceed recursively as follows: all cases except sentences dominated by $\boldsymbol{K}$ operators are handled as in $\mathcal{L}$, and $\boldsymbol{K}\alpha$ is true if α is true at every universe whose epistemic state is e and whose world state is a member of e. In detail, we have

1. $e, w \models P(t_1, \ldots, t_k)$ iff $w[P(n_1, \ldots n_k)] = 1$, where $n_i = w(t_i)$;
2. $e, w \models (t_1 = t_2)$ iff $w(t_1)$ is the same name as $w(t_2)$;
3. $e, w \models \neg\alpha$ iff it is not the case that $e, w \models \alpha$;
4. $e, w \models \alpha \vee \beta$ iff $e, w \models \alpha$ or $e, w \models \beta$;
5. $e, w \models \exists x.\alpha$ iff for some name n, $e, w \models \alpha^x_n$;
6. $e, w \models \boldsymbol{K}\alpha$ iff for every $w' \in e$, $e, w' \models \alpha$.

Except for the last clause, the definition is the same as it was with $\mathcal{L}$, with an extra parameter e carried around. In the last clause, we consider the truth of α at a range of alternative world states w', but in keeping with our assumption, the epistemic state in all these alternatives remains fixed at e.

As before, we say that a set of sentences Γ is *satisfiable* just in case there is some world state w and an epistemic state e such that $e, w \models \alpha$ for every α in Γ, in which case we say that w and e satisfy Γ. We say that α is *valid* if it is satisfied by every world and epistemic state. Finally, we say that a sentence α is *logically implied* by a set of sentences Γ, which we write $\Gamma \models \alpha$, iff the set $\Gamma \cup \{\neg\alpha\}$ is unsatisfiable.

As a notational matter, we will use the Greek letters σ and τ to range only over subjective sentences, and ϕ and ψ to range only over objective sentences. We will often write $e \models \sigma$ and $w \models \phi$ for subjective and objective sentences respectively.

In a sense we are done. The rest of the book can be thought of as an exploration of the properties of this semantic definition.

3.9 Bibliographic notes

The original idea of possible worlds goes back to Leibniz (see [38]), although the first satisfactory mathematical treatment is due to Kripke [81]. For the use of possible worlds in interpreting counterfactual conditionals, see [122] and [179]. The first to apply Kripke's possible-world model to the formalization of knowledge was Hintikka [65]. Excellent general textbooks on modal logic and possible-world semantics are [69] and [16]. The modal system we are using in this book would be called weak-S5 in the terminology of [69] or

K45 in the terminology of [16]. Much of the research effort in the area of modal logic concerns variant modal systems for different applications. The book by Fagin, Halpern, Moses and Vardi [42] offers a modern treatment of modal logic for knowledge. The authors present and discuss in depth a number of variant modal systems, many of them considerably more complex than $\mathcal{KL}$, although they restrict their attention to purely propositional languages. See [69] to get a glimpse of why general first-order modal logic is so troublesome. Much of the difficulty is caused by wanting to allow different domains of discourse in different possible worlds, corresponding to the intuition that what does or does not exist may vary from world to world. But this intuition seems to be fraught with difficulty [157, 71]. The approach we take here is that properties of objects may indeed change from world to world, including perhaps having a physical presence of some sort, but that there is only one fixed universal set of objects (possibly without physical presence) to begin with. See [67] and the references therein for a discussion of such existence assumptions.

3.10 Exercises

1. Prove that if an epistemic state contains more than one world state, then the knowledge is incomplete (that is, some sentence is neither known to be true nor known to be false).
2. Show that the truth value of a sentence where no predicate or function symbol appears within the scope of a $\boldsymbol{K}$ does not depend on the epistemic state. Similarly, show that the truth value of a sentence where no predicate or function symbol appears outside the scope of a $\boldsymbol{K}$ does not depend on the world state.

4 Logical Properties of Knowledge

In this chapter, we will undertake an analysis of the logical properties of knowledge. Since it is the semantics of the language $\mathcal{KL}$ that determines for us what it means for something to be known, we will undertake this analysis by examining closely the logical properties of $\mathcal{KL}$ itself.

We begin by showing how knowledge and truth behave similarly and differently, for objective and subjective knowledge. We then do the same for knowledge and validity. Next, we consider the issue of known individuals and how this relates to knowledge of universals. Then, we show that we have circumscribed the basic characteristics of knowledge by building an axiom system for $\mathcal{KL}$ and proving it sound and complete. Finally, given the simplifying assumption we have made about meta-knowledge, we consider the question as to whether the language itself can be simplified by eliminating all expressions of meta-knowledge.

This chapter contains two non-trivial theorems: Theorem 4.5.1 and Theorem 4.6.2. We have included the proofs of these inline as we feel that it is important to master the mathematical techniques involved in the analysis of knowledge. These techniques are further developed in exercises at the end of the chapter. Theorems or lemmas that do not contain proofs should also be thought of as exercises, with only those of special interest listed explicitly at the end.

4.1 Knowledge and truth

There are many parallels between the notions of knowledge and truth as they appear in $\mathcal{KL}$. First, it is worth noting that they are distinct notions, that is, that something can be true and not known and vice versa:

Theorem 4.1.1: *There are sentences α such that*

1. $\{\alpha \wedge \neg \boldsymbol{K}\alpha\}$ *is satisfiable;*
2. $\{\neg\alpha \wedge \boldsymbol{K}\alpha\}$ *is satisfiable.*

Proof: Let α be any primitive sentence, and choose w and w' so that $w \models \alpha$ and $w' \models \neg\alpha$. Then the first sentence above is satisfied when the world state is w and the epistemic state is $\{w, w'\}$; the second sentence above is satisfied when the world state is w' and the epistemic state is $\{w\}$. ∎

Thus, we have that for some α, $\not\models (\alpha \supset \boldsymbol{K}\alpha)$ and $\not\models (\boldsymbol{K}\alpha \supset \alpha)$. In particular, there is no requirement that what is known be true, and so, as we have noted before, the term "belief" may be more appropriate.

There is, however, a class of sentences for which there is a correspondence between knowledge and truth: the subjective sentences. In all cases, the truth of a subjective sentence implies that the sentence is also known to be true. This is shown by induction over subjective sentences, where the base case consists of equalities between standard names, sentences of the form $\boldsymbol{K}\alpha$, and their negations. So we begin with:

Lemma 4.1.2: $\models (n_1 = n_2) \supset \boldsymbol{K}(n_1 = n_2)$ *and* $\models (n_1 \neq n_2) \supset \boldsymbol{K}(n_1 \neq n_2)$.

Proof: The truth value of $(n_1 = n_2)$ does not depend on the choice of world or epistemic state, but only on whether the two names are the same or not. Consequently, if it is true (or false), it will be true (or false) for every world state in the epistemic state. Thus it will be known to be true. ∎

Lemma 4.1.3: $\models \boldsymbol{K}\alpha \supset \boldsymbol{K}\boldsymbol{K}\alpha$ *and* $\models \neg\boldsymbol{K}\alpha \supset \boldsymbol{K}\neg\boldsymbol{K}\alpha$.

Proof: The truth value of $\boldsymbol{K}\alpha$ does not depend on the world state in question but only on the epistemic state. Thus, if it is true (or false) for some pair w and e, it will also be true for w' and e for every $w' \in e$. Thus it will be known to be true. ∎

Combining these two and using induction, we obtain:

Theorem 4.1.4: *For any subjective sentence* σ, $\models (\sigma \supset \boldsymbol{K}\sigma)$.

Thus any true sentence about what is or is not known is known to be true. Another way of putting this is to say that the concept of knowledge we are dealing with is such that there is never any reason to tell a knowledge base a fact about itself; it already has *complete* knowledge about such matters.

But what about the converse? Is it the case that any subjective sentence that is known to be true is indeed true? In the previous chapter, we assumed informally that meta-knowledge was *accurate*, that is, that subjective meta-knowledge was indeed true. Here, however, we must be more precise and admit that this is not exactly right.

The complication involves the epistemic state where the knowledge is *inconsistent*. This state is modeled by the empty set of world states, meaning that all possibilities have been ruled out. In this state *all* sentences are known, and so it cannot be the case that the

known subjective sentences are all true. For example, for any primitive sentence ϕ, the sentence $\neg \boldsymbol{K}\phi$ will be false, but believed to be true. In other words, the set of sentences $\{\boldsymbol{K}\neg \boldsymbol{K}\phi,\ \boldsymbol{K}\phi\}$ is satisfiable.

However, for consistent epistemic states, we do have that any subjective meta-knowledge is true. Since a consistent epistemic state can be thought of as one where at least one sentence is not believed, we get:

Theorem 4.1.5: *For any α and any subjective σ,* $\models (\neg \boldsymbol{K}\alpha \supset (\boldsymbol{K}\sigma \supset \sigma))$.

The proof of this is similar to the proof of the previous theorem.

4.2 Knowledge and validity

There is, as it turns out, also a parallel between knowledge and validity (or provability). Consider the case of an objective sentence: an objective sentence is known iff it comes out true in all world states contained in the epistemic state; it is valid, on the other hand, if it comes out true in *all* states. Thus for objective sentences, validity coincides with a special case of knowledge, namely where the epistemic state contains all world states. But more importantly, generalizing from this observation, we have:

Theorem 4.2.1: *If* $\models \alpha$ *then* $\models \boldsymbol{K}\alpha$.

Thus, as we saw above, although true sentences need not be known in general, *valid* sentences will always be known. Thus, it is important to distinguish in $\mathcal{KL}$ between the following two claims:

1. $\models (\alpha \supset \boldsymbol{K}\alpha)$;
2. If $\models \alpha$ then $\models \boldsymbol{K}\alpha$.

Only the second one is correct.

We will consider the converse of the above theorem in a moment. First, let us consider another property of validity: closure under rules of inference. Knowledge has this property as well, for exactly the same reason:

Theorem 4.2.2: $\models \boldsymbol{K}\alpha \wedge \boldsymbol{K}(\alpha \supset \beta) \supset \boldsymbol{K}\beta$.

This says that knowledge is closed under *modus ponens*. For universal generalization, we will take the infinitary version of the rule and say that if all (infinitely many) instances of a formula are known, then so is the universal version of the formula:

Theorem 4.2.3: $\models \forall x \boldsymbol{K}\alpha \supset \boldsymbol{K}\forall x\alpha$.

The finitary version of this principle will be discussed later.

Taken together, the previous three theorems allow us to consider knowledge as some sort of provability operation. That is, recalling the derivability operation in $\mathcal{L}$, one possible characterization is the following:

1. All of the axioms are derivable.
2. If α and $(\alpha \supset \beta)$ are derivable, then so is β.
3. If α^x_n is derivable for every name n, then so is $\forall x\alpha$.

The three theorems above allow us to replace "derivable" in the above by "known to be true", as well as moving from $\mathcal{L}$ to $\mathcal{KL}$.

But a better way to look at these theorems is that they are special cases of the principle that knowledge is closed under logical consequence:

Theorem 4.2.4: *Let e be any epistemic state. Let Γ be any set of sentences such that for every $\gamma \in \Gamma$, $e \models \boldsymbol{K}\gamma$. Further, suppose that $\Gamma \models \alpha$. Then $e \models \boldsymbol{K}\alpha$.*

In other words, if (some of) what is known logically implies α, then α must be known as well. This property is sometimes referred to as *logical omniscience* since it says that a knowledge base is always "aware" of all of the logical consequences of what it knows. It is as if the knowledge base were always able to instantly do logical reasoning over everything that it knows, and therefore believe that these sentences must be true also.[1] We will collectively refer to these four theorems by the name of the most general one, Theorem 4.2.4.

Finally, let us return to the converse of Theorem 4.2.1. Is it the case, that sentences that are always believed must be logically valid? The answer is no. For example, although we do not require a knowledge base to be accurate, it turns out that a knowledge base will always *believe* that it is accurate, in the following sense:

Theorem 4.2.5: $\models \boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$

Proof: There are two cases. Suppose $\boldsymbol{K}\alpha$ is true for some e. Then $\boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$ must be true by Theorem 4.2.4, since $\{\alpha\}$ logically implies $(\beta \supset \alpha)$ for any β. On the other hand, if $\neg\boldsymbol{K}\alpha$ is true, then $\boldsymbol{K}\neg\boldsymbol{K}\alpha$ must be true by Theorem 4.1.4, and so $\boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$ is again

1 This is not a very realistic assumption for real agents (with finite resources), but it is one that makes the characterization of knowledge much simpler. We will take up the topic of relaxing this assumption later in Chapters 12 and 13.

true by Theorem 4.2.4. So for any e, $\boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$ must be true, and the theorem follows. ∎

Thus a knowledge base always believes that if it believes something, then it must be true, even though this principle is not valid. We can think of this as saying that a knowledge base is always *confident* of what it knows. It does not allow for the possibility that something it believes is false. Or put another way, this says that all the knowledge base has to go on are its beliefs, all of which are equally reliable, and so it has no *reason* (that is, no belief) to doubt anything that it believes.

If "know" is not the appropriate term here (since what is known in our sense is not required to be true), neither is "believe," at least in the sense of allowing for the fact that you might be mistaken. Perhaps a more accurate term would be "is absolutely sure of" which would not require truth, but would preclude doubts.[2]

4.3 Known individuals

As we observed before, there is more to understanding what is known than simply identifying the sentences known to be true. We also want to be able to distinguish between epistemic states where it is known that Sue has a teacher and epistemic states where the identity of that teacher is known. We can adapt the standard philosophical jargon of *de dicto* and *de re* knowledge to describe the situation. If Sue is known to have a teacher *de dicto*, this means that the sentence saying that Sue has a teacher (that is, the simple existential) is known to be true. If Sue is known to have a teacher *de re*, this means that there is some individual who is known to be a teacher of Sue. Formally, the two conditions would be expressed as follows:

de dicto: $\boldsymbol{K}\exists x\mathit{Teach}(x, \mathit{sue})$ is true at e iff for every $w \in e$, there is a name n such that $w \models \mathit{Teach}(n, \mathit{sue})$.

de re: $\exists x\boldsymbol{K}\mathit{Teach}(x, \mathit{sue})$ is true at e iff there is a name n such that for every $w \in e$, $w \models \mathit{Teach}(n, \mathit{sue})$.

Much of the richness and complexity of $\mathcal{KL}$ is a direct result of this difference, which semantically, reduces to an order of quantifiers.

The first property to observe about this distinction is that *de re* knowledge implies *de dicto*, but not vice versa.

Theorem 4.3.1: $\models (\exists x\,\boldsymbol{K}\alpha \supset \boldsymbol{K}\exists x\alpha)$ *but* $\not\models (\boldsymbol{K}\exists x\alpha \supset \exists x\,\boldsymbol{K}\alpha)$

2 This is still not right because of logical omniscience. A more accurate gloss for $\boldsymbol{K}\alpha$ would be "α follows logically from what the system is absolutely sure of," although even this is not quite right because of introspection and meta-knowledge.

This follows immediately from the semantic characterization given above. What does imply *de re* knowledge, however, is knowledge involving particular standard names:

Theorem 4.3.2: $\models \boldsymbol{K}P(n) \supset \exists x \boldsymbol{K}P(x)$.

But this does not work if there is uncertainty about the identity of the individual:

Theorem 4.3.3: *Let n_1 and n_2 be distinct. Then*

$$\not\models \boldsymbol{K}(P(n_1) \vee P(n_2)) \supset \exists x \boldsymbol{K}P(x).$$

Proof: Let w be such that $w \models P(n)$ iff $n = n_1$, and w' be such that $w \models P(n)$ iff $n = n_2$. Let $e = \{w, w'\}$. Then e satisfies the left hand side, but not the right hand side. ∎

Of course, by Theorem 4.2.4, we would still have *de dicto* knowledge here, since an existential follows from the disjunction.

Similarly, there will be no *de re* knowledge if the uncertainty about the identity of the individual is because of a non-standard name:

Theorem 4.3.4: *Suppose t is a primitive term. Then,* $\not\models (\boldsymbol{K}P(t) \supset \exists x \boldsymbol{K}P(x))$.

Proof: Let w and w' be as above except that $w[t] = n_1$ and $w'[t] = n_2$. Then again e satisfies the left hand side, but not the right hand side. ∎

In all of the above cases, what is at issue is the existence of a *fixed* individual for each of the world states making up the epistemic state. The language $\mathcal{L}$ allows us to express properties of individuals without fixing the identity of the individual in question. The fact that $(\boldsymbol{K}P(t) \supset \exists x \boldsymbol{K}P(x))$ is not valid in $\mathcal{KL}$ means that the sentence

$$\forall x.\boldsymbol{K}P(x) \supset \boldsymbol{K}P(t)$$

is not valid either. This, in turn, is an instance of $(\forall x \alpha \supset \alpha_t^x)$ which in general cannot be valid either. This is very different from the situation in the logic of $\mathcal{L}$, since this last sentence was in fact an *axiom* in the proof theory of $\mathcal{L}$, often called the *axiom of specialization*.

So why exactly does the axiom of specialization fail in $\mathcal{KL}$? Consider this example. Suppose the following sentence is true:

$$\forall x.\boldsymbol{K}\mathit{Teacher}(x) \vee \boldsymbol{K}\neg\mathit{Teacher}(x).$$

That is, for every individual x, either x is known to be a teacher or known not to be a teacher. In other words, the knowledge base has an opinion about every individual, one

way or another. This can happen, for instance, when it is known that $^{\#}3$ is the one and only teacher. Now consider the sentence

$$\boldsymbol{K}Teacher(t) \vee \boldsymbol{K}\neg Teacher(t),$$

where t is the term $best\_friend(mother(sam))$. Clearly the first sentence can be true without the second one being true, when the identity of the best friend of the mother of Sam is unknown. In particular, even if we know that $^{\#}3$ is the one and only teacher, it does not follow that we know whether or not t is a teacher, since we may not know if $(t = {}^{\#}3)$ is true or not.

This is an example of the failure of the axiom of specialization and it is due to the fact that the identity of non-standard terms may be unknown. In fact, we can show that the axiom of specialization *does* hold provided that the replacement for the variable never places a function symbol within the scope of a $\boldsymbol{K}$. First we need this lemma:

Lemma 4.3.5: *Let t be any term, w any world state, e any epistemic state. Suppose that α is a formula with at most a single free variable x, and that none of the free occurrences of x in α are within the scope of a $\boldsymbol{K}$. Assume that $w(t)$ is n. Then*

$$e, w \models \alpha^x_n \text{ iff } e, w \models \alpha^x_t.$$

Proof: The proof is by induction on the length of α. If α is an atomic sentence, this clearly holds. If α is an equality, it also holds by induction on the structure of the terms in the equality. For negations and conjunctions, the lemma holds by induction. If α is of the form $\forall y\beta$, then there are two cases: if y is the same as x, then the lemma holds trivially since x does not appear free in α; if y is distinct from x, then $e, w \models (\forall y\beta)^x_n$ iff $e, w \models \forall y(\beta^x_n)$ iff $e, w \models (\beta^x_n)^y_{n'}$ for every n', iff $e, w \models (\beta^y_{n'})^x_n$ for every n', iff (by induction) $e, w \models (\beta^y_{n'})^x_t$ for every n', iff $e, w \models (\beta^x_t)^y_{n'}$ for every n', iff $e, w \models \forall y(\beta^x_t)$ iff $e, w \models (\forall y\beta)^x_t$. Finally, if α is of the form $\boldsymbol{K}\beta$, then the lemma holds trivially since x does not occur freely within the scope of a $\boldsymbol{K}$. ∎

With this lemma, we then get:

Theorem 4.3.6: $\models (\forall x\alpha \supset \alpha^x_t)$, *provided that when replacing x by t, no function symbol is introduced within the scope of a $\boldsymbol{K}$.*

Proof: If the term t is a standard name, this follows immediately from the semantics of universal quantification. Otherwise, it must be the case that no free occurrence of x in α is within the scope of a $\boldsymbol{K}$. Suppose that for some w and e, $e, w \models \forall x\alpha$. Then we must have, $e, w \models \alpha^x_n$ for every n. In particular, consider the n which is $w(t)$. By the above lemma,

we must have $e, w \models \alpha_t^x$. ∎

Thus, specialization holds as long as the term t is a standard name or x does not appear free in α within the scope of a $\boldsymbol{K}$. Other sound restrictions of the axiom of specialization are also considered in the exercises.

As a simple consequence of this theorem we get that equals can be substituted for equals provided that this does not involve placing a non-standard term within the scope of a $\boldsymbol{K}$:

Theorem 4.3.7: *Suppose that t and t' are terms, and that α has at most a single free variable x. Further assume that neither α_t^x nor $\alpha_{t'}^x$ introduces a function symbol within the scope of a $\boldsymbol{K}$. Then,*

$$\models (t = t' \supset \alpha_t^x \equiv \alpha_{t'}^x).$$

Proof: First observe that for any pair of names, n and n', we have

$$\models (n = n' \supset \alpha_n^x \equiv \alpha_{n'}^n).$$

since equality between names holds iff the names are the same. Thus we get that

$$\models \forall y \forall y' (y = y' \supset \alpha_y^x \equiv \alpha_{y'}^x).$$

The theorem then follows immediately from Theorem 4.3.6. ∎

4.4 An axiom system for $\mathcal{KL}$

Having examined various properties of $\mathcal{KL}$, we are now ready to turn to an axiomatization of the logic. As in the case of $\mathcal{L}$, the principal reason for doing this is to provide a simple but very different picture of the valid sentences, phrased in term of an initial set (the axioms) and closure conditions (the rules of inference). As it turns out, the rules of inference we need for $\mathcal{KL}$ are just those of $\mathcal{L}$: *modus ponens* and universal generalization. So we need only list the axioms, which are in Figure 4.1. The definition of theorem, derivability, consistency, and inconsistency, are the same as they were in $\mathcal{L}$. Again we use the notation $\vdash \alpha$ to say that α is a theorem, and $\Gamma \vdash \alpha$ to say that Γ derives α.

As in $\mathcal{L}$, it is fairly easy to establish soundness:

Theorem 4.4.1: *If a sentence of $\mathcal{KL}$ is derivable, then it is valid.*

The proof of soundness, as usual, is by induction on the length of the derivation. The basis of the induction proof depends on the validity of the above axioms, all of which

1. Axioms of $\mathcal{L}$:
   All instances of the axioms of $\mathcal{L}$, but with the proviso on the axiom of specialization that no function symbol is introduced within the scope of a $\boldsymbol{K}$.
2. Knowledge of axioms:
   $\boldsymbol{K}\alpha$, where α is an instance of an axiom of $\mathcal{L}$, again with the proviso on specialization;
3. Knowledge closed under modus ponens:
   $\boldsymbol{K}(\alpha \supset \beta) \supset (\boldsymbol{K}\alpha \supset \boldsymbol{K}\beta)$;
4. Knowledge closed under universal generalization:
   $\forall x \boldsymbol{K}\alpha \supset \boldsymbol{K}\forall x\alpha$;
5. Complete knowledge of subjective truths:
   $(\sigma \supset \boldsymbol{K}\sigma)$, where σ is subjective.

Figure 4.1: Axioms for $\mathcal{KL}$

were established in the previous section. Note, for example, that without the proviso on the axiom of specialization, the system would be *unsound*, in that it would be possible to derive non-valid sentences. The rule of *modus ponens* clearly preserves validity. So all we need to establish is that the *finitary* version of universal generalization works, that is, that if α_n^x is valid for every name in α and at least one not in α, then $\forall x\alpha$ is valid too. In the case of $\mathcal{L}$, this was Corollary 2.8.11 of Theorem 2.8.8; here we need a similar corollary for a generalized theorem:

Theorem 4.4.2: *Let * be a bijection from names to names. For any term t or wff α, let t^* or α^* be the result of simultaneously replacing in t or α every name by its mapping under *. Then α is valid iff α^* is valid.*

Proof: Similar to the proof of Theorem 2.8.8. We need to define w^* as before, and here we also need to define e^* as $\{w^* \mid w \in e\}$. ∎

Corollary 4.4.3: *Let α have a single free variable x and let n be a standard name not appearing in α. Let $n_1, \ldots, n_k$ be the all the standard names appearing in α. If α_n^x is valid and all the $\alpha_{n_i}^x$ are valid, then so is $\forall x\alpha$.*

Proof: The same argument as that of Corollary 2.8.11. ∎

Establishing the *completeness* of this axiom system, that is, that the above axioms are sufficient to generate *all* the valid sentences is much more challenging, as it requires examining the properties of $\mathcal{KL}$ in fine detail. Before doing so, it is worth looking at some simple derivations.

To help in the presentation of derivations, we will use the following property of the proof theory of $\mathcal{KL}$:

Theorem 4.4.4: *If* $\vdash \alpha$, *then* $\vdash \boldsymbol{K}\alpha$.

Proof: The proof is by induction on the length of the derivation of α. First suppose that α is an axiom of $\mathcal{KL}$. There are two cases: if it is an instance of an axiom of $\mathcal{L}$ (with proviso), then $\boldsymbol{K}\alpha$ is also an axiom of $\mathcal{KL}$ and so is derivable; all other axioms of $\mathcal{KL}$ are subjective, and so if σ is any other axiom, $(\sigma \supset \boldsymbol{K}\sigma)$ is also an axiom, in which case $\boldsymbol{K}\sigma$ is again derivable by *modus ponens*. If, on the other hand, α follows from some earlier derivable β and $(\beta \supset \alpha)$, then by induction, $\boldsymbol{K}\beta$ and $\boldsymbol{K}(\beta \supset \alpha)$ must also be derivable, and so $\boldsymbol{K}\alpha$ is derivable by *modus ponens* and the axiom of closure of knowledge under *modus ponens*. Finally, if α is of the form $\forall x\beta$, and is derivable from $\beta^x_{n_1}$ to $\beta^x_{n_k}$ by universal generalization, then by induction, $\boldsymbol{K}\beta^x_{n_1}$ to $\boldsymbol{K}\beta^x_{n_k}$ are also derivable, in which case, $\forall x\, \boldsymbol{K}\beta$ follows from universal generalization, and then $\boldsymbol{K}\forall x\beta$, by *modus ponens* and the axiom of closure of knowledge under universal generalization. ∎

So although as an axiom we only state that the axioms of $\mathcal{L}$ are known, this theorem shows that any derivable sentence of $\mathcal{KL}$ is also known. This means that the proof theory behaves as if there was an additional rule of inference (sometimes called *knowledge generalization*) which says: from α, infer $\boldsymbol{K}\alpha$.

We will use the same notation for derivations as we did with $\mathcal{L}$, with two additions. First, a justification marked $\boldsymbol{\mathcal{L}}$ means that the current line is derivable from the previous one (and perhaps earlier ones too), as a theorem of $\mathcal{L}$ alone. In other words, we will not go into any detail involving sub-derivations that use only the axioms of $\mathcal{L}$. Second, a justification **KG** means that the current line is formed by putting a $\boldsymbol{K}$ in front of an earlier line (appealing to the above theorem).

Figure 4.2 contains a derivation of $\boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$, whose validity was proven directly in the previous section. The last step is derived using properties of $\mathcal{L}$, from steps 3 and 8: if $(\beta \supset \gamma)$ and $(\neg\beta \supset \gamma)$ are both derivable, then so is γ.

As a second example, consider the fact that subjective knowledge must be accurate when knowledge is consistent, that is, that

$$\vdash \neg\boldsymbol{K}\alpha \supset (\boldsymbol{K}\sigma \supset \sigma).$$

This is easily shown (and left as an exercise) given the following:

Theorem 4.4.5: $\vdash \neg\boldsymbol{K}\alpha \supset (\boldsymbol{K}\beta \supset \neg\boldsymbol{K}\neg\beta)$.

Proof: See Figure 4.3. ∎

1.	$\alpha \supset (\boldsymbol{K}\alpha \supset \alpha)$	$\mathcal{L}$
2.	$\boldsymbol{K}(\alpha \supset (\boldsymbol{K}\alpha \supset \alpha))$	**KG**
3.	$\boldsymbol{K}\alpha \supset \boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$	**MP**
4.	$\neg\boldsymbol{K}\alpha \supset (\boldsymbol{K}\alpha \supset \alpha)$	$\mathcal{L}$
5.	$\boldsymbol{K}(\neg\boldsymbol{K}\alpha \supset (\boldsymbol{K}\alpha \supset \alpha))$	**KG**
6.	$\boldsymbol{K}\neg\boldsymbol{K}\alpha \supset \boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$	**MP**
7.	$\neg\boldsymbol{K}\alpha \supset \boldsymbol{K}\neg\boldsymbol{K}\alpha$	**Ax**
8.	$\neg\boldsymbol{K}\alpha \supset \boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$	**MP**
9.	$\boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$	$\mathcal{L}$

Figure 4.2: A derivation in $\mathcal{KL}$

1.	$\beta \supset (\neg\beta \supset \alpha)$	$\mathcal{L}$
2.	$\boldsymbol{K}(\beta \supset (\neg\beta \supset \alpha))$	**KG**
3.	$\boldsymbol{K}\beta \supset (\boldsymbol{K}\neg\beta \supset \boldsymbol{K}\alpha)$	**MP**
4.	$\neg\boldsymbol{K}\alpha \supset (\boldsymbol{K}\beta \supset \neg\boldsymbol{K}\neg\beta)$	$\mathcal{L}$

Figure 4.3: Derivation of not knowing a wff and its negation

So, as long as there is a single sentence α that is not known, there is no sentence β such that both it and its negation are known. The proviso is necessary since it is possible for every sentence to be known.

As a third example, we derive $(\forall x \forall y(x = y) \supset \boldsymbol{K}(x = y))$ in Figure 4.4. This shows that all equalities (among standard names) are known. A similar derivation can be used to show that inequalities are also known. Note that the first two lines here use the fact that equality sentences that do not use function symbols are subjective, and hence known. So this does *not* permit the derivation of $((t_1 = t_2) \supset \boldsymbol{K}(t_1 = t_2))$ for non-standard terms t_i, since the axiom of specialization with proviso cannot put a function symbol within the scope of a $\boldsymbol{K}$.

4.5 A Completeness proof

We now turn our attention to the completeness of the axiomatization of $\mathcal{KL}$:

Theorem 4.5.1: *If a sentence of $\mathcal{KL}$ is valid, then it is derivable.*

1.	$(^{\#}1 = {}^{\#}1) \supset \boldsymbol{K}(^{\#}1 = {}^{\#}1)$	**Ax**
2.	$(^{\#}1 = {}^{\#}2) \supset \boldsymbol{K}(^{\#}1 = {}^{\#}2)$	**Ax**
3.	$\forall y(^{\#}1 = y) \supset \boldsymbol{K}(^{\#}1 = y)$	**UG**
4.	$\forall x \forall y(x = y) \supset \boldsymbol{K}(x = y)$	**UG**

Figure 4.4: Derivation of knowing equality of names

As we discussed in the case of $\mathcal{L}$, we can prove this by showing that any consistent sentence is satisfiable. This is sufficient since if a sentence α is valid, $\neg\alpha$ is unsatisfiable, and so, $\neg\alpha$ would be inconsistent, and therefore, $\neg\neg\alpha$ derivable, and consequently α derivable as well, since (as it is easy to show) $(\neg\neg\alpha \supset \alpha)$ is derivable in $\mathcal{K}\mathcal{L}$ (and $\mathcal{L}$).

To show that every consistent sentence is satisfiable, we proceed in two stages: first we show that every finite consistent set of sentences can be extended to what we will call a T-set; then we show that every T-set can be satisfied.

To define a T-set, we start with the notion of a maximally consistent set: a set of sentences is *maximally consistent* iff it is consistent and any proper superset is inconsistent. The following are properties of maximally consistent sets that derive directly from properties of ordinary first-order logic and we will not prove here:

Lemma 4.5.2:

1. *Every consistent set can be extended to a maximally consistent set. That is, for every consistent* Γ, *there is a maximally consistent* Γ' *such that* $\Gamma \subseteq \Gamma'$.
2. *If* Γ *is maximally consistent then* $\neg\alpha \in \Gamma$ *iff* $\alpha \notin \Gamma$.
3. *If* Γ *is maximally consistent then* $(\alpha \wedge \beta) \in \Gamma$ *iff* $\alpha \in \Gamma$ *and* $\beta \in \Gamma$.
4. *If* Γ *is maximally consistent and* $\Gamma \vdash \alpha$, *then* $\alpha \in \Gamma$.

Note that we are *not* claiming for a maximally consistent Γ that if $\exists x\alpha \in \Gamma$, that for some n, $\alpha_n^x \in \Gamma$. In fact, the set

$$\{\exists x\, P(x),\ \neg P(^{\#}1),\ \neg P(^{\#}2),\ \ldots\}$$

is consistent (and can be extended to a maximally consistent set) since there is no contradiction for any finite subset of the set. Similarly, the infinite set

$$\{\neg\boldsymbol{K}\forall x\, P(x),\ \boldsymbol{K}P(^{\#}1),\ \boldsymbol{K}P(^{\#}2),\ \ldots\}$$

is consistent as is

$$\{(t \neq {}^{\#}1),\ (t \neq {}^{\#}2),\ (t \neq {}^{\#}3),\ \ldots\}.$$

However, none of these sets are satisfiable, and so a T-set must go beyond maximal consistency if it is to be satisfiable.

To handle these cases, we first define the concept of an *E-form*: the E-forms with respect to a variable x are the least set of wffs with only x free such that:

1. If α is a formula with just a free variable x, then $(\exists x\alpha \supset \alpha)$ is an E-form with respect to x;
2. If t is a closed term, then $(t = x)$ is an E-form with respect to x;
3. If α is any sentence and β is an E-form with respect to x, then so is the formula $(\neg \boldsymbol{K}\alpha \supset \neg \boldsymbol{K}(\beta \supset \alpha))$.

A substitution α^x_n, where α is an E-form with respect to x and n is any standard name is called an *instance* of the E-form. Now we define a *T-set* to be a maximally consistent set that contains at least one instance of every E-form.

Notice how T-sets rule out cases like the above: for example, if a T-set contains $\exists x P(x)$, then for some n it must also contain $(\exists x P(x) \supset P(n))$, and thus, it must also contain $P(n)$ by Lemma 4.5.2. That is, if a T-set contains $\exists x\alpha$, it must also contain a witness to this existential.

4.5.1 Part 1

In this subsection, we prove

Theorem 4.5.3: *Every finite consistent set can be extended to a T-set.*

We begin by showing that the existential closure of every E-form is derivable in $\mathcal{KL}$:

Lemma 4.5.4: *If α is an E-form wrt x, then $\vdash \exists x\alpha$.*

Proof: The proof is by induction on the composition of the E-form. If the E-form is one of the two base cases, then the lemma holds by virtue of properties of $\mathcal{L}$, and were given as exercises in Chapter 2. Otherwise assume that α is any sentence, β is an E-form wrt x, and that by induction, $\exists x\beta$ is derivable. It is easy to show that $(\forall x \boldsymbol{K}\gamma \supset \boldsymbol{K}\exists x\gamma)$ is derivable for any γ, and in particular,

$$\forall x \boldsymbol{K}(\beta \supset \alpha) \supset \boldsymbol{K}\exists x(\beta \supset \alpha)$$

is derivable. By properties of $\mathcal{L}$,

$$\boldsymbol{K}\exists x(\beta \supset \alpha) \supset \boldsymbol{K}((\exists x\beta) \supset \alpha)$$

is also derivable, since x does not appear free in α. However, $\vdash \boldsymbol{K}\exists x\beta$ since $\vdash \exists x\beta$, by

Theorem 4.4.4. So putting all these together, we get that

$$\vdash \forall x \boldsymbol{K}(\beta \supset \alpha) \supset \boldsymbol{K}\alpha,$$

and so

$$\vdash \neg\boldsymbol{K}\alpha \supset \exists x \neg\boldsymbol{K}(\beta \supset \alpha).$$

Finally, using properties of $\mathcal{L}$, we can move the existential to the front, and get that

$$\vdash \exists x.\neg\boldsymbol{K}\alpha \supset \neg\boldsymbol{K}(\beta \supset \alpha),$$

since x does not occur free in α. ∎

Next we have that

Lemma 4.5.5: *If Γ is a finite consistent set of sentences, and β is an E-form, then there is a name n such that $\Gamma \cup \{\beta_n^x\}$ is consistent.*

Proof: Suppose not. Let γ be the conjunction of sentences in Γ. Then,

$$\vdash (\beta_n^x \supset \neg\gamma),$$

for every n, and so,

$$\vdash \forall x(\beta \supset \neg\gamma).$$

But, by the previous lemma, $\vdash \exists x\beta$. Therefore, since x does not occur free in γ, we get $\vdash \neg\gamma$, contradicting the consistency of Γ. ∎

We can now prove the theorem of this subsection:

Proof: Suppose all sentences of $\mathcal{KL}$ are enumerated by $\alpha_1, \alpha_2, \alpha_3, \ldots$ and that all E-forms are enumerated by $\beta_1, \beta_2, \beta_3, \ldots$. We will first define a sequence of finite sets of sentences, $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$ and show that each must be consistent. First, let Γ_0 be the given finite consistent set of sentences. Now assume that Γ_i has been defined and is consistent. Let α be α_i if $\Gamma_i \cup \{\alpha_i\}$ is consistent, and $\neg\alpha_i$, otherwise; then $\Gamma_i \cup \{\alpha\}$ is consistent. Let β be the instance of the E-form β_i that is consistent with $\Gamma_i \cup \{\alpha\}$, promised by the previous lemma, and let Γ_{i+1} be $\Gamma_i \cup \{\alpha, \beta\}$. This set must be consistent also. Finally, let Γ be the union of all Γ_i. This set is maximally consistent and also contains an instance of every E-form. ∎

This shows that any finite consistent set can be extended to a T-set.

4.5.2 Part 2

What remains to be shown is this:

Theorem 4.5.6: *Every T-set can be satisfied.*

In fact, what we will show is that a T-set completely determines an epistemic and world state, that is, that for each T-set Γ, there is an e and w such that

$$\Gamma = \{\gamma \mid e, w \models \gamma\}.$$

As we will show, from the fact that a T-set is maximally consistent, negations and conjunctions are handled properly; because a T-set also has an instance of every E-form, existentials are also accounted for. So all we really need to do is establish that the $\boldsymbol{K}$ operator is treated properly.

In what follows, we let $\Re(\Gamma)$ be the set of all T-sets Γ' such that for every α, if $\boldsymbol{K}\alpha \in \Gamma$, then $\alpha \in \Gamma'$.

First we define a mapping from T-sets to world states: for any Γ that is a T-set, w_Γ is the world state w such that for any primitive, $w[\phi] = 1$ iff $\phi \in \Gamma$, and $w[t] = n$ iff $(t = n) \in \Gamma$. From the properties of T-sets, we get by induction:

Lemma 4.5.7: *If Γ is a T-set, then for any objective ϕ, $\phi \in \Gamma$ iff $w_\Gamma \models \phi$.*

Note that this handles the completeness for the objective part of $\mathcal{L}$. To handle the rest of $\mathcal{KL}$, first we show:

Lemma 4.5.8: *If Γ is a T-set, and $\neg\boldsymbol{K}\alpha \in \Gamma$, then for some $\Gamma' \in \Re(\Gamma)$, $\neg\alpha \in \Gamma$.*

Proof: We will show that there must be a Γ' that has these properties: it contains $\neg\alpha$, it contains an instance of every E-form, it contains every γ such that $\boldsymbol{K}\gamma \in \Gamma$, and it is consistent. It is then immediate that this Γ' can be extended to a maximally consistent set, which is therefore a member of $\Re(\Gamma)$.

First observe, that since $\neg\boldsymbol{K}\alpha \in \Gamma$, and Γ is a T-set, every E-form β has an instance β_n^x such that $\neg\boldsymbol{K}(\beta_n^x \supset \alpha) \in \Gamma$. Let $\beta_1, \beta_2, \ldots$ and so on, be all such instances, and let Γ' be this set, together with $\neg\alpha$ and all γ such that $\boldsymbol{K}\gamma \in \Gamma$. What remains is to show that this Γ' is consistent.

Observe that for any finite subset $\{\beta_1, \ldots, \beta_k\}$ of the instances of E-forms in Γ', we have the sentence

$$\neg\boldsymbol{K}(\beta_1 \supset (\beta_2 \supset \ldots \supset \alpha)\ldots)$$

in Γ. This is by induction on the size of the subset, using the closure property of T-sets. Now suppose to the contrary that Γ' is inconsistent. Then for some γ such that $\boldsymbol{K}\gamma \in \Gamma$, and some finite set of β_i as above, we have that

$$\vdash (\gamma \supset (\beta_1 \supset (\beta_2 \supset \ldots \supset \alpha)\ldots)),$$

and thus,

$$\vdash (\boldsymbol{K}\gamma \supset \boldsymbol{K}(\beta_1 \supset (\beta_2 \supset \ldots \supset \alpha)\ldots)).$$

Since $\boldsymbol{K}\gamma \in \Gamma$, this would imply that

$$\boldsymbol{K}(\beta_1 \supset (\beta_2 \supset \ldots \supset \alpha)\ldots)$$

was in Γ too, contradicting the consistency of Γ itself. ∎

Next, we associate an epistemic state to each T-set as follows: for any Γ that is a T-set, let e_Γ be defined as

$$\{w_{\Gamma'} \mid \Gamma' \in \Re(\Gamma)\}.$$

Now we are ready to prove the theorem of this subsection. Specifically, we prove that if Γ is a T-set, and $w = w_\Gamma$ and $e = e_\Gamma$ then $\alpha \in \Gamma$ iff $e, w \models \alpha$, and consequently Γ is satisfied by this w and e.

Proof: The proof is by induction on the length of α. If α is a atomic sentence or an equality, the theorem holds by Lemma 4.5.7. The theorem holds for negations and conjunctions by induction, and for existential quantification by induction and the properties of T-sets. Finally, consider the case of $\boldsymbol{K}\alpha$. If $\boldsymbol{K}\alpha \in \Gamma$, then for every every $\Gamma' \in \Re(\Gamma)$, $\alpha \in \Gamma'$; thus, by induction, for every $w' \in e$, $e, w' \models \alpha$, and so $e \models \boldsymbol{K}\alpha$. Conversely, if $\boldsymbol{K}\alpha \notin \Gamma$, then $\neg\boldsymbol{K}\alpha \in \Gamma$, and so by Lemma 4.5.8, for some $\Gamma' \in \Re(\Gamma)$, $\alpha \notin \Gamma'$; thus, by induction, for some $w' \in e$, $e, w' \models \neg\alpha$, and so $e \models \neg\boldsymbol{K}\alpha$. ∎

This ends the completeness proof.

4.5.3 Variant systems

It is appropriate at this stage to consider some simple variants of $\mathcal{KL}$ and see how the axiomatization and the proof of completeness would have to be modified to deal with them.

Perhaps the simplest variant would be one where knowledge was required to be *consistent*. Currently, the set consisting of all sentences of the form $\boldsymbol{K}\alpha$ is satisfiable, but only by the epistemic state that is the empty set of world states. Semantically, to make sure that knowledge is consistent, we need only require an epistemic state to be non-empty. To capture this property axiomatically, we simply change the axiom stating that subjective knowledge is complete to one stating that it is both complete and accurate:

subjective knowledge is complete and accurate: $(\sigma \equiv \boldsymbol{K}\sigma)$.

This is clearly sound for the new semantics. To show that it is complete, we need only show that for any T-set Γ, the set $\Re(\Gamma)$ is non-empty. To see why it must be, observe that Γ cannot contain every $\boldsymbol{K}\gamma$ since it would have to contain $\boldsymbol{K}\neg\boldsymbol{K}\alpha$, and then $\neg\boldsymbol{K}\alpha$ by the above axiom, violating consistency. Thus it must contain, $\neg\boldsymbol{K}\gamma$ for some γ, and then by Lemma 4.5.8, $\Re(\Gamma)$ is non-empty.

Another simple variant of $\mathcal{KL}$ would require all knowledge to be *accurate*: we only look at pairs $\langle e, w\rangle$ such that $w \in e$. In terms of the proof theory, this can be handled by adding the axiom

knowledge is accurate: $(\boldsymbol{K}\alpha \supset \alpha)$

To see why this is sufficient, we need only show that for any T-set Γ we have that $w_\Gamma \in e_\Gamma$. In fact, we get a stronger property, namely that $\Gamma \in \Re(\Gamma)$, as a direct consequence of the above axiom and T-set closure.

A final variant that is less plausible in general is that the knowledge is *complete*. As we said earlier, this is modeled semantically by having an epistemic state consisting of a single world state. In the proof theory, we would add the following axiom

knowledge is complete: $(\neg\boldsymbol{K}\alpha \supset \boldsymbol{K}\neg\alpha.)$

To see why this is sufficient, we need only show that for any T-set Γ, $\Re(\Gamma)$ consists of a singleton set, which we leave as an exercise.

4.6 Reducibility

Having looked at a proof theory for $\mathcal{KL}$ and a few simple variants, we now turn our attention to a very different logical property of knowledge having to do with meta-knowledge. This will also constitute the first time there is clear difference between the quantifier-free subset of $\mathcal{KL}$ and the full version. We will use the term <u>*propositional*</u> subset to mean the subset of $\mathcal{KL}$ without quantifiers.

If we look at the semantic and axiomatic accounts of $\mathcal{KL}$, it might appear that the simplifying assumption made regarding meta-knowledge makes the whole notion dispensable. Assuming that knowledge is consistent, for example, we have that both $\boldsymbol{K}\boldsymbol{K}\alpha \equiv \boldsymbol{K}\alpha$ and $\boldsymbol{K}\neg\boldsymbol{K}\alpha \equiv \neg\boldsymbol{K}\alpha$ are valid. This means that we can always reduce strings of $\boldsymbol{K}$ operators and negations down to at most a *single* $\boldsymbol{K}$ operator. So the question we wish to address in this section is this: can we generalize this idea and eliminate *all* nesting of $\boldsymbol{K}$ operators, without losing expressive power? In other words, is it possible to take any sentence and find an equivalent one where the $\boldsymbol{K}$ operator only dominates objective sentences? If we can, this would mean that meta-knowledge offers essentially nothing over objective knowledge.

As it turns out, the answer to the question is *yes* for the propositional subset of $\mathcal{KL}$, and *no* for the full language. First the propositional case:

Theorem 4.6.1: *For any α in the propositional part of $\mathcal{KL}$, there is an α', where α' has no nesting of $\boldsymbol{K}$ operators and $\models (\alpha \equiv \alpha')$.*

Proof: The proof is based on induction on the depth of nesting of $\boldsymbol{K}$ operators in α, but

here we will merely present it in outline. Assume that α has a subformula $\boldsymbol{K}\beta$ where β uses $\boldsymbol{K}$ operators. First, we observe that because of the usual properties of the propositional part of $\mathcal{L}$, we can put β into a logically equivalent conjunctive normal form (CNF) β', where β' is a conjunction of disjunctions of extended literals, where an extended literal is a (possibly negated) sentence that is either objective or the form $\boldsymbol{K}\gamma$. So we have that $\boldsymbol{K}\beta$ is equivalent to $\boldsymbol{K}\beta'$. Next, we use the fact that both of these are valid:

$$\boldsymbol{K}(\beta_1 \wedge \beta_2) \;\equiv\; (\boldsymbol{K}\beta_1 \wedge \boldsymbol{K}\beta_2)$$

and

$$\boldsymbol{K}(\phi \vee \boldsymbol{K}\gamma_1 \vee \neg\boldsymbol{K}\gamma_2) \;\equiv\; (\boldsymbol{K}\phi \vee \boldsymbol{K}\gamma_1 \vee \neg\boldsymbol{K}\gamma_2).$$

(See the exercises.) By applying this repeatedly, we get that $\boldsymbol{K}\beta'$ is equivalent to $\boldsymbol{K}\beta''$ where the latter has reduced the level of nesting by one. By applying this repeatedly to α, we eliminate all nesting of $\boldsymbol{K}$ operators. ∎

So this theorem shows that talk of meta-knowledge in the propositional part of $\mathcal{KL}$ can be replaced by completely equivalent talk about objective knowledge. If there is anything *new* to meta-knowledge, it is in its interaction with the quantifiers. Note that the above proof fails for the full version of $\mathcal{KL}$ because there is no way to distribute the $\boldsymbol{K}$ over some version of a CNF: although we can move $\boldsymbol{K}$ operators inwards when we have something like $\boldsymbol{K}\forall x\alpha$, we cannot do so for sentences like $\boldsymbol{K}\exists x\alpha$.

In the full quantified version of $\mathcal{KL}$, we will show that there are indeed sentences with nested $\boldsymbol{K}$ operators that cannot be rephrased in terms of objective knowledge. In particular, the sentence

$$\boldsymbol{K}\exists x[P(x) \wedge \neg\boldsymbol{K}P(x)],$$

which we will call λ, cannot be so reduced:

Theorem 4.6.2: *For any α, if $\models (\alpha \equiv \lambda)$, then α has nested $\boldsymbol{K}$ operators.*

The proof proceeds by constructing two epistemic states e_1 and e_2 that agree on all objective knowledge but disagree on λ. This is sufficient, since any proposed α without nested $\boldsymbol{K}$ operators cannot be equivalent to λ, since although e_1 and e_2 will assign the same truth value to α, they will assign different truth values to λ.

We construct e_1 and e_2 as follows. Let Ω be some infinite set of standard names containing $^{\#}1$ whose complement is also infinite.[3] Let Φ be the set of objective sentences consisting of $\{(t = {}^{\#}1)\}$ for every primitive term t, $\{\neg\phi\}$ for every primitive sentence ϕ whose predicate letter is not P, and finally $\{P(n)\}$ for every $n \in \Omega$. Let e_1 be $\{w \mid w \models \Phi\}$.

3 An example is: $\{{}^{\#}1, {}^{\#}3, {}^{\#}5, \ldots\}$.

Let $\overline{w}$ be the (unique) element of e_1 such that for every $n \notin \Omega$, $w \models \neg P(n)$. Finally, let e_2 be $e_1 \setminus \{\overline{w}\}$. This gives us that for all $w \in e_1$ and for all $n \in \Omega$, $w \models P(n)$; worlds in e_2 have this property also, and in addition, because $\overline{w} \notin e_2$, they also each have $w \models P(n)$ for some $n \notin \Omega$.

The first thing to observe is that e_1 and e_2 disagree on λ. Specifically,

$$e_1 \models \neg \boldsymbol{K} \exists x [P(x) \wedge \neg \boldsymbol{K} P(x)]$$

but

$$e_2 \models \boldsymbol{K} \exists x [P(x) \wedge \neg \boldsymbol{K} P(x)].$$

This is because they do agree on the *known* instances of P, in that

$$e_1 \models \boldsymbol{K} P(n) \text{ iff } e_2 \models \boldsymbol{K} P(n) \text{ iff } n \in \Omega,$$

and so the presence of $\overline{w}$ in e_1 makes λ false, since $\overline{w}$ satisfies $P(n)$ only for the known instances of P.

To complete the proof, we need only show that e_1 and e_2 agree on all objective knowledge and hence on all sentences without nested $\boldsymbol{K}$ operators. Showing that if $e_1 \models \boldsymbol{K}\phi$ then $e_2 \models \boldsymbol{K}\phi$ is trivial, since $e_2 \subset e_1$; the converse will take some work.

First we prove the following:

Lemma 4.6.3: *Let n_1 and n_2 be distinct names that are not members of Ω. Then for any ϕ, $\overline{w} \models \phi$ iff $\overline{w} \models \phi^*$, where ϕ^* is ϕ with n_1 and n_2 interchanged.*

Proof: By induction on the structure of ϕ, given that $\overline{w}$ is defined in a way that treats the two names exactly the same. ∎

Next, assume that the names that are not in Ω are enumerated as $m_1, m_2, m_3, \ldots$, and define a corresponding sequence of worlds $w_1, w_2, w_3, \ldots$, as follows: w_i is the unique element of e_2 such that $w_i \models P(n)$ iff $n \in \Omega$ or $n = m_i$. Then we get the following:

Lemma 4.6.4: *Let n be any name in Ω other than $^{\#}1$. Then for any ϕ, $w_i \models \phi$ iff $w_i \models \phi^*$, where ϕ^* is the result of interchanging n and m_i in ϕ.*

Proof: By induction on the structure of ϕ, given that w_i is defined in a way that treats the two names exactly the same. (The proviso regarding $^{\#}1$ is necessary because we have made it be the value of all primitive terms.) ∎

Using these two lemmas, we obtain:

Lemma 4.6.5: *Suppose $m_i \notin \Omega$. Then for any ϕ which does not mention m_i, $\overline{w} \models \phi$ iff $w_i \models \phi$.*

Proof: The proof is by induction on ϕ. The only tricky case is for existentials.
In one direction, if $\overline{w} \models \exists x\phi$, then $\overline{w} \models \phi^x_n$ for some n. There are two cases: if $n \neq m_i$, we get that $w_i \models \phi^x_n$ by induction, and so $w_i \models \exists x\phi$; however, if $n = m_i$, then by the first lemma above, if we let n' be some distinct name that does not appear in ϕ, and such that $n' \notin \Omega$, we get that $\overline{w} \models \phi^x_{n'}$. Then by induction, $w_i \models \phi^x_{n'}$, and so $w_i \models \exists x\phi$.
In the other direction, if $w_i \models \exists x\phi$, then $w_i \models \phi^x_n$ for some n. Again, there are two cases: if $n \neq m_i$, we get that $\overline{w} \models \exists x\phi$ as above; however, if $n = m_i$, then by the second lemma above, if we choose any n' not mentioned in ϕ such that $n' \in \Omega$ and $n' \neq {}^{\#}1$, we get that $w_i \models \phi^x_{n'}$. Then by induction, we get that $\overline{w} \models \phi^x_{n'}$, and so $\overline{w} \models \exists x\phi$. ∎

Now we can finish the proof of the theorem. If $e_1 \models \neg \boldsymbol{K}\phi$, then for some $w \in e_1$ we have $w \models \neg\phi$. If $w \in e_2$, we are done; otherwise, $w = \overline{w}$, and so choose some $m_i \notin \Omega$ that does not appear in ϕ. By the lemma above, $w_i \models \neg\phi$, where $w_i \in e_2$. Either way, for some $w \in e_2$ we have $w \models \neg\phi$, and so $e_2 \models \neg \boldsymbol{K}\phi$.

In the end, what this theorem shows is that the knowledge expressed by λ cannot be expressed in terms of objective knowledge, even allowing that subjective knowledge is complete and accurate. What the sentence λ expresses is that the KB knows that it has incomplete knowledge about P: there is an individual with property P not currently known to have that property. The above theorem shows that this is a form of knowledge that goes beyond mere objective knowledge.

This completes our purely logical analysis of $\mathcal{KL}$. In the chapters to follow, we will apply $\mathcal{KL}$ to the task of interacting with a knowledge base.

4.7 Bibliographic notes

The properties of $\mathcal{KL}$ were first presented in [111], and then in [113]. Many of the properties discussed here will come up again in later chapters. Logical omniscience was first discussed by Hintikka [65] and, because it appears to have direct bearing on computational issues, has received considerable attention since then. See Chapters 12 and 13 and the references there for a more thorough discussion of this issue and a model of knowledge without logical omniscience. Other properties of the propositional subset of $\mathcal{KL}$ and numerous variants can be found in [42]. Turning to the quantificational aspects, the *de dicto* / *de re* distinction is a major one in quantified modal logics. See [69] for an introduction to the issue. A more philosophical discussion can be found in [127]. The philosopher Quine,

among others, has maintained that quantifying into a modal context is fundamentally incoherent [157], although his arguments (concerning, for instance, confusion of identity) require *de re* belief without using standard names. In this context, our standard names are often called "rigid designators" in that that they designate the same individual in every possible world. Note that we use these as logical names [167], that is, as terms in our logical language, without claiming them to be anything like proper names found in natural languages. See [128, 83] for a discussion of these issues. The Henkin-style completeness proof presented here (including the use of E-forms) is adapted from [69].

4.8 Exercises

1. State whether or not each of the following properties of knowledge holds in general, and if not, whether it holds when the sentence α is subjective and when knowledge is consistent:
   (a) $\models (\alpha \supset \boldsymbol{K}\alpha)$;
   (b) $\models (\boldsymbol{K}\alpha \supset \alpha)$;
   (c) if $\models \alpha$ then $\models \boldsymbol{K}\alpha$;
   (d) if $\models \boldsymbol{K}\alpha$ then $\models \alpha$;
   (e) $\models \boldsymbol{K}(\alpha \supset \boldsymbol{K}\alpha)$;
   (f) $\models \boldsymbol{K}(\boldsymbol{K}\alpha \supset \alpha)$.
2. Show that $\models \boldsymbol{K}\alpha \wedge \boldsymbol{K}(\alpha \supset \beta) \supset \boldsymbol{K}\beta$ and $\models \forall x \boldsymbol{K}\alpha \supset \boldsymbol{K}\forall x\alpha$.
3. Divide the set of subjective sentences into three categories: positive, negative, and mixed. Show for the positive case, we have that $(\sigma \equiv \boldsymbol{K}\sigma)$, even when the knowledge is inconsistent.
4. Use Theorem 4.2.5 to show that a knowledge base will always believe that either it does not believe α or it does not believe $\neg\alpha$. Thus, a knowledge base always believes it is consistent.
5. Show that $\vdash \boldsymbol{K}\alpha \supset \boldsymbol{K}(\alpha \vee \beta)$.
6. Use Theorem 4.4.5 to show $\vdash \neg\boldsymbol{K}\alpha \supset (\boldsymbol{K}\sigma \supset \sigma)$.
7. Show that $\{(t_1 = t_2),\ \boldsymbol{K}(t_1 \neq t_2)\}$ is satisfiable.
8. Show that Theorem 4.3.7 is false without the proviso on introducing a function symbol within the scope of a $\boldsymbol{K}$.
9. Show a derivation of a non-valid sentence that could happen if there were no proviso on the axiom of specialization.

10. Show that the axiom of specialization is valid without proviso when the value of the term t being substituted is correctly known. That is, show that

    $$\models \exists y(y = t) \wedge \boldsymbol{K}(y = t)) \supset (\forall x\alpha \supset \alpha_t^x).$$

    Show that this does not hold when the value of the term t is known but need not be correct. That is, show that

    $$\not\models \exists y \boldsymbol{K}(y = t) \supset (\forall x\alpha \supset \alpha_t^x).$$

11. Show that $(\neg \boldsymbol{K}\alpha \supset \boldsymbol{K}\neg\alpha)$ is sufficient to characterize complete knowledge. That is, show that the given axiomatization of $\mathcal{KL}$ with this axiom added is complete for a semantics where an epistemic state is required to be a singleton set.
12. Show that $\models [\boldsymbol{K}(\alpha \vee \sigma) \equiv \boldsymbol{K}\alpha \vee \sigma]$, when σ is subjective. This is a generalization of the property used in the proof of Theorem 4.6.1.
13. Show that the axiom $\forall x \boldsymbol{K}\alpha \supset \boldsymbol{K}\forall x\alpha$ can be replaced by $\boldsymbol{K}\forall x(\boldsymbol{K}\alpha \supset \alpha)$, in the sense that one is derivable from the other, given the other axioms.

5 The TELL and ASK Operations

In the previous chapter, we examined the properties of the language $\mathcal{KL}$ in detail to develop a clear understanding of when sentences in this language were true or false. In this chapter, we will use the language as a way of communicating with a knowledge base or KB. We will use $\mathcal{KL}$ both to find out what is known and to provide new knowledge. We begin by defining these interaction operations, and examining a few immediate properties. We then illustrate the use of these operations on a larger example KB, emphasizing the power of $\mathcal{KL}$ as an interaction language. In the next chapter, we will examine some of the deeper properties and implications of the definitions presented here.

5.1 Overview

After our somewhat lengthy excursion into the logical properties of $\mathcal{KL}$, it is perhaps worthwhile to briefly review where we stand. What we have, so far, is a logical language $\mathcal{KL}$ together with a precise specification of what it means for sentences in this language to be true or false, and what it means for sentences to be known or unknown, as a function of a given world and epistemic state. A world state here is modeled as a function from primitive expressions to their values, and an epistemic state is modeled as a set of world states.

What we intend to do with this language is use it as a way of interacting with the KB of a knowledge-based system. Roughly, we will find out if something is known by the KB by asking it a question formulated as a sentence in $\mathcal{KL}$. Similarly, we will make something known to the KB by telling it that some sentence of $\mathcal{KL}$ is true. Thus, we envision for now two operations to be performed on a KB:

1. **ASK**$[\alpha, e] \in \{yes, no\}$
   In an epistemic state e, we determine if α is known, by using an **ASK** operation. The result we expect is a simple answer, *yes* or *no*.
2. **TELL**$[\alpha, e] = e'$
   In an epistemic state e, we add information to the KB by performing a **TELL** operation. The result is a new epistemic state, e'.

Note that while the first argument to these operations is a symbolic structure (that is, a sentence of $\mathcal{KL}$), the second argument is an epistemic state, not some symbolic representation of one. For now at least, our approach to these operations will not depend on how what is known is actually represented in a KB.

If we think of a KB as an abstract data type, accessible only in terms of the operations it provides, we need one more operation to give us an initial epistemic state:

3. **INITIAL**$[\ ] = e_0$
   The epistemic state before any **TELL** operations.

The idea is that we imagine the full lifetime of a KB as proceeding through a number of states $e_0, e_1, e_2, \ldots$ in sequence, where e_0 is as above, and for every $i > 0$, there are sentences α_i, such that $e_i = \mathbf{TELL}[\alpha_i, e_{i-1}]$. In any such state, we can use **ASK** to determine what is known.

These three operations together constitute a full functional interface to a KR system as described in Chapter 1. The user of the system has access to the KB only through this fairly narrow interface. The KR system builder's job is to implement these three operations somehow using whatever representational means are appropriate.

In the rest of this chapter, we examine the three operations in more detail.

5.2 The ASK operation

The purpose of the **ASK** operation is clear: ultimately, we want to find out from a knowledge base if a sentence α is true or not. The complication is that a KB will not always be able to answer this, since all it has to go on is what it knows. There are, in fact, four possible outcomes:

1. it might believe that α is true;
2. it might believe that α is false;
3. it might not know whether α is true or false;
4. it might be inconsistent, and believe both that α is true and that it is false.

This is not a result of anything like a multi-valued logic, but simply the result of the different possible epistemic states that can arise. In effect, for various α, each of the following sets of sentences are satisfiable (corresponding to the cases above):

1. $\{\boldsymbol{K}\alpha,\ \neg\boldsymbol{K}\neg\alpha\}$;
2. $\{\neg\boldsymbol{K}\alpha,\ \boldsymbol{K}\neg\alpha\}$;
3. $\{\neg\boldsymbol{K}\alpha,\ \neg\boldsymbol{K}\neg\alpha\}$;
4. $\{\boldsymbol{K}\alpha,\ \boldsymbol{K}\neg\alpha\}$;

At best then, **ASK** can do no more than tell us which of the four mutually exclusive alternatives holds for the current epistemic state.

There is, however, a slightly simpler interaction we can use. Instead of asking whether or not α is *true*, we could ask the KB whether or not α is *known.* In the first and fourth cases above, the KB would answer affirmatively, and in the second and third cases, it would answer negatively. If we then ask if $\neg\alpha$ is known, in the second and fourth case, it would answer affirmatively, and in the first and third, negatively. The net effect of this shift is

that we can still determine which of the four cases holds if we so desire, but by asking two questions: one for α and one for $\neg\alpha$.

Since this decoupling of the positive and negative cases leads to a simpler definition for **ASK**, we will use this convention. The translation to the other form of question is straightforward. With this in mind, we define $\underline{ASK}$ as follows:

Definition 5.2.1: For any sentence α of $\mathcal{KL}$ and any epistemic state e,

$$\mathbf{ASK}[\alpha, e] = \begin{cases} \mathit{yes} & \text{if } e \models \boldsymbol{K}\alpha \\ \mathit{no} & \text{otherwise} \end{cases}$$

So at a given epistemic state e, **ASK** returns *yes* not if α is true (which would require a world state), but if $\boldsymbol{K}\alpha$ is true at e. Thus the semantics of $\mathcal{KL}$, which tells us what it means for a sentence like $\boldsymbol{K}\alpha$ to be true, does all the work in the definition.

5.3 The initial epistemic state: e_0

As discussed in Chapter 3 and will become clearer when we look at **TELL**, an epistemic state is a set of world states that is progressively narrowed as information is acquired. In finding out that ϕ is true, we eliminate from the epistemic state the world states where ϕ is false. So the smaller the set of world states, the more complete the knowledge.

It is worth considering the properties of the most uninformed epistemic state, which we will call $\underline{e_0}$. This state consists of *all* world states, in that nothing is known that could eliminate any possible world. Of course, all of the valid sentences will be known in e_0. Furthermore, these are the only objective sentences that are known:

Theorem 5.3.1: *If ϕ is objective, then* $\mathbf{ASK}[\phi, e_0] = \mathit{yes}$ *iff* $\models \phi$.

So nothing is known about the world in e_0, in that any objective sentence that is false in some world state is not known to be true.

What other knowledge does e_0 have? Like any other epistemic state, it will know about its own subjective state. So for example, if $\neg\phi$ is satisfiable, then $e_0 \models \neg\boldsymbol{K}\phi$, and so $e_0 \models \boldsymbol{K}\neg\boldsymbol{K}\phi$, which means that $\mathbf{ASK}[\neg\boldsymbol{K}\phi, e_0]$ is *yes*.

So like every other epistemic state, e_0 knows all the valid sentences as well as all the true subjective ones. Somewhat surprisingly, perhaps, still more is known:

Theorem 5.3.2: *If α is* $[\exists x\, P(x) \supset \exists x.P(x) \wedge \neg\boldsymbol{K}P(x)]$, *then* $\mathbf{ASK}[\alpha, e_0] = \mathit{yes}$.

Proof: First observe that $e_0 \models \neg \boldsymbol{K} P(n)$, for every n. Thus, for every $w \in e$, if $w \models \exists x P(x)$, then $e_0, w \models \exists x[P(x) \wedge \neg \boldsymbol{K} P(x)]$. From this it follows that $e_0 \models \boldsymbol{K}\alpha$. ∎

Note that this α is neither a valid sentence nor a true subjective one. What α says is that if there is a P, then it must be an unknown P. Intuitively, this sentence is known in e_0 because although it is not known in e_0 whether or not there are any instances of P, it is known that there no *known* instances, and consequently, any instance of P must be an unknown one.

So although e_0 has no purely objective knowledge, it does have non-trivial knowledge, something perhaps not immediately obvious given the logical analysis of $\mathcal{KL}$ in the previous chapter. In fact, we have:

Theorem 5.3.3: *The set* $\{\alpha \mid \mathbf{ASK}[\alpha, e_0] = \mathit{yes}\}$ *is not recursively enumerable.*

Proof: It is a property of ordinary first-order logic (and consequently, of $\mathcal{L}$) that the set of all objective satisfiable sentences is not recursively enumerable. But observe that $\mathbf{ASK}[\neg \boldsymbol{K} \neg \phi, e_0] = \mathit{yes}$ iff ϕ is satisfiable. ∎

This shows the considerable power assumed under our concept of knowledge, and is yet another reason for wanting to look at more limited versions in Chapters 12 and 13.

5.4 The monotonicity of knowledge

As described above, acquiring knowledge means moving from an epistemic state modeled by a set of world states to a subset of those world states. It follows that objective knowledge is preserved by this acquisition of knowledge:

Theorem 5.4.1: *If* ϕ *is objective and* $\mathbf{ASK}[\phi, e] = \mathit{yes}$, *then for any epistemic state* $e' \subseteq e$, $\mathbf{ASK}[\phi, e'] = \mathit{yes}$.

So as information is acquired, anything objective that was believed will continue to be believed. We refer to this property of knowledge as *objective monotonicity*.[1]

However, this monotonicity does *not* hold in general for arbitrary sentences of $\mathcal{KL}$. That is to say, there is no guarantee that once a sentence is believed it will continue to be believed as further information is acquired:

1 The idea of acquiring knowledge to *revise* existing objective knowledge is much more complex and is discussed in the bibliographic notes.

Theorem 5.4.2: *Knowledge is nonmonotonic: there is a sentence α and an epistemic state e such that* $\mathbf{ASK}[\alpha, e] = yes$, *but* $\mathbf{ASK}[\alpha, e'] = no$, *for some* $e' \subset e$.

Proof: Let α be the sentence $\neg \boldsymbol{K} P(n)$, for some standard name n. Since $e_0 \models \alpha$, we have that $\mathbf{ASK}[\alpha, e_0] = yes$. In other words, the initial epistemic state e_0 knows that it does not believe that $P(n)$. Now suppose we eliminate all world states where $P(n)$ is false: let e' be the set of all worlds states w such that $w \models P(n)$. Clearly, $e' \models \boldsymbol{K} P(n)$, and in particular, $e' \models \neg \boldsymbol{K} \alpha$, and so $\mathbf{ASK}[\alpha, e'] = no$. ∎

So a knowledge base may know that it does not believe some sentence, but if it later comes to believe that sentence, it needs to revise its original subjective belief. Thus, a knowledge base will change its mind about certain things even though it is acquiring objective knowledge purely monotonically.

5.5 The TELL operation

It is the **TELL** operation that defines how knowledge is acquired. What we are after with **TELL** is perhaps the simplest form of information acquisition, where there is no revision of objective knowledge, no retraction, and no forgetting. We assert using **TELL** that a sentence is true; if this assertion is inconsistent with what is already known, we simply move into the inconsistent epistemic state, and leave it at that.

After some objective ϕ has been asserted, we would expect that in the resulting state, ϕ should be believed, as should anything previously known about the world. Moreover, we want to be in a state where this is *all* that is known. That is, we should not get new objective beliefs that are independent of ϕ and what was known before. Consequently, on being told ϕ in some state e, we want the *largest* (that is the least informed) epistemic state where ϕ is believed and which is a subset of e. When ϕ is objective, there is a unique epistemic state that has these properties:

$$\mathbf{TELL}[\phi, e] = e \cap \{w \mid w \models \phi.\}$$

Thus, as we have been assuming, asserting ϕ simply means eliminating the world states where ϕ is false.

Consider now a non-objective sentence, α, which is $\exists x[P(x) \wedge \neg \boldsymbol{K} P(x)]$. This sentence says that there is an unknown P, and would be true (at w and e) if the instances of P in w contained some individual that was not among the known instances in e. What could it mean for a KB to be *told* that this sentence is true? There are at least two possible interpretations:

- one might interpret this is as asserting that there is a *forever* unknown P. In other

words, there is a P that is not known to be P and will continue to be unknown, no matter what further knowledge is acquired.

- one might interpret this as asserting that there is a *currently* unknown P. In other words, there is a P that is not among the known P, in the current epistemic state. But as this state changes, this P may become known.

Clearly both readings have their uses: if we want to state properties about this and future states of knowledge, the first reading is more appropriate; if we want to state properties about the world, but in terms of the current epistemic state, the second reading is preferred. Our focus here is clearly on objective knowledge, and so we will take the second reading, although the issue of constraining (future) states of knowledge more generally will reappear in Chapter 8.

The impact of this decision is that on being told a sentence that contains $\boldsymbol{K}$ operators, we can interpret references to what is known to be about the current epistemic state. We do not have to worry about the epistemic state that will result after the update, or any other future state. Consequently, we can do what we did for objective assertions, except we now use the current epistemic state to deal with the non-objective parts. Thus, we define *TELL* as follows:

Definition 5.5.1: For any sentence α of $\mathcal{KL}$ and any epistemic state e,

$$\mathbf{TELL}[\alpha, e] = e \cap \{w \mid e, w \models \alpha\}.$$

By interpreting $\boldsymbol{K}$ operators in α in terms of the given epistemic state (the e argument), we do not need a complex fixed-point construction, and we are guaranteed that the resulting epistemic state is always uniquely defined, as it was in the objective case.

For example, consider $\mathbf{TELL}[\exists x.P(x) \wedge \boldsymbol{K}Q(x), e]$. As an assertion this tells the system that there exists something that has property P. Moreover, this something is also known to have property Q. But known when? The answer we take here is that the individual is known to have property Q in the epistemic state e just before the assertion. In other words, we are asserting that there is an individual n such that $P(n)$ holds and such that $e \models \boldsymbol{K}Q(n)$.

The reason we need to be careful about what epistemic state we use is that as a result of the assertion, we may be changing whether or not an individual is known to have some property. Its status before and after the assertion can be different. Because of this, it can happen that a sentence α is not believed after it has been asserted:

Theorem 5.5.2: *There is a sentence α and an epistemic state e such that*

$$\mathbf{ASK}[\alpha, \mathbf{TELL}[\alpha, e]] = no.$$

Proof: Let e be e_0 and let α be $[P(n) \wedge \exists x.P(x) \wedge \neg \boldsymbol{K} P(x)]$, for some standard name n. If we let $e' = \mathbf{TELL}[\alpha, e_0]$, then $e' = \{w \mid w \models P(n)\}$. But e' has no reason to believe that there is another P apart from n, and so $e' \models \neg \boldsymbol{K} \exists x.P(x) \wedge \neg \boldsymbol{K} P(x)$, from which it follows that $\mathbf{ASK}[\alpha, e'] = no$. ∎

So by the above definition, a knowledge base can be told something and yet end up not believing it. This means that a **TELL** operation cannot be interpreted as an instruction to believe its sentence argument. Rather, it needs to be understood as an assertion that this argument is true (or more precisely, that it *was* true at the time of the assertion), and an instruction to draw conclusions about the world from this fact. Because of this, an assertion of a purely subjective sentence (which carries no information about the world) is either redundant or contradictory:

Theorem 5.5.3: *If σ is subjective, then* $\mathbf{TELL}[\sigma, e] = e$ *or* $\mathbf{TELL}[\sigma, e] = \{\}$.

It is not too surprising that we cannot tell the knowledge base anything about what it knows, since we have assumed that it already has complete knowledge of its own subjective state. In fact, this last result generalizes to any sentence whose truth value is known:

Theorem 5.5.4: *If* $\mathbf{ASK}[\alpha, e] = yes$, *then* $\mathbf{TELL}[\alpha, e] = e$; *if* $\mathbf{ASK}[\neg\alpha, e] = yes$, *then* $\mathbf{TELL}[\alpha, e] = \{\}$.

So, for example, if we start in e_0, we get that $\mathbf{TELL}[\boldsymbol{K} P(c) \vee \boldsymbol{K} Q(c), e_0] = \{\}$. That is, if we tell the system that it either knows $P(c)$ or it knows $Q(c)$, this is inconsistent because it already knows that it does not know either. On the other hand, a similar assertion, $\mathbf{TELL}[P(c) \vee Q(c), e_0]$ is fine, since the objective sentence here is unknown.

5.6 Closed world assertions

What is the purpose of non-objective assertions? The answer is that they allow us to express facts about the world that could otherwise not be made without knowing the contents of the knowledge base. A simple example of this is what has been called the *closed world assumption*. The idea, roughly, is to be able to tell a knowledge base in a certain epistemic state that its information about some part of the world is complete. In its simplest form, we would like to be able tell the system that it already knows every instance of a predicate P. For this, we use a sentence of $\mathcal{KL}$ like

$$\forall x[P(x) \supset \boldsymbol{K} P(x)],$$

which we will call γ.[2] This sentence can be read as saying that every P is currently known, or equivalently, anything not currently known to be a P is not one. So telling the system γ is like telling it $\neg P(n)$, for every n such that $e \models \neg \boldsymbol{K} P(n)$.

For example, suppose that

$$e = \{w \mid w \models P(^{\#}1) \text{ and } w \models P(^{\#}2)\}.$$

For this epistemic state, we have that $e \models \boldsymbol{K}[P(^{\#}1) \wedge P(^{\#}2)]$. That is, in e, it is known that $P(^{\#}1)$ and $P(^{\#}2)$ are both true. But it is not known in e whether or not there are any other n such that $P(n)$ is true. In other words, there are world states in e where $^{\#}1$ and $^{\#}2$ are the only instances of P, and there are world states in e where there are others. So we have:

$$e \models \neg \boldsymbol{K} \forall x[P(x) \equiv (x = {}^{\#}1) \vee (x = {}^{\#}2)]$$

and

$$e \models \neg \boldsymbol{K} \neg \forall x[P(x) \equiv (x = {}^{\#}1) \vee (x = {}^{\#}2)].$$

However, if we now assert γ and get $e' = \mathbf{TELL}[\gamma, e]$, then we have

$$e' \models \boldsymbol{K} \forall x[P(x) \equiv (x = {}^{\#}1) \vee (x = {}^{\#}2)].$$

We have told the system that $^{\#}1$ and $^{\#}2$ are the *only* instances of P. The important point is that using γ we told it this without having to list the instances of P. We were able to do this because in e, we have the following:

$$e \models \boldsymbol{K} \forall x[\boldsymbol{K} P(x) \equiv (x = {}^{\#}1) \vee (x = {}^{\#}2)].$$

In other words, although the system does not know if there are any additional instances of P, it does know what all the *known* instances are. So by telling it γ, it is then able to determine that these known instances are the complete list.

As a second example, suppose that we have

$$e = \{w \mid w \models P(n), \text{ for every } n \neq {}^{\#}1\}.$$

In this epistemic state, $P(n)$ is known to be true for every n apart from $^{\#}1$. However, $P(^{\#}1)$ is undecided since there are world states in e where $P(^{\#}1)$ is true and others where it is false. If we now assign $e' = \mathbf{TELL}[\gamma, e]$, we settle the issue:

$$e' \models \boldsymbol{K} \forall x[P(x) \equiv x \neq {}^{\#}1].$$

In this case, we could not have listed explicitly all of the instances of P even if we had wanted to since there are infinitely many. But γ works as intended here as well.

As a final example, suppose we have

$$e = \{w \mid w \models P(^{\#}1) \text{ or } w \models P(^{\#}2)\}.$$

In this state, we have all world states that satisfy either $P(^{\#}1)$ or $P(^{\#}2)$. So an instance of P is known to exist, $e \models \boldsymbol{K} \exists x P(x)$, although there are no known instances, since for every n, there is a world state in e satisfying $\neg P(n)$. If we now define $e' = \mathbf{TELL}[\gamma, e]$, we get

2 Note that the formula λ introduced on Page 74 is actually $\boldsymbol{K} \neg \gamma$.

the result that e' is empty, the inconsistent epistemic state. The reason for this is that in e, γ is already known to be false. In other words, $e \models \boldsymbol{K}\exists x[P(x) \wedge \neg \boldsymbol{K}P(x)]$, because $e \models \boldsymbol{K}\neg \boldsymbol{K}P(^{\#}1)$ and $e \models \boldsymbol{K}\neg \boldsymbol{K}P(^{\#}2)$, but $e \models \boldsymbol{K}[P(^{\#}1) \vee P(^{\#}2)]$. Thus, in this example, it is already known that there is an instance of P apart from the known ones, and any attempt to assert otherwise leads to an inconsistency.

So although we can express the closed world assumption as a sentence of $\mathcal{KL}$, we must be careful in how it is used. In certain states, the knowledge base will already know that it is missing an instance of the predicate in question and should not be told otherwise. However, we can prove that in all other cases, **TELL** will behave properly and result in a consistent epistemic state where the closed world assumption is known to be true. To show this, we need the following lemma:

Lemma 5.6.1: *Suppose e is an epistemic state where $e \models \neg \boldsymbol{K}\neg\gamma$. Let $e' = \mathbf{TELL}[\gamma, e]$. Then for any w, e, $w \models \gamma$ iff e', $w \models \gamma$.*

Proof: We will show that for any n, $e \models \boldsymbol{K}P(n)$ iff $e' \models \boldsymbol{K}P(n)$, from which the conclusion follows. If $e \models \boldsymbol{K}P(n)$, then $e' \models \boldsymbol{K}P(n)$, since $e' \subseteq e$. Conversely, if $e \models \neg \boldsymbol{K}P(n)$, then since $e \models \neg \boldsymbol{K}\neg\gamma$, there is a $w \in e$ such that $e, w \models \gamma$, and thus for which, $w \models \neg P(n)$. Since $e, w \models \gamma$, we have that $w \in e'$, and so, $e' \models \neg \boldsymbol{K}P(n)$. ∎

Then we get:

Theorem 5.6.2: *Suppose e is an epistemic state where $e \models \neg \boldsymbol{K}\neg\gamma$. Then*

$$\mathbf{ASK}[\gamma, \mathbf{TELL}[\gamma, e]] = \mathit{yes}.$$

Proof: Let $e' = \mathbf{TELL}[\gamma, e]$. Suppose that $w \in e'$, and so $e, w \models \gamma$. From the lemma above, it follows that $e', w \models \gamma$. Since this applies to any $w \in e'$, we have that $e' \models \boldsymbol{K}\gamma$, and hence $\mathbf{ASK}[\gamma, e'] = \mathit{yes}$. ∎

So this shows that as long as the closed world assumption γ is not known to be false, we can always assert that it is true, and end up in a consistent epistemic state where it is known to be true (quite unlike the example in Theorem 5.5.2). Moreover, we will see in Chapter 8 that the resulting epistemic state also has the desirable property that nothing else is known: in a precise sense, *all* that is known in this state is γ together with what was known before.

- $Teach(tom, sam)$
- $Teach(tina, sue) \wedge [Teach(tom, sue) \vee Teach(ted, sue)]$
- $\exists x Teach(x, sara)$
- $\forall x[Teach(x, sandy) \equiv (x = ted)]$
- $\forall x \forall y[Teach(x, y) \supset (y = sam) \vee (y = sue) \vee (y = sara) \vee (y = sandy)]$

Figure 5.1: The Example KB

5.7 A detailed example

In this final section, we will consider a larger example of the use of **TELL** and **ASK**. This will allow us to explore how these operations can be used to probe what is known and add to it in a way that would be impossible if we only used $\mathcal{L}$. In the first subsection below, we will start with an initial epistemic state and ask a series of questions. In the second subsection, we will consider the effect of various assertions.

To keep the discussion as intuitive as possible, we will define the initial epistemic state as the set of all worlds satisfying a finite collection of sentences which can be thought of as a KB, a representation of what is known. As in earlier chapters, we will use a single predicate, *Teach*, and again use proper names that begin with a "*t*" for the first argument, the teacher, and proper names that begin with an "*s*" for the second argument, the student. As before, these should be understood as standard names. We will not use either function or constant symbols in any of the examples unless explicitly indicated.

The knowledge base we begin with appears in Figure 5.1. Notice that the first four sentences tell us what we know about each of the four students. The last sentence says that as far as the teaching relation is concerned, there are only four students. The starting epistemic state e is defined as the set of all world states satisfying this KB. Equivalently, $e = \mathbf{TELL}[\text{KB}, e_0]$. Thus we have that $e \models \boldsymbol{K}\phi$ (where ϕ is objective) iff (KB $\supset \phi$) is valid.

5.7.1 Examples of ASK

To illustrate the operation of **ASK**, we consider various arguments of the form $\mathbf{ASK}[\alpha, e]$, where e is the epistemic state above and α is one of the sentences below. For clarity, we will use three possible answers: TRUE means that α is answered *yes* (and $\neg\alpha$, *no*), FALSE means that $\neg\alpha$ is answered *yes* (and α, *no*), and UNKNOWN, otherwise.

1. $Teach(tom, sam)$ TRUE
   This is the simplest type of question, and is clearly known to be true.
2. $Teach(tom, sandy)$ FALSE
   The sentence $Teach(tom, sandy)$ is known to be false, since Ted is the only teacher of

Sandy, and because these are assumed to be standard names, they are distinct.

3. $Teach(tom, sue)$ UNKNOWN
   Neither this question nor its negation are known to be true. Tom may or may not teach Sue, according to what is known in e.

4. $\boldsymbol{K}Teach(tom, sue)$ FALSE
   In contrast to the previous question, here we are asking a subjective question about what is known. Such questions are always known to be true or to be false. In this case, it is known in e that Tom is not known to teach Sue. That is, the system realizes that the previous question was not known.

5. $\exists x\, Teach(x, sara)$ TRUE
   Again this is a simple objective question that can be answered directly. It is known that Sara has at least one teacher, although no teacher has been named.

6. $\exists x\, \boldsymbol{K}Teach(x, sara)$ FALSE
   Here the question is whether Sara has any known teachers. The system knows that although Sara has a teacher, none are as yet known. With this question and the preceding, we can distinguish between knowledge of the existence of a teacher and knowledge of the identity of that teacher.

7. $\exists x\, \boldsymbol{K}Teach(x, sue)$ TRUE
   Unlike Sara, Sue does have a known teacher, namely Tina. Thus in her case, we know of both the existence and the identity of a teacher. Note that this question and the one before it is subjective and so would always be answered TRUE or FALSE.

8. $\exists x[Teach(x, sue) \wedge \neg \boldsymbol{K}Teach(x, sue)]$ TRUE
   Having established in the previous question that Sue has a known teacher, here we are asking if she has a teacher apart from the known ones. In other words, is the list of teachers known for Sue incomplete? The answer is yes. There is only a single known teacher for Sue, Tina; but it is also known that one of Tom or Ted teaches her, and neither is a known teacher of Sue.

9. $\exists x[Teach(x, sandy) \wedge \neg \boldsymbol{K}Teach(x, sandy)]$ FALSE
   If we ask the same question of Sandy, the answer is no. It is known that Ted is her teacher and her only one. Thus, Sandy has no teachers other than her single known teacher.

10. $\exists x[Teach(x, sam) \wedge \neg \boldsymbol{K}Teach(x, sam)]$ UNKNOWN
    If we ask the same question about Sam, the answer is unknown. We know that Tom teaches Sam, but we have no other information. So Sam may or may not have a teacher apart from this known one. Note that this question (like the two preceding ones) is not a subjective sentence, and consequently can be believed to be true, believed to be false,

or here, neither.

11. $\exists y \boldsymbol{K} \forall x [Teach(x, y) \supset \boldsymbol{K}Teach(x, y)]$ TRUE
    This is a generalized version of the preceding question posed subjectively. Do you know someone whose teachers are all known? That is, can we name somebody whose list of teachers is complete? The answer is yes: Sandy. Note that to verify this, we need to establish that the question is known to be true, which involves checking the truth of a sentence with $\boldsymbol{K}$ operators nested to depth three.

12. $\exists y(y \neq sam) \wedge \neg \boldsymbol{K}if[\forall x Teach(x, y) \supset \boldsymbol{K}Teach(x, y)]$ FALSE
    where $\boldsymbol{K}if\alpha$ is an abbreviation for $\boldsymbol{K}\alpha \vee \boldsymbol{K}\neg\alpha$.
    We established in an earlier question that Sam may or may not have teachers other than the known ones. This question asks if there is an individual other than Sam for which this is true. In other words, is there anyone other than Sam for which you don't know if you are missing any teachers? The answer is no because we know that there are only three cases to consider apart from Sam: for Sandy, we know that we have all the teachers, and for Sue and Sara, we know that we are missing one.

13. $\exists y \boldsymbol{K} \exists x [Teach(x, y) \wedge \exists z[(y \neq z) \wedge \boldsymbol{K}Teach(x, z)]]$ TRUE
    This question asks if there is an individual y known to have the property that one of her teachers x is known to teach somebody else z. For this to be true, we need to know who y is, we need not know who the x is, but for each such x, we must know who the z is.
    The answer here is yes because of Sue. For every $w \in e$, we either have

    $$e, w \models Teach(tom, sue) \wedge (sue \neq sam) \wedge \boldsymbol{K}Teach(tom, sam)$$

    or

    $$e, w \models Teach(ted, sue) \wedge (sue \neq sandy) \wedge \boldsymbol{K}Teach(ted, sandy).$$

    So no matter if Tom or Ted teaches Sue, both of them are known to teach someone other than Sue. Consequently, Sue is known to have a teacher x who is known to teach somebody else, even though we do not know who that x is:

    $$e \models \boldsymbol{K}\exists x.Teach(x, sue) \wedge \exists z[(sue \neq z) \wedge \boldsymbol{K}Teach(x, z)].$$

    Note that by using nested $\boldsymbol{K}$ operators and quantifiers, we can insist on the individual z being known as a function of some other unknown individual x. This would fail, for example, if instead of knowing that Tom teaches Sam, all we knew was that Tom teaches someone other than Sue.

5.7.2 Examples of TELL

To illustrate the operation of **TELL**, we will consider some example assertions. In each case, we present a sentence α followed by notation $[e \rightarrow e']$, where e is either the initial

epistemic state above or the result of a previous assertion, and where $e' = \mathbf{TELL}[\alpha, e]$.

1. $\forall x[\mathit{Teach}(x, \mathit{sue}) \supset \mathit{Teach}(x, \mathit{sara})]$ $[e \rightarrow e_1]$
   This is an assertion of a sentence without $\boldsymbol{K}$ operators, and so the resulting state is simply the set of world states w that are in e and satisfy the assertion. In all such world states, we have that Tina teaches Sara, as does one of Tom or Ted. Consequently, these facts would be known in e_1.
2. $\forall x[\boldsymbol{K}\mathit{Teach}(x, \mathit{sue}) \supset \mathit{Teach}(x, \mathit{sara})]$ $[e \rightarrow e_2]$
   In this assertion, we do not say that every teacher of Sue teaches Sara, but only the currently known ones. In the initial state e, Tina is the only known teacher of Sue, so this assertion says nothing about either Tom or Ted. In both e_1 and e_2 we would have a single known teacher for Sara, namely Tina, but only in e_1 would we also know that there was an additional teacher apart from Tina. So if β is the sentence

   $$\exists x[\mathit{Teach}(x, \mathit{sara}) \wedge \neg \boldsymbol{K}\mathit{Teach}(x, \mathit{sara})]$$

   then $\mathbf{ASK}[\beta, e]$ is *yes*, and $\mathbf{ASK}[\beta, e_1]$ is *yes*, but since $e_2 \models \neg\boldsymbol{K}\beta \wedge \neg\boldsymbol{K}\neg\beta$, both $\mathbf{ASK}[\beta, e_2]$ and $\mathbf{ASK}[\neg\beta, e_2]$ would be *no*. In other words, we started out by knowing we were missing some of Sara's teachers, but after finding out that Tina is one of them, we no longer know whether or not we are still missing any.
3. $\forall x[\mathit{Teach}(x, \mathit{sara}) \supset \exists y \boldsymbol{K}\mathit{Teach}(x, y)]$ $[e_2 \rightarrow e_3]$
   This starts in the state e_2 where all that is known about Sara is that Tina teaches her, and we assert that all of her teachers have a currently known student. This means that all of her teachers must be one of Tom, Tina, or Ted, since these are the only individuals with known students (for Sam, Sue and Sara, and Sandy, respectively) in e_2. The assertion, however, does not identify any new teachers for Sara so that the known teachers in e_3 are the same as the known teachers in e_2: just Tina.
   This is a good example of how a set can be bounded from below by the known instances and from above by the potential instances (that is, the individuals not known to be non-instances). In this case, Sara's teachers are bounded from below by $\{\mathit{tina}\}$, and from above by $\{\mathit{tom}, \mathit{tina}, \mathit{ted}\}$. To have complete knowledge of Sara's teachers, all that is needed is to settle the case of Tom and Ted.
4. $\forall x[\mathit{Teach}(x, \mathit{sara}) \supset \neg\exists y \boldsymbol{K}[\mathit{Teach}(x, \mathit{sue}) \vee \mathit{Teach}(y, \mathit{sue})]]$ $[e \rightarrow e_4]$
   The assertion starts from e again and says that anyone who teaches Sara cannot be one of two individuals such that it is known that one of them teaches Sue. After this assertion we have the same set of known teachers for Sara, but the assertion addresses the potential teachers. It clearly rules out Tina as a teacher (since for this x there is such a y, namely Tina herself). Taking the case of Tom, although he is not known to teach Sue in e, it is known that either Tom or Ted teaches Sue, so Tom is an x such that there is a y for which it is known that either x or y teaches Sue. Thus Tom cannot be a

teacher of Sara. A similar argument rules out Ted. Thus, we have ruled out as potential teachers Tina, Tom and Ted.

The last assertion above is a good example of using a set of potential instances as candidates or suspects in an assertion. Consider an individual, Terry, about which absolutely nothing is known in e. Neither Tom nor Terry are known teachers of Sue, and both Terry and Tom are potential teachers of Sue. However, there is a difference between the two: Tom is a candidate teacher in the sense that it is known that one of Tom or Ted teaches Sue; nothing comparable is known about Terry. More generally, we might say that x is a *candidate* instance of P if x is not a known instance, but is a member of a candidate set for P, where a candidate set is a minimal set of individuals such that it is known that one of them is an instance of P. In the case of Sue's teachers, the set {Tom, Ted} is a candidate set, and so both are candidates. We know that one of Tom, Ted, and Terry must also be a teacher of Sue, but this is not a minimal set, and so Terry is not a candidate.

This notion of a candidate is useful to help form intuitions about default reasoning, which we explore in Chapters 10 and 11. It is often useful to be able to assume of certain individuals that they have a certain property unless they are known not to have it. For example, a person might be assumed to be innocent of a crime unless proven guilty. In some cases, however, there may be a set of suspects, where there is good reason to believe that one of them is guilty, although none of them are known to be guilty. Although any suspect might be innocent, they cannot all be assumed to be innocent. Although we may wish to assume innocence as broadly as possible, we may have to temporarily withhold that assumption for the candidate set.

5.8 Other operations

Having examined definitions for the knowledge-level operations of **TELL** and **ASK**, it is worth remembering that these are only two of many possible interaction operations we might consider. In this section, we briefly investigate two others: one involving definitions, and one involving wh-questions.

5.8.1 Definitions

The idea behind introducing *definitions* in a KB is this: we want to extend our vocabulary of predicate or function symbols in terms of the existing ones. For example, we might want to have a predicate symbol *FlyingBird*, which instead of being independent of all other predicate symbols, simply means the conjunction of *Bird* and *Fly*.

We could, of course, simply assert a universally quantified biconditional such as

$$\forall x (FlyingBird(x) \equiv Bird(x) \wedge Fly(x)).$$

The trouble with this is that it looks exactly like any other fact we might know about the predicate. For example, in some application we might know that all and only the birds in my cage fly:

$$\forall x(\mathit{FlyingBird}(x) \equiv \mathit{Bird}(x) \wedge \mathit{InCage}(x)).$$

Logically, the two facts are indistinguishable, but clearly the second one is not intended to say what the predicate *means*.

One simple way of handling definitional information is to imagine an epistemic state as having two components: $e = \langle e_a, e_d \rangle$, where e_a is the *assertional epistemic state* resulting from **TELL** operations as before, and e_d is the *definitional epistemic state* resulting from new **DEFINE** operations. So **TELL**$[\alpha, \langle e_a, e_d \rangle]$ would be defined to change e_a only, as specified earlier. Similarly, **DEFINE**$[P(\vec{x}), \phi[\vec{x}], \langle e_a, e_d \rangle]$ would be defined to change e_d only. The P here is an *n-ary* predicate symbol, and ϕ is a formula with n free variables, its definition.[3] For example, we could have

$$\textbf{DEFINE}[\mathit{FlyingBird}(x),\ (\mathit{Bird}(x) \wedge \mathit{Fly}(x)),\ e]$$

Having separated e_a and e_d, we can now define the **DEFINE** operation to be that of asserting the universally quantified biconditional over e_d (with no danger of confusion).[4]

With the epistemic state broken into two parts, we can consider asking questions about the definitions only. For example, we can define

$$\textbf{ASK-DEF}[\alpha, \langle e_a, e_d \rangle] = \begin{cases} \mathit{yes} & \text{if } e_d \models \boldsymbol{K}\alpha \\ \mathit{no} & \text{otherwise} \end{cases}$$

This is just like ordinary **ASK** except that it only uses e_d: it determines whether or not α is known to be true *by definition*. For **ASK** itself, we want to use both definitions and assertions to determine what is known. Thus, we would redefine it as

$$\textbf{ASK}[\alpha, \langle e_a, e_d \rangle] = \begin{cases} \mathit{yes} & \text{if } (e_a \cap e_d) \models \boldsymbol{K}\alpha \\ \mathit{no} & \text{otherwise} \end{cases}$$

So for example, if we **DEFINE** the predicate *FlyingBird* as above, then assert using **TELL** that *FlyingBird*(*tweety*) is true, then **ASK** will correctly confirm that *Fly*(*tweety*) is true.

5.8.2 Wh-questions

In addition to yes/no questions, any knowledge representation system will need to answer *wh-questions*, that is, questions beginning with words like "who," "what," "when," "where," and "how." For example, we might want to find out: who are the teachers of

3 There may be good reasons to restrict the language of ϕ as is often done in description logics, or even to extend it to allow, for example, recursive definitions.

4 We may also want to consider other forms of definitions that do not lead to biconditionals. For example, while it may not be possible to define a "natural kind" term like *Bird*, we may still wish to express necessary definitional properties, such as its being an animal.

Sara? In an epistemic state with incomplete knowledge about teachers, however, it is not clear what form of answer would be appropriate. The simplest solution is to take an operator like **WH-ASK**[*Teacher*$(x, sara), e$] to be a question about the *known* teachers of Sara. While this would mean that the question *Teacher*$(x, sara)$ and $\boldsymbol{K}$*Teacher*$(x, sara)$ would get the same answer, the question $\neg\boldsymbol{K}\neg$*Teacher*$(x, sara)$ could still be used to find out about *potential* teachers of Sara. The actual teachers of Sara, of course, lie between these two sets.

This suggests the following definition:

$$\textbf{WH-ASK}[\alpha[\vec{x}], e] = \{\vec{n} \mid e \models \boldsymbol{K}\alpha[\vec{n}]\}.$$

The main problem with this definition is that the answer could be an infinite set of standard names, for example, when α is $(x \neq {}^{\#}1)$. We will see in Chapter 7 how even an infinite answer to a **WH-ASK** question can be finitely represented at the symbol level.

Simply returning a set of standard names for a wh-question may not be very illuminating, however. What we would ultimately like is an answer in *descriptive* terms, using meaningful constants and function symbols. This is not to suggest that we would be better off with

$$\{\vec{t} \mid e \models \boldsymbol{K}\alpha[\vec{t}]\}$$

as the answer to a wh-question, since we would then have no way of knowing how many answers there were (because many of the terms t could be co-referential). A better idea is to define a new operator, **DESCRIBE**$[n, e]$ which takes a standard name as argument and returns the set of terms known to be co-referential with n:

$$\textbf{DESCRIBE}[n, e] = \{t \mid e \models \boldsymbol{K}(t = n)\}.$$

We will see again in Chapter 7 how this potentially infinite set of terms can be represented finitely.

5.9 Bibliographic notes

The idea of characterizing a knowledge representation and reasoning service in terms of tell and ask operations first appeared in [111] and [113]. It appeared subsequently in a variety of publications, most notably in a general AI textbook [168]. The idea was inspired by similar operations defining abstract data types like stacks and queues [129]. The **TELL** operation itself presents the simplest possible model of how an epistemic state changes, where contradicting information leads to an inconsistent state. For a more delicate approach, which may involve giving up some past beliefs to preserve consistency, see [49, 72], or any of the many papers in the area of belief revision. The closed world assumption first appeared in [158], and was one of the motivations for the earliest forms of nonmonotonic

reasoning, further discussed in Chapter 9. The distinction between assertions and definitions is discussed in [9] and appeared in the KRYPTON knowledge representation system [8]. The idea of using a limited language for defining predicates (or concepts) derives from early work on semantic networks (see [45], for example), and especially the KL-ONE system [12]. Some of this research then evolved into the subarea of description logics (see [4], for instance). See [23] for a treatment of natural-kind concepts that do not admit necessary and sufficient definitions. As to wh-questions, returning more than just yes/no answers is of course the mainstay of database systems. A comparable story can be told for simple forms of knowledge bases, as in logic programming [133]. For a more general KB, perhaps the clearest account is that of Green's answer extraction [56, 152], although this still only applies when the KB is in a restricted syntactic form.

5.10 Exercises

1. Show that subjective knowledge is complete, in that if σ is subjective, then either $\mathbf{ASK}[\sigma, e] = yes$ or $\mathbf{ASK}[\neg\sigma, e] = yes$.
2. Show that $\mathbf{ASK}[\forall x.P(x) \supset \neg \boldsymbol{K} P(x), e_0] = yes$.
3. Prove that if ϕ and ψ are objective, the order in which they are asserted is unimportant: $\mathbf{TELL}[\phi, \mathbf{TELL}[\psi, e]] = \mathbf{TELL}[\psi, \mathbf{TELL}[\phi, e]]$.
4. Give an example epistemic state where an individual is known to have some property, but after an assertion, it not known to have that property.
5. Present a non-subjective sentence for which knowledge is nonmonotonic.
6. Consider the positive and negative subjective sentences of Exercise 3 of Chapter 4. Show that knowledge is monotonic for the positive ones.
7. Construct an example epistemic state (as a set of world states satisfying some property) where the assertion of γ is redundant because it is already known to be true.
8. Construct a variant of the example KB such that unlike Question 11,

   $$\exists y \boldsymbol{K} \forall x [Teach(x, y) \supset \boldsymbol{K} Teach(x, y)]$$

   comes out FALSE, but

   $$\boldsymbol{K} \exists y \forall x [Teach(x, y) \supset \boldsymbol{K} Teach(x, y)]$$

   comes out TRUE.
9. Describe or construct an epistemic state where the candidate instances of P are the same as the potential instances.
10. Consider the following generalization of candidate instances. Define an objective sentence ϕ to be *explainable* in state e iff there is some objective ψ such that $\neg \boldsymbol{K} \neg \psi$

and $\boldsymbol{K}(\psi \supset \phi)$ are both true. If also $\models (\psi \supset \phi)$, we say that ϕ is trivially explainable. Show that if n is a candidate instance of P then $P(n)$ is non-trivially explainable, but that the converse is not true. Show that n is a potential instance of P iff $P(n)$ is explainable.

11. What is **ASK**$[\exists x[\neg\exists y(\boldsymbol{K}Teach(x, y)) \wedge Teach(x, sue)], e]$?
12. In **ASK** Number 13, show that we cannot move any other quantifiers outside the $\boldsymbol{K}$ operator without changing the meaning of the question.
13. Show that
$$e_3 \models \exists y_1 y_2 y_3 \boldsymbol{K}\forall x[Teach(x, sara) \supset (x = y_1) \vee (x = y_2) \vee (x = y_3)].$$
14. Describe the result of **TELL**$[\forall x[Teach(x, n) \supset \boldsymbol{K}Teach(x, n)], e]$, for $n = sue$, sam, $sandy$, $sara$.
15. Define a new interaction operator **HOW-MANY** which tells us how many instances of a formula are known to be true. In particular, it should take a formula with one free variable as argument, and return a pair of numbers $\langle i, j\rangle$, such that it is known that there are at least i and at most j true instances of the formula. Describe a KB where this operator would provide useful information, but **WH-ASK** would not.

6 Knowledge Bases as Representations of Epistemic States

In what we have seen so far, we have been thinking about knowledge in two very different ways:

- as characterized by a symbolic *knowledge base* or KB, that is, a collection of sentences about the world, where what is known is what can be inferred from the sentences;
- as characterized by an *epistemic state*, that is, a set of world states, where what is known is what is true in all of the world states.

While these two notions are clearly related, it turns out that there are also interesting differences between them. In this chapter we will explore in detail their relationship.

In the first section, we observe that many epistemic states are equivalent, in the sense that they satisfy exactly the same set of sentences. Choosing representatives for these equivalence classes of epistemic states simplifies the results to follow. In Section 6.2, we define what it means for an objective KB to represent what is known in an epistemic state. An epistemic state is defined to be representable if such a KB exists. In Section 6.3 we prove that there are epistemic states that are not representable in this sense, but in Section 6.4, we show that we can usually ignore these states, in the sense that any satisfiable sentence of $\mathcal{KL}$ is satisfied in a representable epistemic state. The KB in question may need to be infinite, however, and we prove in Section 6.5 that finitely representable epistemic states are not sufficient: there is a satisfiable sentence of $\mathcal{KL}$ that is false at every finitely representable epistemic state. Finally, in Section 6.6 we discuss the implications of these results for **TELL** and **ASK**.

6.1 Equivalent epistemic states

The easiest way to see that there is a difference between the two views of knowledge mentioned above is to consider a simple cardinality argument. If we take knowledge to be characterized by the set of all sentences known, then because there are only $\aleph_0$ sentences in $\mathcal{KL}$, there can be at most $2^{\aleph_0}$ distinct states of knowledge. But if we take knowledge to be characterized by a set of world states, then because there are $2^{\aleph_0}$ world states, there would be $2^{2^{\aleph_0}}$ distinguishable states of knowledge.

It follows from this observation that many epistemic states know exactly the same set of sentences. We call two epistemic states e and e' *equivalent* (which we write $e \approx e'$) iff for every $\alpha \in \mathcal{KL}$, $e \models \boldsymbol{K}\alpha$ iff $e' \models \boldsymbol{K}\alpha$. This clearly defines an equivalence relation over epistemic states.

For example, consider e_0, the set of all world states. We will show that e_0 and the epistemic state formed by removing a single world state from e_0 are equivalent. Observe

that for any α, w, and e, if $e, w \models \alpha$, and if w' is the same as w except for the truth value it gives to some primitive sentence whose predicate symbol does not appear in α, then $e, w' \models \alpha$ (by induction on α). Thus, if α is true for *all* elements of $(e_0 - w)$,[1] it will be true for w too. Consequently, $e_0 \models \boldsymbol{K}\alpha$ iff $(e_0 - w) \models \boldsymbol{K}\alpha$. In other words removing a single world state from e_0 does not affect the sentences believed, and so results in an equivalent state.

Because we want to think of knowledge functionally, in terms of the operations a knowledge-based system can perform, and because these operations are mediated by linguistic arguments, the difference between e_0 and $e_0 - w$ is not something we really care about. The **ASK** operation would not be able to tell the difference between them, which can be thought of as an artifact of the modeling process. This suggests that we should restrict our attention to *equivalence classes* of epistemic states. We can do this by finding suitable representatives for each equivalence class.

To do so, first observe that a world state can be added to an epistemic state if it satisfies everything that is known:

Theorem 6.1.1: *For any e and w*,

$$e \approx (e + w) \textit{ iff for every } \alpha \textit{ such that } e \models \boldsymbol{K}\alpha,\ e, w \models \alpha.$$

Proof: Let e' be $e + w$. First we assume that $e \approx e'$ and show that if $e \models \boldsymbol{K}\alpha$, then $e, w \models \alpha$. Observe that in general, for any β and any w', $e, w' \models \beta$ iff $e', w' \models \beta$, by a simple induction argument. So suppose $e \models \boldsymbol{K}\alpha$. Then $e' \models \boldsymbol{K}\alpha$ since $e \approx e'$, in which case, $e', w \models \alpha$, and so $e, w \models \alpha$.

For the converse, assume that for every α such that $e \models \boldsymbol{K}\alpha$, $e, w \models \alpha$. We will show $e \approx e'$ by showing that for any β and any w', $e, w' \models \beta$ iff $e', w' \models \beta$. The proof is by induction. It clearly holds for atomic sentences and equalities, and by induction for negations, conjunctions, and quantifications. Also if $e' \models \boldsymbol{K}\beta$, then $e \models \boldsymbol{K}\beta$, since e' is a superset of e. So finally, suppose that $e' \models \neg\boldsymbol{K}\beta$. Then, for some $w' \in e'$, we have $e', w' \models \neg\beta$, and so $e, w' \models \neg\beta$ by induction. Now there are two cases: if $w' \in e$, then $e \models \neg\boldsymbol{K}\beta$ directly; if $w' = w$, then again $e \models \neg\boldsymbol{K}\beta$, by assumption about w. This completes the proof. ∎

So under certain conditions, we can add world states to an epistemic state and preserve equivalence. Let us say that an epistemic state e is *maximal* iff for every w, if $e \approx e + w$ then $w \in e$. So a maximal epistemic state is one where the addition of any world state would involve a change in belief for some sentence. Clearly e_0 is maximal and $e_0 - w$ is

1 When X is a set, we use the notation $X - x$ to mean $X \setminus \{x\}$. Similarly, $X + x$ means $X \cup \{x\}$.

not. We can show that every equivalence class has a *unique* maximal element, which can serve as the representative for the class:

Theorem 6.1.2: *For any epistemic state e, there is a unique maximal state e^+ that is equivalent to it.*

Proof: For any e, let $e^+ = \{w \mid e \approx e + w\}$. We will first show that $e \approx e^+$ by showing that for any w and α, $e, w \models \alpha$ iff $e^+, w \models \alpha$, again by induction. As in the previous theorem, the only difficult case is when $e^+ \models \neg \boldsymbol{K}\alpha$. Then, for some $w \in e^+$, we have $e^+, w \models \neg\alpha$, and so $e, w \models \neg\alpha$ by the induction hypothesis. But since $w \in e^+$, we know that $e \approx e + w$. Therefore, $(e + w), w \models \neg\alpha$ also, and so $(e + w) \models \neg \boldsymbol{K}\alpha$ and consequently, $e \models \neg \boldsymbol{K}\alpha$. This establishes that $e \approx e^+$.

To show that e^+ is the unique maximal set, we show that for any e', if $e \approx e'$ then $e' \subseteq e^+$. That is, we need to show that if $e \approx e'$ then for any $w \in e'$, $e \approx e + w$, and so $w \in e^+$. To show this, we simply observe that if $w \in e'$, then w satisfies everything that is known in e as well as in e', and so by the previous theorem, we have that $e \approx e + w$. ∎

Thus maximal states can be used as representatives of the equivalence classes. Moreover, by Theorem 6.1.1 we have that

Corollary 6.1.3: *A state e is maximal iff there is a set Γ such that*

$$e = \{w \mid e, w \models \alpha, \textit{for every } \alpha \in \Gamma\}.$$

In much of what follows, we will restrict our attention to maximal epistemic states, since these cover all of the possibilities admitted by the logic $\mathcal{KL}$:

Corollary 6.1.4: *For any (possibly infinite) set of sentences Γ, if Γ is satisfiable, then it is satisfied by a maximal epistemic state.*

So in terms of the logic $\mathcal{KL}$, maximal sets are fully *sufficient*; in Section 6.3, we will consider the converse question: are *all* maximal epistemic states required, or can we get by with a subset of them?

6.2 Representing knowledge symbolically

As Corollary 6.1.3 shows, maximal epistemic states can be completely characterized by the sentences that are known. Let us call a set of sentences Γ a *belief set* iff there is an

epistemic state e such that $\Gamma = \{\alpha \mid e \models \boldsymbol{K}\alpha\}$. So Γ is a belief set for e if it is everything believed in e. Then we have the following:

Theorem 6.2.1: *There is a bijection between belief sets and maximal epistemic states.*

It follows then that there are only as many maximal states as there are belief sets.

On the other hand, when we think of knowledge linguistically, at least informally, we usually do not think in terms of *all* sentences known. Rather we think in terms of a symbolic representation of what is known, that is, a collection of sentences (or perhaps some other symbolic data structures), typically finite, that is sufficient to characterize the complete belief set.

For example, if we use a sentence α of $\mathcal{KL}$ as part of our representation, then not only is α known, but so are all of its logical consequences. Moreover, by introspection, $\boldsymbol{K}\alpha$ and $\boldsymbol{K}\neg\boldsymbol{K}\neg\alpha$ are also known. In fact, any β such that $(\boldsymbol{K}\alpha \supset \boldsymbol{K}\beta)$ is valid should also be known.

But this is still not the full belief set. To see why, suppose that ϕ is objective, and the epistemic state we are trying to represent is $e = \{w \mid w \models \phi\}$. The belief set associated with e clearly starts with a belief in ϕ, and contains all sentences β as above. But now, assuming that ϕ is satisfiable, let ψ be any objective sentence such that $(\phi \supset \psi)$ is *not* valid. Then, there is a $w \in e$ such that $w \models \neg\psi$, and so $e \models \neg\boldsymbol{K}\psi$, and thus, $e \models \boldsymbol{K}\neg\boldsymbol{K}\psi$. Therefore, $\neg\boldsymbol{K}\psi$ is part of the belief set too. Notice that $(\boldsymbol{K}\phi \supset \boldsymbol{K}\neg\boldsymbol{K}\psi)$ is *not* valid: just because you believe ϕ, it does not *follow* that you do not believe ψ.

So if we are to extract a belief set (and an epistemic state) from a representation like ϕ, we have to consider not only what follows from believing ϕ, but also what follows from *not* believing other sentences. In other words, a representation of an epistemic state must capture what is believed and the fact that what is represented is *all* that is believed. This is of course how we understand a knowledge base: it represents what is known and all that is known.

Can we use any collection of sentences of $\mathcal{KL}$ to represent an epistemic state? This is a difficult question in general, and we will defer it to Chapter 8. But one special case is much simpler: the objective sentences. Clearly our emphasis has been on objective knowledge about the world, and it ought to be possible to use objective sentences to represent knowledge.

So imagine that we start with KB, an arbitrary set (not necessarily finite) of objective sentences of $\mathcal{KL}$. The *epistemic state represented by* KB, which we write $\Re[\![\text{KB}]\!]$, is the one where all the sentences in the KB are known, and nothing else. Formally,

$$\Re[\![\text{KB}]\!] = \{w \mid w \models \phi, \text{ for every } \phi \in \text{KB}\}.$$

Note that according to this definition, $\Re[\![\mathrm{KB}]\!]$ is always maximal. When the KB in question is a finite set $\{\phi_1, \ldots, \phi_k\}$, we have that $\Re[\![\mathrm{KB}]\!]$ is the same as

$$\mathbf{TELL}[(\phi_1 \wedge \ldots \wedge \phi_k), e_0],$$

so that the epistemic state represented by KB is the state that results from being told in the initial state that everything in the KB is true. Let us call an epistemic state e *representable* if for some set KB of objective sentences, $e = \Re[\![\mathrm{KB}]\!]$, and we call it *finitely representable* if there is a finite KB such that $e = \Re[\![\mathrm{KB}]\!]$.

We can now distinguish (conceptually, at least) among three varieties of epistemic states. Given an epistemic state e, if there is a set Γ such that

$$e = \{w \mid \text{for every } \alpha \in \Gamma,\ e, w \models \alpha\},$$

then e is maximal; if there is an *objective* Γ satisfying the above, then e is representable; finally, if there is a Γ that is both *finite* and *objective*, then e is finitely representable.

One very important and immediate property of representable epistemic states is that they are completely determined by the objective knowledge they contain.

Theorem 6.2.2: *If e_1 and e_2 are representable states, then*

$$e_1 = e_2 \textit{ iff for every objective } \phi,\ e_1 \models \boldsymbol{K}\phi \textit{ iff } e_2 \models \boldsymbol{K}\phi.$$

So any two representable states that agree on the objective facts agree on everything else. This is not true in general for maximal epistemic states, as can be seen from the two states e_1 and e_2 used in Theorem 4.6.2 of Chapter 4.

Clearly every finitely representable state is representable, and every representable state is maximal. But what about the converses? In the next sections, we will examine these questions in detail.

6.3 Some epistemic states are not representable

In the previous section we showed that we could limit our attention to maximal epistemic states without any loss of generality whatsoever. For representable states, however, this is not the case:

Theorem 6.3.1: *There is an infinite satisfiable set of sentences of $\mathcal{KL}$ that is not satisfied by any representable epistemic state.*

The proof uses details from the proof of Theorem 4.6.2 from Chapter 4. To recap, let Ω be the set $\{{}^{\#}1, {}^{\#}3, {}^{\#}5, \ldots\}$, let Φ be the set of objective sentences consisting of $\{(t = {}^{\#}1)\}$ for every primitive term t, $\{\neg\phi\}$ for every primitive sentence ϕ whose predicate letter is

not P, and finally $\{P(n)\}$ for every $n \in \Omega$. Let e_1 be $\{w \mid w \models \Phi\}$. Let $\overline{w}$ be the (unique) element of e_1 such that for every $n \notin \Omega$, $w \models \neg P(n)$, and let e_2 be $e_1 - \overline{w}$. Finally, define Γ_1 and Γ_2 by

$$\Gamma_1 = \{\boldsymbol{K}\phi \mid e_2 \models \boldsymbol{K}\phi\} \cup \{\neg\boldsymbol{K}\phi \mid e_2 \models \neg\boldsymbol{K}\phi\}$$
$$\Gamma_2 = \{\sigma \mid e_2 \models \sigma\}.$$

The set Γ_2 is the one that we will show is not satisfied by any representable state.

First observe that Γ_2 is indeed satisfiable, since e_2 satisfies it. Also, Γ_1 is a subset of Γ_2, so that anything claiming to satisfy the latter must also satisfy the former. Next, e_1 is a representable state, represented by Φ. Finally note that from the proof of Theorem 4.6.2, although e_1 satisfies Γ_1, it does not satisfy Γ_2 since it fails to satisfy the sentence $\boldsymbol{K}\exists x[P(x) \wedge \neg\boldsymbol{K}P(x)]$.

So to prove the theorem: let e be any representable state that satisfies Γ_1. By Theorem 6.2.2, $e = e_1$, and so e does not satisfy Γ_2. Thus, no representable state satisfies Γ_2.

6.4 Representable states are sufficient

In the previous section, we showed that in terms of satisfiability in $\mathcal{KL}$, we needed to allow for epistemic states that were not representable. To prove this, we had to consider an infinite set of sentences in $\mathcal{KL}$. In any realistic application, however, we will only be using *finite* sets of sentences. The question we now ask is whether the above theorem would continue to hold.

Fortunately, the answer here is no:

Theorem 6.4.1: *Any sentence (or finite set of sentences) of $\mathcal{KL}$ that is satisfiable is satisfied by some representable epistemic state. Equivalently, a sentence of $\mathcal{KL}$ is valid iff it is true at all world states and all representable epistemic states.*

This is an important result since it shows that as far as the *logic* of $\mathcal{KL}$ is concerned, representable states are sufficient: validity in the logic is exactly the same as truth in all representable states.

The proof, however, is not trivial, and proceeds as follows: starting with some satisfiable sentence γ, we will construct an infinite satisfiable set Γ that includes γ and that is satisfied by a representable state. The construction is similar to that used in the completeness proof of Chapter 4, in that we construct the set iteratively, adding sentences while preserving the set's satisfiability. To allow the set to be satisfied by a representable state, we must ensure that all required knowledge in that state can be reduced to objective terms. To do so, we will use new predicates not appearing in the set to capture any non-objective

knowledge required by γ. That is, we use new predicate letters to convert non-objective knowledge into objective knowledge.

First some notation: suppose $\phi[x_1, \ldots, x_k]$ is an objective formula with free variables $x_1, \ldots, x_k$, and P is a predicate letter. We will let $P \bullet \phi$ be the sentence

$$\forall x_1 \ldots \forall x_k \boldsymbol{K}[P(x_1, \ldots, x_k) \equiv \boldsymbol{K}\phi].$$

Thus the subjective sentence $P \bullet \phi$ expresses the property that it is known that instances of P correspond exactly to the *known* instances of ϕ.

Now we need the following lemma:

Lemma 6.4.2: *Suppose predicate P does not appear in formula ϕ or sentence α, and that $(P \bullet \phi \supset \alpha)$ is valid. Then so is α.*

Proof: Assume to the contrary that the given α is not valid, and so $e, w \models \neg\alpha$. For any w, let $w^\bullet$ be exactly like w except that $w^\bullet[P(n_1, \ldots, n_k)] = 1$ iff $e \models \boldsymbol{K}\phi[n_1, \ldots, n_k]$, and let $e^\bullet$ be the set of $w^\bullet$ for all $w \in e$. Then we have that $e \models \boldsymbol{K}\phi[n_1, \ldots, n_k]$ iff $e^\bullet \models \boldsymbol{K}\phi[n_1, \ldots, n_k]$, since ϕ does not use P, and so, $e^\bullet \models P \bullet \phi$. However, because $e, w \models \neg\alpha$, we have that $e^\bullet, w \models \neg\alpha$, since α does not use P either. This contradicts the assumption that $(P \bullet \phi \supset \alpha)$ is valid. ∎

From this, we get:

Corollary 6.4.3: *If Γ is finite and satisfiable, then for any objective formula ϕ, there is a predicate P such that $\Gamma \cup \{P \bullet \phi\}$ is satisfiable.*

and then:

Corollary 6.4.4: *If γ is satisfiable, then there is a satisfiable set Γ containing γ and such that for every objective formula ϕ, there is a predicate P such that $P \bullet \phi$ is in Γ.*

The claim here is that this set Γ is not only satisfiable, but is satisfied by a representable state. Let e be a maximal state that satisfies Γ, and let Φ be the set of all objective sentences ϕ such that $e \models \boldsymbol{K}\phi$. We will show that Φ represents e.

First some terminology: for any formula α, define $\alpha^\bullet$ to be the following formula:

if α is objective, then $\alpha^\bullet = \alpha$;
$(\neg\alpha)^\bullet = \neg\alpha^\bullet$;
$(\alpha \wedge \beta)^\bullet = (\alpha^\bullet \wedge \beta^\bullet)$;

$(\forall x\alpha)^\bullet = \forall x\alpha^\bullet$;
$(\boldsymbol{K}\alpha)^\bullet = P(x_1, \ldots, x_k)$, where $x_1, \ldots, x_k$ are
the free variables in α and $P\bullet\alpha^\bullet \in \Gamma$.

Note that $\alpha^\bullet$ is always objective. Then we have the following:

Lemma 6.4.5: *For the given e above, for any w such that* $w \models \Phi$*, for every formula* α*, and for all names* $n_1, \ldots, n_k$

$e, w \models \alpha[n_1, \ldots, n_k]$ *iff* $w \models \alpha^\bullet[n_1, \ldots, n_k]$.

Proof: The proof is by induction on α. The lemma clearly holds for atomic sentences, equalities, and by induction, for negations, conjunctions, and quantifications (given that α can be an open formula with any number of free variables).

Finally consider $\boldsymbol{K}\alpha[n_1, \ldots, n_k]$. First observe that $e \models \boldsymbol{K}\alpha[n_1, \ldots, n_k]$ iff $e \models \boldsymbol{K}\alpha^\bullet[n_1, \ldots, n_k]$: the former holds iff for every $w' \in e$, $e, w' \models \alpha[n_1, \ldots, n_k]$ iff for every $w' \in e$, $w' \models \alpha^\bullet[n_1, \ldots, n_k]$ (by the induction hypothesis, since $w' \models \Phi$ when $w' \in e$) iff the latter holds.

Now suppose that $P\bullet\alpha^\bullet$ is the element of Γ guaranteed for $\alpha^\bullet$ in the construction. So $e \models P\bullet\alpha^\bullet$. There are two cases to consider: if $e \models \boldsymbol{K}\alpha[n_1, \ldots, n_k]$, then by the above, $e \models \boldsymbol{K}\alpha^\bullet[n_1, \ldots, n_k]$, and therefore $e \models \boldsymbol{K}P(n_1, \ldots, n_k)$, which means that $P(n_1, \ldots, n_k) \in \Phi$, and thus, $w \models P(n_1, \ldots, n_k)$; similarly, if $e \models \neg\boldsymbol{K}\alpha[n_1, \ldots, n_k]$, then $e \models \neg\boldsymbol{K}\alpha^\bullet[n_1, \ldots, n_k]$, and so $e \models \boldsymbol{K}\neg P(n_1, \ldots, n_k)$ and $w \models \neg P(n_1, \ldots, n_k)$, by the same argument. Either way, $e \models \boldsymbol{K}\alpha[n_1, \ldots, n_k]$ iff $w \models (\boldsymbol{K}\alpha)^\bullet[n_1, \ldots, n_k]$, which completes the proof. ∎

When α has no free variables, we get as an obvious corollary:

Corollary 6.4.6: *Let* $w \in e$*. Then* $e, w \models \alpha$ *iff* $w \models \alpha^\bullet$*, and so* $e \models \boldsymbol{K}\alpha$ *iff* $e \models \boldsymbol{K}\alpha^\bullet$.

Finally, to show that e is represented by Φ, we need to show that $e = \{w \mid w \models \Phi\}$. Clearly, if $w \in e$, then $w \models \Phi$. For the converse, we assume that $w \models \Phi$, and show that $w \in e$. Since e is maximal, by Theorem 6.1.1, we need only show that w satisfies everything known in e, that is, that for any α such that $e \models \boldsymbol{K}\alpha$, we have that $e, w \models \alpha$. So suppose that $e \models \boldsymbol{K}\alpha$. By the above corollary, we have that $e \models \boldsymbol{K}\alpha^\bullet$, and so $w \models \alpha^\bullet$, since $\alpha^\bullet \in \Phi$. Then by the lemma above, we get that $e, w \models \alpha$, which completes the proof of the theorem.

So what this theorem shows is that when we are talking about a state of knowledge using only a finite set of sentences of $\mathcal{KL}$, we are justified in interpreting this as pertaining to a representable state of knowledge. We saw in the previous section that it is possible to

1. $\forall xyz[R(x, y) \wedge R(y, z) \supset R(x, z)]$
   R is transitive.
2. $\forall x \neg R(x, x)$
   R is irreflexive.
3. $\forall x[\boldsymbol{K}P(x) \supset \exists y.R(x, y) \wedge \boldsymbol{K}P(y)]$
   For every known instance of P, there is another one that is R related to it.
4. $\forall x[\boldsymbol{K}\neg P(x) \supset \exists y.R(x, y) \wedge \boldsymbol{K}\neg P(y)]$
   For every known non-instance of P, there is another one that is R related to it.
5. $\exists x \boldsymbol{K}P(x) \wedge \exists x \boldsymbol{K}\neg P(x)$
   There is at least one known instance and known non-instance of P.
6. $\exists x \neg \boldsymbol{K}P(x)$
   There is something that is not known to be an instance of P.

Figure 6.1: A sentence unsatisfiable in finite states

force the state of knowledge to be non-representable, but to do so requires an infinite set of sentences.

6.5 Finite representations are not sufficient

When we think of a representation of knowledge, we usually have in mind a finite one, that is, a finite collection of symbolic structures that can be stored and manipulated computationally. It would be nice if the characterization of knowledge offered by $\mathcal{KL}$ conformed to this view. So the question here is whether the theorem of the previous section can be strengthened: is validity in $\mathcal{KL}$ the same as truth in all finitely representable epistemic states?

Unfortunately, this is not the case. As we will show, if we were to limit ourselves to finitely representable epistemic states, we would have to change the logic $\mathcal{KL}$, in the sense that new sentences would be valid. We will discuss the implications of this later in Section 6.6. First, we state the fact formally:

Theorem 6.5.1: *There is a satisfiable sentence π such that π is false at every finitely representable epistemic state. Equivalently, there is a sentence that is not valid in $\mathcal{KL}$, but that is true at every finitely representable epistemic state.*

The proof involves a sentence π that states that there is an infinite set of known instances of a predicate P and an infinite set of known non-instances of P. We then show that although π is satisfiable, it cannot be satisfied by any finitely representable state.

The π in question is the conjunction of the sentences appearing in Figure 6.1. First, observe that π is satisfiable. Choose an ordering of the standard names, n_1, and n_2, and

so on. Let w be any world state such that $R(m, n)$ is true exactly when m appears earlier than n in the ordering. Let e be the set of all world states that satisfy all of the following objective sentences:

$$\{P(n_1), \neg P(n_2), P(n_3), \neg P(n_4), \ldots.\}$$

Then it is easy to verify that $e, w \models \pi$.

Notice what π is doing here: by making R be transitive and irreflexive, we are forcing it to behave like *less than*, and so we are forcing every known instance of P to have a "greater" one, and similarly, for the known non-instances. The second to last conjunct makes sure that these sets are not empty, so that they must be infinite, That is, we must have an infinite chain of known instances, and an infinite one of known non-instances. The very last conjunct makes sure that the epistemic state is consistent, so that not every sentence is known.

To complete the proof, we first need the following easy to prove property of representable states:

Theorem 6.5.2: *Suppose* KB *is objective and* $e = \Re[\![\text{KB}]\!]$. *Then for any objective sentence* ϕ, $e \models \boldsymbol{K}\phi$ *iff* $\text{KB} \cup \{\neg\phi\}$ *is unsatisfiable.*

We also need the following property of objective sentences:

Lemma 6.5.3: *Suppose* ϕ *is a satisfiable objective sentence. Let*

$$A = \{n \mid (\phi \supset P(n)) \textit{ is valid}\} \qquad B = \{n \mid (\phi \supset \neg P(n)) \textit{ is valid}\}.$$

Then either A or B is finite.

Proof: Consider all the names appearing in ϕ. If A only contains these names, then clearly A is finite, and we are done. Otherwise, there must be an n such that $(\phi \supset P(n))$ is valid, but such that n does not appear in ϕ. Let m be any other name that does not appear in ϕ. By Theorem 4.4.2 of Chapter 4, $(\phi \supset P(m))$ must be valid too. Consequently, A contains all the names that do not appear in ϕ. Since ϕ is satisfiable, A and B must be disjoint, and so B can only contain names that do appear in ϕ, and so must be finite. ∎

Now suppose to the contrary that π is satisfied by some finitely representable state. That is, for some e and w, we have that $e, w \models \pi$, where $e = \Re[\![\phi]\!]$. Because of the last conjunct in π, e must be non-empty, and so ϕ must be satisfiable. By the theorem above, the known instances of P will be the set A in the above lemma, and the known non-instances of P will be the set B. Moreover, by the lemma, one of A or B must be finite. Thus, if e is finitely representable and consistent, it cannot have an infinite set of known instances and

known non-instances of P, which contradicts π. This completes the proof of the theorem.

What we have shown is that it is possible using a single sentence of $\mathcal{KL}$ to assert that there is knowledge that cannot be represented finitely. If we were to restrict our attention to finite states alone, we would have to arrange the semantics so that the negations of sentences like π somehow came out valid, an unlikely prospect.

6.6 Representability and TELL

After such a tortuous route, we should summarize where we stand in terms of the types of epistemic states we have considered:

1. *maximal states* are fully general in that any satisfiable set of sentences is satisfied by a maximal state.
2. *representable states*, that is, those that can be represented by a set of objective sentences, are also sufficient in that a sentence of $\mathcal{KL}$ is satisfiable iff it is true at some representable state. However, this generality does not extend to infinite sets of sentences, as it did with maximal states.
3. *finitely representable states*, that is, those that can be represented by a finite set of objective sentences, are *not* adequate for the semantics of $\mathcal{KL}$ in that there are satisfiable sentences that are false at every finitely representable state.

Thus the three categories of epistemic states are semantically distinct.

So where does this leave us in terms of providing a specification for knowledge-based systems via the **TELL** and **ASK** operations. Intuitively, it would be nice to say that finitely representable epistemic states (that is, those resulting from finite symbolic KBs) are our only concern. But the results above show that these are overly restrictive as far as $\mathcal{KL}$ is concerned. We need to allow for all representable states, including those resulting from an infinite KB, although it is far from clear how we are supposed to "implement" them.

But there is another problem. Consider the definition of the **TELL** operation from Chapter 5, and the states e_1 and e_2 used in the proof in Section 6.3. As we showed, e_1 is representable, but e_2, defined as $e_1 - \overline{w}$, is not. The problem is that e_2 is the result of a **TELL** operation:

$$e_2 = \mathbf{TELL}[\exists x(P(x) \wedge \neg \boldsymbol{K} P(x)), e_1].$$

Thus we have the following unfortunate result:

Theorem 6.6.1: *Representable states are not closed under* **TELL**.

In other words, given a representable state as argument, the result of **TELL** need not be

representable. This is a problem since in allowing for infinitely representable states, we can move to a non-representable one by applying a **TELL** operation.

How can we resolve these difficulties? As it turns out, there is a simple and elegant answer. First of all, we need a certain property of $\mathcal{KL}$ which will be demonstrated in the next chapter: although representable states are not closed under **TELL**, *finitely representable* states are. That is, the main result of the next chapter is that given a finitely represented epistemic state as argument, the result of a **TELL** operation can always be represented finitely. This is encouraging since we are clearly more interested in finite representations of knowledge than in the (non-physically realizable) infinite ones.

On the other hand, we have already cautioned against limiting our attention to finitely representable states. As we showed, the logic of $\mathcal{KL}$ must allow for the infinite ones. The solution is that, while we have to allow for these non-finite states, we do not need to ever implement them.

Imagine, epistemic states coming to exist as a result of a sequence of **TELL** operations. We start with e_0, the least informed state, and as a result of being told sentences α_1, α_2, and so on, we move through states e_1, e_2, and so on. Because e_0 is finitely representable, by the theorem of the next chapter, each of the e_i will also be finitely representable. The infinitely represented states, then can be thought of as *limit points* in this process, where an infinite number of sentences have been asserted. This means that we never actually arrive at an infinite state, but that we can get arbitrarily close. And since we never get there, we never get to go *beyond* them either with additional **TELL** operations. Thus we never have to consider a non-finitely representable state as an argument to **TELL**. This picture fully resolves the problem since the full range of states to consider are all those that result from a sequence of **TELL** operations *including the limit points* even though the limit points are themselves never further arguments to **TELL**.

In the next chapter, we supply the remaining piece, showing that finitely representable states are indeed closed under **TELL** operations, and hence that a specification of knowledge-based systems can be meaningfully limited to representable epistemic states.

6.7 Bibliographic notes

This chapter deals with the relationship between representations of knowledge and abstract epistemic states. Somewhat surprisingly, although the idea of representing knowledge symbolically is a familiar one within AI, to our knowledge, very little has been written about the correspondence between representations and states of knowledge. One reason is that there appear to be two somewhat separate communities involved: the knowledge representation community, as seen, for example in the Conference on Principles of Knowl-

edge Representation and Reasoning,[2] and the logics of knowledge community, as seen, for example, in the Conference on Theoretical Aspects of Rationality and Knowledge.[3] Early version of the results reported here appeared in [111] and [113]. These results depend on our view of a KB as encoding objective knowledge about the world. The idea that facts about what is or is not known could also be part of a KB and thus contribute to what is known is a much more complex notion, and the basis of autoepistemic logic, discussed in Chapter 10.

6.8 Exercises

1. Show that maximal states satisfy $e = \{w \mid e, w \models \boldsymbol{K}\alpha \supset \alpha\}$.
2. Prove Theorem 6.2.1.
3. Prove Theorem 6.2.2.
4. Prove Theorem 6.5.2.
5. Show that Theorem 6.2.2 can be strengthened for the quantifier-free subset of $\mathcal{KL}$: any two epistemic states that have the same objective knowledge are equivalent.
6. Show that Theorem 6.5.1 is false for the quantifier-free subset of $\mathcal{KL}$: any quantifier-free sentence that is true at all finitely representable states is valid.

2 See `https://kr.org`.
3 See `http://www.tark.org`.

7 The Representation Theorem

In our analysis of what it would mean for a system to have knowledge, we started with an informal picture of a knowledge-based system, that is, one containing a *knowledge base*: a collection of symbolic structures representing what is known to be true about the world. As we developed the logic $\mathcal{KL}$, we gradually replaced this symbolic understanding of knowledge with a more abstract one where we talked about an *epistemic state*: a set of world states any of which could be, according to what is known, the correct specification of what is true in the world.

In Chapter 5, we showed how we could define **TELL** and **ASK** operations in terms of these abstract epistemic states without appealing to any notion of symbolic representation, except as part of the interface language for assertions and questions. In the previous chapter, we showed that even though there were far more epistemic states than possible symbolic knowledge bases, as far as the logic $\mathcal{KL}$ was concerned, we could restrict our attention to those epistemic states that were representable by knowledge bases.

The question to be addressed in this chapter is the impact of this representational view of epistemic states on the **TELL** and **ASK** operations. What we will show here is how **TELL** and **ASK** can be realized or implemented using ordinary first-order knowledge bases. In particular, we will show that is possible to reduce the use of $\mathcal{KL}$ in these operations to that of $\mathcal{L}$ in the following way:

- Given a finite KB representing an epistemic state and any sentence of $\mathcal{KL}$ used as an argument to **ASK**, we can eliminate the $\boldsymbol{K}$ operators from the question, reducing the question to an objective one, and answer it using ordinary (first-order) theorem-proving operations.
- Given a finite KB representing an epistemic state and any sentence of $\mathcal{KL}$ used as an argument to **TELL**, we can eliminate the $\boldsymbol{K}$ operators from the assertion, reducing the assertion to an objective one, and represent the new epistemic state by conjoining this objective assertion and the original KB.

So starting with a finitely representable epistemic state, it will always be possible to find a finite representation of the result of a **TELL** operation, even if the assertion uses $\boldsymbol{K}$ operators. Moreover, the operation is *monotonic* in the sense that it involves adding a new objective sentence conjunctively to the previous representation. This result also establishes the fact that finitely representable states are closed under **TELL**. Perhaps more importantly, it provides a clear link between the abstract knowledge level view of knowledge and the more concrete symbol level view where symbolic representations are manipulated.

Since this representation theorem involves several steps that are interesting in their own right, before presenting the proof, we begin by discussing the argument informally.

7.1 The method

To see how the representation theorem will work, it is useful to consider a very simple example. Suppose that we have a state e represented by a KB consisting of two sentences: $P(^{\#}1)$, and $P(^{\#}2)$. What we want to consider is how to represent the result of asserting a sentence containing $\boldsymbol{K}$ operators. For example, consider the result of

$$\mathbf{TELL}[\exists x(P(x) \wedge \neg \boldsymbol{K}P(x)),\ \Re[\![\mathrm{KB}]\!]].$$

The intent of this assertion is to say that there is an instance of P apart from the known ones. Since in this case the known ones are $^{\#}1$ and $^{\#}2$, the resulting state can be represented by

$$\{P(^{\#}1),\ \ P(^{\#}2),\ \ \exists x[P(x) \wedge \neg((x = {}^{\#}1) \vee (x = {}^{\#}2))]\}.$$

So we add the assertion to the KB except that the subwff $\boldsymbol{K}P(x)$ is replaced by the wff

$$((x = {}^{\#}1) \vee (x = {}^{\#}2)).$$

In general this is the tactic we will follow: replace subwffs containing a $\boldsymbol{K}$ by objective wffs that carry the same information, and then conjoin the result with the original KB.

But what does it mean to carry the same information? It is certainly not true that the two wffs above are logically equivalent. What we do have, however, is that for the initial state, $e = \Re[\![\mathrm{KB}]\!]$, the known instances of P are precisely $^{\#}1$ and $^{\#}2$. Thus, we get for any n that

$$e \models \boldsymbol{K}P(n) \quad \text{iff} \quad n \in \{^{\#}1, {}^{\#}2\} \quad \text{iff} \quad \models ((n = {}^{\#}1) \vee (n = {}^{\#}2)).$$

If our initial KB had also contained $P(^{\#}3)$, we would have wanted the formula

$$((x = {}^{\#}1) \vee (x = {}^{\#}2) \vee (x = {}^{\#}3)).$$

Clearly the objective formula we need to replace $\boldsymbol{K}P(x)$ is a function of the initial epistemic state, $\Re[\![\mathrm{KB}]\!]$.

Consider a more difficult case. Suppose KB had been the sentence

$$\forall x[(x \neq {}^{\#}3) \supset P(x)].$$

In this case, there are an *infinite* number of known instances of P so we cannot disjoin them as above. However, we can still represent the set finitely using the wff $(x \neq {}^{\#}3)$, since

$$e \models \boldsymbol{K}P(n) \quad \text{iff} \quad n \notin \{^{\#}3\} \quad \text{iff} \quad \models (n \neq {}^{\#}3).$$

The result of the same assertion can be represented in this case by

$$\{\forall x[(x \neq {}^{\#}3) \supset P(x)],\ \exists x[P(x) \wedge \neg(x \neq {}^{\#}3)]\},$$

which is logically equivalent to

$$\{\forall x.P(x)\}.$$

In other words, if we start with all but $^{\#}3$ as the known instances of P, and then we are told that there is another P apart from the known ones, we end up knowing that everything is an instance of P.

So the procedure we will follow in general is this: Given a KB and a subwff $\boldsymbol{K}\phi$ appearing in an assertion, we find an *objective* formula with the same free variables as ϕ, which we call $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]$, and use it to replace $\boldsymbol{K}\phi$. Once all such subwffs have been replaced, the resulting objective sentence is added to the KB. For this to work, we need $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]$ to satisfy the property that for any n

$$e \models \boldsymbol{K}\phi^x_n \quad \text{iff} \quad \models \mathrm{RES}[\![\phi, \mathrm{KB}]\!]^x_n,$$

for $e = \Re[\![\mathrm{KB}]\!]$, (and suitably generalized for additional free variables). In other words, we need the formula $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]$ to correctly capture the known instances of ϕ for the epistemic state $\Re[\![\mathrm{KB}]\!]$.

7.2 Representing the known instances of a formula

The definition of RES is a recursive one, based on the number of free variables in the wff ϕ. For the base case, we need to consider what to do if ϕ has no free variables. For example, if we were to assert

$$P(^{\#}1) \wedge (P(^{\#}2) \vee \boldsymbol{K}P(^{\#}3)),$$

then since the subwff $\boldsymbol{K}P(^{\#}3)$ is subjective, it is either known to be true or known to be false. In the former case, the assertion overall should reduce to

$$P(^{\#}1),$$

and in the latter case, to

$$P(^{\#}1) \wedge P(^{\#}2).$$

The simplest way of achieving this is to let RES return an always-true objective sentence in the former case and an always-false objective sentence in the latter. It will be important not to introduce any new standard names in the process so we will use $\forall z(z = z)$ and its negation as the two sentences. We will call the former TRUE and the latter FALSE.

So for example, if KB is $\{P(^{\#}1), P(^{\#}2)\}$, then we have that

$$\mathrm{RES}[\![P(^{\#}1), \mathrm{KB}]\!] = \mathrm{TRUE},$$
$$\mathrm{RES}[\![(P(^{\#}2) \vee P(^{\#}6)), \mathrm{KB}]\!] = \mathrm{TRUE},$$
$$\mathrm{RES}[\![(P(^{\#}3) \vee P(^{\#}6)), \mathrm{KB}]\!] = \mathrm{FALSE}.$$

The decision to return the true or the false sentence is based on whether the ϕ in question is known. Because ϕ is objective, by Theorem 6.5.2, this is the same as whether or not the objective sentence $(\mathrm{KB} \supset \phi)$ is valid.

Now consider the case of $\phi(x)$ containing a single free variable x. The idea here is to construct a wff that carries the same information as an infinite disjunction that runs through all the standard names, and for each name n, considers whether or not $\phi(n)$ is known. If it is, we keep $(x = n)$ as part of the disjunction; if it is not, we discard it. For example, for the above KB, we want something like

$$\text{RES}[\![P(x), \text{KB}]\!] = ((x = {}^{\#}1) \vee (x = {}^{\#}2)),$$

since for every other n, we have that $P(n)$ is not known to be true. The test for $\phi(n)$ being known is actually a recursive call to RES with one fewer free variable, so we will really get something more like this:

$$\begin{array}{l}((x = {}^{\#}1) \wedge \text{RES}[\![P({}^{\#}1), \text{KB}]\!]) \ \vee \\ ((x = {}^{\#}2) \wedge \text{RES}[\![P({}^{\#}2), \text{KB}]\!]) \ \vee \\ ((x = {}^{\#}3) \wedge \text{RES}[\![P({}^{\#}3), \text{KB}]\!]) \ \vee \\ ((x = {}^{\#}4) \wedge \text{RES}[\![P({}^{\#}4), \text{KB}]\!]) \ \vee \ \ldots,\end{array}$$

which simplifies to the same thing, since only the first two return TRUE.

The only thing left to do is to convert this infinite disjunction to a finite formula. To do so, we focus on the names appearing in either the KB or ϕ. Since the KB is assumed to be finite, there are only a finite number of these. Assuming for the moment that $\phi(x)$ only uses the name ${}^{\#}3$, for the above KB, this gives us the first three terms of the disjunction:

$$\begin{array}{l}((x = {}^{\#}1) \wedge \text{RES}[\![\phi({}^{\#}1), \text{KB}]\!]), \\ ((x = {}^{\#}2) \wedge \text{RES}[\![\phi({}^{\#}2), \text{KB}]\!]), \\ ((x = {}^{\#}3) \wedge \text{RES}[\![\phi({}^{\#}3), \text{KB}]\!]).\end{array}$$

For all the remaining n, that is, for the infinite set of names not appearing in either KB or ϕ, we use Theorem 2.8.8 and its corollaries to establish that we need only consider a single name, since all such names will behave the same. In other words, instead of asking if $\phi(n)$ is known for an infinite set of names, we choose a single new name n', and ask if $\phi(n')$ is known. However, we cannot simply use the disjunct

$$((x = n') \wedge \text{RES}[\![\phi(n'), \text{KB}]\!]),$$

since this is what we do when n' appears normally in the KB or ϕ (like ${}^{\#}1$). Rather, we construct the final disjunct so that it does not mention n' directly: instead of $(x = n')$ we use

$$((x \neq {}^{\#}1) \wedge (x \neq {}^{\#}2) \wedge (x \neq {}^{\#}3)),$$

and instead of $\text{RES}[\![\phi(n'), \text{KB}]\!]$, which might end up containing n', we use

$$\text{RES}[\![\phi(n'), \text{KB}]\!]_x^{n'}$$

which (abusing notation somewhat) means replacing any n' that occurs by the free variable x. Putting all this together, then, for a KB that uses just ${}^{\#}1$ and ${}^{\#}2$ and a ϕ that uses just ${}^{\#}3$,

we get

$$\begin{aligned}&\mathrm{RES}[\![\phi(x), \mathrm{KB}]\!] = \\ &\quad ((x = {}^{\#}1) \wedge \mathrm{RES}[\![\phi({}^{\#}1), \mathrm{KB}]\!]) \ \vee \\ &\quad ((x = {}^{\#}2) \wedge \mathrm{RES}[\![\phi({}^{\#}2), \mathrm{KB}]\!]) \ \vee \\ &\quad ((x = {}^{\#}3) \wedge \mathrm{RES}[\![\phi({}^{\#}3), \mathrm{KB}]\!]) \ \vee \\ &\quad ((x \neq {}^{\#}1) \wedge (x \neq {}^{\#}2) \wedge (x \neq {}^{\#}3) \wedge \mathrm{RES}[\![\phi(n'), \mathrm{KB}]\!]^{n'}_{x}),\end{aligned}$$

where n' is some name other than ${}^{\#}1$, ${}^{\#}2$, or ${}^{\#}3$. For example, for the above KB, we have that

$$\begin{aligned}&\mathrm{RES}[\![P(x), \mathrm{KB}]\!] = \\ &\quad ((x = {}^{\#}1) \wedge \textsc{true}) \ \vee \\ &\quad ((x = {}^{\#}2) \wedge \textsc{true}) \ \vee \\ &\quad ((x \neq {}^{\#}1) \wedge (x \neq {}^{\#}2) \wedge \textsc{false}),\end{aligned}$$

which correctly simplifies to $((x = {}^{\#}1) \vee (x = {}^{\#}2))$. If instead the KB had been of the form $\forall x[(x \neq {}^{\#}3) \supset P(x)]$, then we would have had

$$\begin{aligned}&\mathrm{RES}[\![P(x), \mathrm{KB}]\!] = \\ &\quad ((x = {}^{\#}3) \wedge \textsc{false}) \ \vee \ ((x \neq {}^{\#}3) \wedge \textsc{true}),\end{aligned}$$

which simplifies to $(x \neq {}^{\#}3)$, as desired. Further examples are in the exercises.

We now provide the definition of RES in its full generality:

Definition 7.2.1: Let ϕ be an objective formula and KB be a finite set of objective sentences. Suppose that $n_1, \ldots, n_k$, are all the names in ϕ or in KB, and that n' is some name that does not appear in ϕ or in KB. Then $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]$ is defined by:

1. If ϕ has no free variables, then $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]$ is
   TRUE, if $\mathrm{KB} \models \phi$, and FALSE, otherwise.
2. If x is a free variable in ϕ, then $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]$ is
   $((x = n_1) \wedge \mathrm{RES}[\![\phi^x_{n_1}, \mathrm{KB}]\!]) \ \vee \ldots$
   $((x = n_k) \wedge \mathrm{RES}[\![\phi^x_{n_k}, \mathrm{KB}]\!]) \ \vee$
   $((x \neq n_1) \wedge \ldots \wedge (x \neq n_k) \wedge \mathrm{RES}[\![\phi^x_{n'}, \mathrm{KB}]\!]^{n'}_{x})$.

To make this definition completely determinate, we can choose n' and x to be the first (in lexicographic order) standard name and variable that satisfy their respective criterion.

The main property of this definition that we require is:

Lemma 7.2.2: *For any finite KB with $e = \Re[\![\mathrm{KB}]\!]$, any objective formula ϕ with free variables $x_1, \ldots, x_k$, and any standard names $n_1, \ldots, n_k$,*

$$e \models \boldsymbol{K}\phi^{x_1 \cdots x_k}_{n_1 \cdots n_k} \quad \textit{iff} \quad \models \mathrm{RES}[\![\phi, \mathrm{KB}]\!]^{x_1 \cdots x_k}_{n_1 \cdots n_k}.$$

Proof: Since $e \models \boldsymbol{K}\phi_{n_1}^{x_1}\cdots_{n_k}^{x_k}$ iff $\models (\mathrm{KB} \supset \phi_{n_1}^{x_1}\cdots_{n_k}^{x_k})$ by Theorem 6.5.2, it suffices to show that

$$\models \mathrm{RES}[\![\phi, \mathrm{KB}]\!]_{n_1}^{x_1}\cdots_{n_k}^{x_k} \quad \text{iff} \quad \models (\mathrm{KB} \supset \phi)_{n_1}^{x_1}\cdots_{n_k}^{x_k}.$$

The proof is by induction on the number of free variables in ϕ.

If ϕ has no free variables, the lemma clearly holds since $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]$ will be valid iff it is equal to TRUE, which happens iff $(\mathrm{KB} \supset \phi)$ is valid.

Now suppose that ϕ has k free variables, and that by induction, for any n we have that $\phi_n^{x_1}$ satisfies the lemma. Now consider $\mathrm{RES}[\![\phi, \mathrm{KB}]\!]_{n_1 n_2}^{x_1 x_2}\cdots_{n_k}^{x_k}$, and call this sentence ψ. Looking at the name n_1, there are two cases to consider, depending on whether or not n_1 appears in KB or ϕ. If it does appear, all disjuncts in ψ but the one naming n_1 simplify to false, and so $\models \psi$ iff $\models \mathrm{RES}[\![\phi_{n_1}^{x_1}, \mathrm{KB}]\!]_{n_2}^{x_2}\cdots_{n_k}^{x_k}$, which by induction happens iff $\models (\mathrm{KB} \supset \phi_{n_1}^{x_1})_{n_2}^{x_2}\cdots_{n_k}^{x_k}$, and so the lemma is satisfied.

If on the other hand, n_1 does not appear in KB or ϕ, then all but the last disjunct in ψ simplifies to false, and so $\models \psi$ iff $\models \mathrm{RES}[\![\phi_{n'}^{x_1}, \mathrm{KB}]\!]_{x_1 n_1 n_2}^{n' x_1 x_2}\cdots_{n_k}^{x_k}$, where n' is some name that also does not appear in either KB or ϕ. Now consider the formula $\mathrm{RES}[\![\phi_{n'}^{x_1}, \mathrm{KB}]\!]_{x_1}^{n'}$. A trivial induction argument shows that this objective formula does not contain either n_1 or n', since RES does not introduce any new names in its result. Now we will apply Corollary 2.8.9 using a bijection * that swaps the names n_1 and n' but leaves all other names unchanged. We get that

$$\models \mathrm{RES}[\![\phi_{n'}^{x_1}, \mathrm{KB}]\!]_{x_1 n_1 n_2}^{n' x_1 x_2}\cdots_{n_k}^{x_k} \quad \text{iff} \quad \models \mathrm{RES}[\![\phi_{n'}^{x_1}, \mathrm{KB}]\!]_{x_1 n' n_2{}^*}^{n' x_1 x_2}\cdots_{n_k{}^*}^{x_k}.$$

But the formula $\mathrm{RES}[\![\phi_{n'}^{x_1}, \mathrm{KB}]\!]_{x_1 n'}^{n' x_1}$ is just $\mathrm{RES}[\![\phi_{n'}^{x_1}, \mathrm{KB}]\!]$, and by induction,

$$\models \mathrm{RES}[\![\phi_{n'}^{x_1}, \mathrm{KB}]\!]_{n_2{}^*}^{x_2}\cdots_{n_k{}^*}^{x_k} \quad \text{iff} \quad \models (\mathrm{KB} \supset \phi)_{n' n_2{}^*}^{x_1 x_2}\cdots_{n_k{}^*}^{x_k}.$$

Again applying Corollary 2.8.9 we have that

$$\models (\mathrm{KB} \supset \phi)_{n' n_2{}^*}^{x_1 x_2}\cdots_{n_k{}^*}^{x_k} \quad \text{iff} \quad \models (\mathrm{KB} \supset \phi)_{n_1 n_2}^{x_1 x_2}\cdots_{n_k}^{x_k}$$

as desired, which completes the proof. ∎

This lemma shows that RES properly captures the known instances of ϕ. As a corollary to this, we have

Corollary 7.2.3: *Let KB, e, ϕ and the names n_i be as in the lemma. Let w be an arbitrary world state. Then $e \models \boldsymbol{K}\phi_{n_1}^{x_1}\cdots_{n_k}^{x_k}$ iff $w \models \mathrm{RES}[\![\phi, \mathrm{KB}]\!]_{n_1}^{x_1}\cdots_{n_k}^{x_k}$.*

The proof is that because the result of RES is a wff that does not use predicate symbols, function symbols, or $\boldsymbol{K}$ operators, its instances are either valid or their negations are valid.

7.3 Reducing arbitrary sentences to objective terms

Now that we have seen how we can replace $\boldsymbol{K}\phi$ by an objective wff that correctly represents its instances, we need to reconsider more precisely the idea discussed earlier of replacing all such subwffs in a sentence, so that the result can then be conjoined with a KB. We call this operation *reducing* a formula to objective terms, and for any wff α of $\mathcal{KL}$, we use the notation $\|\alpha\|_{\mathrm{KB}}$ to mean the objective reduction α with respect to KB. Formally the definition is as follows:

Definition 7.3.1: Given a finite KB and α an arbitrary wff of $\mathcal{KL}$, $\|\alpha\|_{\mathrm{KB}}$ is the objective wff defined by

$$\|\alpha\|_{\mathrm{KB}} = \alpha, \quad \text{when } \alpha \text{ is objective;}$$

$$\|\neg\alpha\|_{\mathrm{KB}} = \neg\|\alpha\|_{\mathrm{KB}};$$

$$\|(\alpha \wedge \beta)\|_{\mathrm{KB}} = (\|\alpha\|_{\mathrm{KB}} \wedge \|\beta\|_{\mathrm{KB}});$$

$$\|\forall x\alpha\|_{\mathrm{KB}} = \forall x\|\alpha\|_{\mathrm{KB}};$$

$$\|\boldsymbol{K}\alpha\|_{\mathrm{KB}} = \mathrm{RES}[\![\|\alpha\|_{\mathrm{KB}}, \mathrm{KB}]\!].$$

Note that this recursive definition works from the "inside out" in that we first reduce the argument to $\boldsymbol{K}$ to objective terms before applying RES. If the argument to $\boldsymbol{K}$ happens to be objective, $\|\cdot\|$ will not change it, and RES is called directly. Otherwise, that is, when we have nested $\boldsymbol{K}$ operators, the call to $\|\cdot\|$ produces an objective wff which can then be passed to RES.[1]

For example, if KB is $\{P(^{\#}1), P(^{\#}2)\}$ as we had before, then

$$\|\exists x[P(x) \wedge \neg\boldsymbol{K}P(x)]\|_{\mathrm{KB}} =$$
$$\exists x[P(x) \wedge \neg((x = {}^{\#}1) \vee (x = {}^{\#}2))].$$

Note that the only part of the sentence that gets changed by $\|\cdot\|$ is the part involving $\boldsymbol{K}$. Unlike RES, the goal of $\|\cdot\|$ is not to produce the known instances of a wff; rather, it is to take an arbitrary wff and produce an objective version that is true in exactly the same world states by encoding what is needed from the epistemic state. More precisely, we have the following:

Lemma 7.3.2: *For any finite KB with* $e = \Re[\![\mathrm{KB}]\!]$, *any world state* w, *any formula* α *of*

1 Strictly speaking, the well-formedness of this definition should not simply be assumed; we need to prove (by induction) that $\|\cdot\|$ always returns an objective formula.

$\mathcal{KL}$ with free variables $x_1, \ldots, x_k$, and any standard names $n_1, \ldots, n_k$,

$$e, w \models \alpha^{x_1}_{n_1} \cdots {}^{x_k}_{n_k} \quad \textit{iff} \quad w \models \|\alpha\|_{\mathrm{KB}} {}^{x_1}_{n_1} \cdots {}^{x_k}_{n_k}.$$

Proof: The proof is by induction on the structure of α. If α is atomic or an equality, the lemma clearly holds since α is then objective. The lemma also holds by induction for negations, conjunctions, and quantifications. Now consider the formula $\boldsymbol{K}\alpha$. We have that

$e \models \boldsymbol{K}\alpha^{x_1}_{n_1} \cdots {}^{x_k}_{n_k}$

iff for every $w' \in e$, we have

$e, w' \models \alpha^{x_1}_{n_1} \cdots {}^{x_k}_{n_k}$

iff (by induction) we have for every $w' \in e$,

$w' \models \|\alpha\|_{\mathrm{KB}} {}^{x_1}_{n_1} \cdots {}^{x_k}_{n_k}$

iff we have

$e \models \boldsymbol{K}\|\alpha\|_{\mathrm{KB}} {}^{x_1}_{n_1} \cdots {}^{x_k}_{n_k}$.

Since the formula within the $\boldsymbol{K}$ operator here is now objective, by Corollary 7.2.3, this holds iff

$w \models \mathrm{RES}[\![\|\alpha\|_{\mathrm{KB}}, \mathrm{KB}]\!]^{x_1}_{n_1} \cdots {}^{x_k}_{n_k}$

which, by definition of $\|\cdot\|$, is the same as

$w \models \|\boldsymbol{K}\alpha\|_{\mathrm{KB}} {}^{x_1}_{n_1} \cdots {}^{x_k}_{n_k}$. ∎

So this lemma shows that $\|\cdot\|$ preserves the truth value of α with respect to the world state, but removes the dependency of α on the epistemic state. This is just what we need to represent a new epistemic state in objective terms.

7.4 TELL and ASK at the symbol level

With the lemma of the previous section, we saw how we could take an arbitrary formula of $\mathcal{KL}$, and reduce it to objective form in a way that captures its dependency on the epistemic state. This will allow us to deal with **TELL** and **ASK** completely in objective terms, provided that we start with a finitely representable objective state.

Theorem 7.4.1: [The Representation Theorem] *Let KB be any finite set of objective sentences and α be any sentence of $\mathcal{KL}$. Then:*

1. $\mathbf{TELL}[\alpha, \Re[\![\mathrm{KB}]\!]] = \Re[\![(\mathrm{KB} \wedge \|\alpha\|_{\mathrm{KB}})]\!]$.
2. $\mathbf{ASK}[\alpha, \Re[\![\mathrm{KB}]\!]] = \textit{yes} \quad \textit{iff} \quad \mathrm{KB} \models \|\alpha\|_{\mathrm{KB}}$.

The proof is immediate from the fact that for any w, and for $e = \Re[\![\mathrm{KB}]\!]$,

$$e, w \models \alpha \quad \text{iff} \quad w \models \|\alpha\|_{\mathrm{KB}},$$

which is just Lemma 7.3.2 when α has no free variables. We can also state a variant for **ASK**:

Corollary 7.4.2: *Under the same conditions as the Theorem,*

$$\mathbf{ASK}[\alpha, \Re[\![\mathrm{KB}]\!]] = \textit{yes} \;\; \textit{iff} \;\; \|\boldsymbol{K}\alpha\|_{\mathrm{KB}} \textit{ is } \mathrm{TRUE}.$$

This important theorem tells us that the result of a **TELL** can always be represented by conjoining an objective sentence to the KB, and that an **ASK** can always be calculated in terms of the (objective) logical implications of the KB. Moreover, as can be seen by examining the definition of $\| \cdot \|$ and RES, the reduction to objective terms itself can be done using only the (objective) logical implications of the KB. The conclusion: we can calculate the answers to **TELL** and to **ASK** for arbitrary sentences of $\mathcal{KL}$ using ordinary first-order theorem proving.

Of course, this does not make it easy to perform these operations, since in general, it is *impossible* to calculate the objective logical implication of a KB. Moreover, the way RES was defined was not particularly realistic, since it involved constructing a formula that could be as large as twice the total number of constants in the KB. But the representation theorem at least shows that the operation is definable in terms of these ordinary first-order operations, which opens the door to possible optimizations in special cases.

Another way of looking at this theorem is to consider a symbolic "implementation" of **TELL** and **ASK** which works directly on representations of the epistemic states. Call these procedures TELL′ and ASK′ respectively, where

- ASK′[α, KB] is defined as
  1. Calculate $\|\alpha\|_{\mathrm{KB}}$ using the recursive definition. Call this ϕ.
  2. Test if (KB $\models \phi$); if it does, return *yes*; otherwise, *no*.
- TELL′[α, KB] is defined as
  1. Calculate $\|\alpha\|_{\mathrm{KB}}$ using the recursive definition. Call this ϕ.
  2. Return (KB $\wedge \phi$).

The representation theorem can thought of as a proof of *correctness* for these symbol level procedures.

7.5 The example KB reconsidered

Let us now return to the example KB introduced in Section 5.7 to examine the workings of the representation theorem. We begin by looking at an example of **ASK**, in particular, Question 8:

8. $\exists x[\mathit{Teach}(x, \mathit{sue}) \wedge \neg \boldsymbol{K}\mathit{Teach}(x, \mathit{sue})]$ TRUE

We have already considered why the answer should be *yes* on semantic grounds. In terms of the representation theorem, we need to reduce the question to objective terms. To do so, we need to calculate the known instances of $\mathit{Teach}(x, \mathit{sue})$. If we apply the definition of RES, we get that

$$
\begin{aligned}
&\mathrm{RES}[\![\mathit{Teach}(x, \mathit{sue}), \mathrm{KB}]\!] = \\
&\quad ((x = \mathit{tom}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{tom}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sam}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{sam}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{tina}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{tina}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{tara}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{tara}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sue}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{sue}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{ted}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{ted}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sara}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{sara}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sandy}) \wedge \mathrm{RES}[\![\mathit{Teach}(\mathit{sandy}, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad ((x \neq \mathit{tom}) \wedge (x \neq \mathit{sam}) \wedge (x \neq \mathit{tina}) \wedge (x \neq \mathit{tara}) \wedge \\
&\qquad (x \neq \mathit{sue}) \wedge (x \neq \mathit{ted}) \wedge (x \neq \mathit{sara}) \wedge (x \neq \mathit{sandy}) \wedge \\
&\qquad\quad \mathrm{RES}[\![\mathit{Teach}(\mathit{tania}, \mathit{sue}), \mathrm{KB}]\!]_x^{\mathit{tania}}),
\end{aligned}
$$

where *tania* is the chosen new name not appearing in the KB. Here all the recursive calls are of the form $\mathit{Teach}(n, \mathit{sue})$ and have no free variables. So they will either return TRUE or FALSE, depending on whether $\mathrm{KB} \models \mathit{Teach}(n, \mathit{sue})$. For all but the standard name *tina*, the answer will be FALSE. So simplifying, we have

$$\mathrm{RES}[\![\mathit{Teach}(x, \mathit{sue}), \mathrm{KB}]\!] = (x = \mathit{tina}),$$

meaning that Tina is the only *known* teacher of Sue. Then we get that

$$
\begin{aligned}
&\|\exists x \mathit{Teach}(x, \mathit{sue}) \wedge \neg \boldsymbol{K}\mathit{Teach}(x, \mathit{sue})\|_{\mathrm{KB}} = \\
&\quad \exists x[\mathit{Teach}(x, \mathit{sue}) \wedge \neg(x = \mathit{tina})].
\end{aligned}
$$

So the original question reduces to whether or not Sue has a teacher apart from Tina. The answer depends on whether or not

$$\mathrm{KB} \models \exists x[\mathit{Teach}(x, \mathit{sue}) \wedge \neg(x = \mathit{tina})].$$

Since the entailment holds (because of what is known about Tom and Ted), the answer is *yes*, as desired.

Notice the two step operation: we first reduce the question to objective terms (using the implications of the KB) and then we determine if the result is implied by the KB.

Alternatively, we could have used the corollary to the representation theorem, and simply calculated

$$\|\boldsymbol{K}\exists x[\mathit{Teach}(x, \mathit{sue}) \wedge \neg \boldsymbol{K}\mathit{Teach}(x, \mathit{sue})]\|_{\mathrm{KB}}.$$

First the argument to the outermost $\boldsymbol{K}$ must be reduced, which produces

$$\exists x[\mathit{Teach}(x, \mathit{sue}) \wedge \neg(x = \mathit{tina})],$$

as before because of $\mathrm{RES}[\![\mathit{Teach}(x, \mathit{sue}), \mathrm{KB}]\!]$. Then we need to apply RES to this sentence which, because it is implied by the KB, gives us TRUE, and so once again the answer is *yes*.

As a second example, consider Question 11:

11. $\exists y \boldsymbol{K}\forall x[\mathit{Teach}(x, y) \supset \boldsymbol{K}\mathit{Teach}(x, y)]$ TRUE

We need to apply RES to the innermost formula dominated by a $\boldsymbol{K}$, $\mathit{Teach}(x, y)$. This has two free variables, and so the depth of recursion will be two. Assuming the y is used first, say, we get

$$\begin{aligned}
&\mathrm{RES}[\![\mathit{Teach}(x, y), \mathrm{KB}]\!] = \\
&\quad((y = \mathit{tom}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{tom}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y = \mathit{sam}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{sam}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y = \mathit{tina}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{tina}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y = \mathit{tara}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{tara}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y = \mathit{sue}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{sue}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y = \mathit{ted}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{ted}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y = \mathit{sara}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{sara}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y = \mathit{sandy}) \wedge \mathrm{RES}[\![\mathit{Teach}(x, \mathit{sandy}), \mathrm{KB}]\!]) \ \vee \\
&\quad((y \neq \mathit{tom}) \wedge (y \neq \mathit{sam}) \wedge (y \neq \mathit{tina}) \wedge (y \neq \mathit{tara}) \wedge \\
&\qquad(y \neq \mathit{sue}) \wedge (y \neq \mathit{ted}) \wedge (y \neq \mathit{sara}) \wedge (y \neq \mathit{sandy}) \wedge \\
&\qquad\quad \mathrm{RES}[\![\mathit{Teach}(x, \mathit{sally}), \mathrm{KB}]\!]^{\mathit{sally}}_{x}),
\end{aligned}$$

where *sally* is the chosen new name that does not appear in the KB. For each name n, we need to calculate $\mathrm{RES}[\![\mathit{Teach}(x, n), \mathrm{KB}]\!]$ which, as in the previous example, gives us the known teachers of n. For example, if n has no known teachers, such as for *sally* or *tom*, we

get the following:

$$
\begin{aligned}
&\text{RES}[\![\mathit{Teach}(x, \mathit{sally}), \text{KB}]\!] = \\
&\quad ((x = \mathit{tom}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{tom}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sam}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{sam}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{tina}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{tina}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{tara}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{tara}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sue}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{sue}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{ted}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{ted}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sara}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{sara}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sandy}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{sandy}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x = \mathit{sally}) \wedge \text{RES}[\![\mathit{Teach}(\mathit{sally}, \mathit{sally}), \text{KB}]\!]) \ \vee \\
&\quad ((x \neq \mathit{tom}) \wedge (x \neq \mathit{sam}) \wedge (x \neq \mathit{tina}) \wedge (x \neq \mathit{tara}) \wedge (x \neq \mathit{sue}) \wedge \\
&\qquad (x \neq \mathit{ted}) \wedge (x \neq \mathit{sara}) \wedge (x \neq \mathit{sandy}) \wedge (x \neq \mathit{sally}) \wedge \\
&\qquad\quad \text{RES}[\![\mathit{Teach}(\mathit{tony}, \mathit{sara}), \text{KB}]\!]_x^{\mathit{tony}}),
\end{aligned}
$$

where *tony* is the new name. Note that in this case, *tony* must be distinct from all the names in the KB and from the name of *sally* as well, which was introduced earlier in the recursion. Here each of the recursive calls returns FALSE, and so RES$[\![\mathit{Teach}(x, \mathit{sally})]\!]$ simplifies to FALSE. So overall, after simplification, we get

$$
\begin{aligned}
&\text{RES}[\![\mathit{Teach}(x, y), \text{KB}]\!] = \\
&\quad ((y = \mathit{sam}) \wedge (x = \mathit{tom})) \ \vee \\
&\quad ((y = \mathit{sue}) \wedge (x = \mathit{tina})) \ \vee \\
&\quad ((y = \mathit{sandy}) \wedge (x = \mathit{ted})),
\end{aligned}
$$

which captures all the known instances of the *Teach* predicate. Given this, we can reduce the formula

$$\forall x[\mathit{Teach}(x, y) \supset \boldsymbol{K}\mathit{Teach}(x, y)],$$

which says that all of the teachers of y are known, to

$$
\begin{aligned}
&\forall x[\mathit{Teach}(x, y) \ \supset \ ((y = \mathit{sam}) \wedge (x = \mathit{tom})) \vee \\
&\quad ((y = \mathit{sue}) \wedge (x = \mathit{tina})) \vee ((y = \mathit{sandy}) \wedge (x = \mathit{ted}))].
\end{aligned}
$$

This formula, call it ψ, will now become the argument to RES for the outermost $\boldsymbol{K}$ operator. It has a single free variable y, and we wish to find names n for which $\psi(n)$ is known to be

true. As usual, we need to consider all the names in the KB:

$$\begin{aligned}
&\mathrm{RES}[\![\psi(y), \mathrm{KB}]\!] = \\
&\quad ((y = tom) \wedge \mathrm{RES}[\![\psi(tom), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y = sam) \wedge \mathrm{RES}[\![\psi(sam), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y = tina) \wedge \mathrm{RES}[\![\psi(tina), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y = tara) \wedge \mathrm{RES}[\![\psi(tara), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y = sue) \wedge \mathrm{RES}[\![\psi(sue), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y = ted) \wedge \mathrm{RES}[\![\psi(ted), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y = sara) \wedge \mathrm{RES}[\![\psi(sara), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y = sandy) \wedge \mathrm{RES}[\![\psi(sandy), \mathrm{KB}]\!]) \ \vee \\
&\quad ((y \neq tom) \wedge (y \neq sam) \wedge (y \neq tina) \wedge (y \neq tina) \wedge \\
&\qquad (y \neq sue) \wedge (y \neq ted) \wedge (y \neq sara) \wedge (y \neq sandy) \wedge \\
&\qquad\quad \mathrm{RES}[\![\phi(steve), \mathrm{KB}]\!]_{y}^{steve}).
\end{aligned}$$

For each name n, the sentence $\psi(n)$ is true if either n has no teachers or n is *sam* and his only teacher is *tom*, or n is *sue* and her only teacher is *tina*, or n is *sandy* and her only teacher is *ted*. The values of n for which this is implied by the KB are *tom*, *tina*, *tara*, *ted* and *steve*, (for which it is known that they have no teachers), and *sandy* (for which it is known that *ted* is her only teacher). For these values of n, $\mathrm{RES}[\![\psi(n), \mathrm{KB}]\!]$ will return TRUE, and for the others, FALSE. Thus, we get that

$$\begin{aligned}
&\mathrm{RES}[\![\psi(y), \mathrm{KB}]\!] = \\
&\quad (y = ted) \vee (y = tom) \vee (y = tina) \vee (y = tara) \vee (y = sandy) \vee \\
&\quad [(y \neq tom) \wedge (y \neq sam) \wedge (y \neq tina) \wedge (y \neq tara) \wedge (y \neq sue) \wedge \\
&\qquad (y \neq ted) \wedge (y \neq sara) \wedge (y \neq sandy)]
\end{aligned}$$

which simplifies to

$$[(y \neq sam) \wedge (y \neq sue) \wedge (y \neq sara)].$$

Thus for anyone but *sam*, *sue*, or *sara*, it is known that all of the teachers are known (for all but *sandy*, this is because they are known to not have any teachers). So ψ is an example of a non-trivial formula with infinitely many known instances, captured by the last disjunct of RES using inequalities.

Finally, we can reduce the original question

$$\exists y \boldsymbol{K} \forall x [Teach(x, y) \supset \boldsymbol{K} Teach(x, y)]$$

to

$$\exists y.(y \neq sam) \wedge (y \neq sue) \wedge (y \neq sara).$$

This will be answered *yes*, since it is implied by the KB, because of what is known about Sandy. She is the only individual that has a teacher and for which it is known that all of her teachers are known.

As a final example in this chapter, we will consider the effect of the Assertion 3 in terms of the representation theorem:

3. $\forall x[\mathit{Teach}(x, \mathit{sara}) \supset \exists y \boldsymbol{K}\mathit{Teach}(x, y)]$ $[e_2 \rightarrow e_3]$

Earlier, we considered the result of this assertion after two preliminary assertions. To simplify here, we assume that this is being asserted starting in the original KB.

What this sentence says is that anyone who teaches Sara must have someone they are known to teach. To reduce this sentence, we must calculate $\mathrm{RES}[\![\mathit{Teach}(x, y), \mathrm{KB}]\!]$, which we did in the previous example. Using that result, the assertion reduces to

$$\begin{aligned}&\forall x(\mathit{Teach}(x, \mathit{sara}) \supset \exists y\\ &\quad[((y = \mathit{sam}) \wedge (x = \mathit{tom})) \vee\\ &\quad((y = \mathit{sue}) \wedge (x = \mathit{tina})) \vee\\ &\quad((y = \mathit{sandy}) \wedge (x = \mathit{ted}))]),\end{aligned}$$

which simplifies to

$$\forall x(\mathit{Teach}(x, \mathit{sara}) \supset [(x = \mathit{tom}) \vee (x = \mathit{tina}) \vee (x = \mathit{ted})]).$$

This sentence is objective and can be conjoined to the KB. So the effect of the **TELL** is to assert that Sara's teachers must be among Tom, Tina, or Ted, the only individuals with a known student.

7.6 Wh-questions at the symbol level

In Chapter 5, we introduced a new interaction operation **WH-ASK** which returned the known instances of a formula, defined by

$$\textbf{WH-ASK}[\alpha[\vec{x}], e] = \{\vec{n} \mid e \models \boldsymbol{K}\alpha[\vec{n}]\}.$$

We also mentioned that this set of standard names could be infinite, which presents a problem from an implementation standpoint. But $\|\cdot\|$, which can be used to represent the known instances of a formula, provides a perfect solution to this problem:

$$\textbf{WH-ASK}[\alpha, \Re[\![\mathrm{KB}]\!]] = \|\alpha\|_{\mathrm{KB}}.$$

Instead of returning a possibly infinite set of known instances of α, we return instead a finite formula from which as many standard names as desired can easily be extracted. From Lemma 7.3.2, it follows that these standard names are precisely the known instances of α.

We also considered in Chapter 5, an interaction operation **DESCRIBE** which returned terms known to be co-referential with a given standard name, defined by

$$\textbf{DESCRIBE}[n, e] = \{t \mid e \models \boldsymbol{K}(t = n)\}.$$

Again, this presents an implementation problem since for a KB like $\{\forall x.x = f(x)\}$, the standard name $^{\#}1$ is known to be co-referential with an infinite number of other terms:

$$\{^{\#}1,\ f(^{\#}1),\ f(f(^{\#}1)),\ f(f(f(^{\#}1))),\ \ldots\}.$$

A suggestion here is that instead of returning all co-referential terms, we only return co-referential *primitive* terms, that is, containing exactly one function or constant symbol:

$$\textbf{DESCRIBE}[n, e] = \{t \mid t \text{ is primitive and } e \models \boldsymbol{K}(t = n)\}.$$

For example, asked to describe a standard name like $^{\#}23$, we might get the set

$$\{\mathit{jake},\ \mathit{jack},\ \mathit{jackie},\ \mathit{best\_friend}(^{\#}13),\ \mathit{first\_child}(^{\#}5, {}^{\#}79)\}$$

of primitive co-referring terms. We are then free to further elaborate on this set by describing any of the standard names mentioned, and so on to any depth. We leave it as an exercise to show that **DESCRIBE** as redefined above always returns a finite set.

7.7 Bibliographic notes

As discussed in Chapter 4, knowledge has many of the closure properties of entailment, validity, or provability. Further, as seen in Chapter 6, what is known objectively in the epistemic state represented by some KB is precisely the logical entailments of that KB. At its simplest, the Representation Theorem of the current chapter is based on the idea of going through a formula and replacing knowledge of an objective sentence by either TRUE or FALSE according to whether the sentence is entailed by the given KB. This idea is then generalized to non-objective knowledge by working recursively on formulas from the inside out. Finally, we use standard names and equality to deal with formulas with free variables and quantifying in. An early version of these ideas appeared in [111] and [113]. The Representation Theorem was subsequently used to describe integrity constraints on databases in [160]. We will see another use in the final chapter of the book in the context of reasoning about action (see also [20, 172], for example).

7.8 Exercises

1. Show that for any KB, $\text{RES}[\![(x = {}^{\#}1), \text{KB}]\!]$ is equivalent to $(x = {}^{\#}1)$.
2. Consider the KB that is the conjunction of
   $$\forall y.R(^{\#}1, y) \equiv (y = {}^{\#}1) \vee (y = {}^{\#}2)$$
   $$\forall y.R(^{\#}2, y) \equiv (y \neq {}^{\#}2) \wedge (y \neq {}^{\#}3)$$
   $$\forall y.\neg R(^{\#}3, y)$$
   $$\forall x, \forall y.((x \neq {}^{\#}1) \wedge (x \neq {}^{\#}2) \wedge (x \neq {}^{\#}3)) \supset (R(x, y) \equiv (x = y))$$

Calculate each of the following: RES[$[\![R(^{\#}1, y), \mathrm{KB}]\!]$; RES[$[\![R(^{\#}5, y), \mathrm{KB}]\!]$; RES[$[\![R(x, ^{\#}2), \mathrm{KB}]\!]$; and RES[$[\![R(x, ^{\#}5), \mathrm{KB}]\!]$.

3. Show using the representation theorem, why the answer to the question

   $$\exists x . Teach(x, sam) \wedge \neg \boldsymbol{K} Teach(x, sam)$$

   for the example KB in Section 5.7 is UNKNOWN.

4. The definition of RES requires constructing a formula using every standard name mentioned in the KB. Describe a more practical class of KB's and queries where it would not be necessary to enumerate all the standard names in the KB.

5. Call an epistemic state *quasi-finitely representable* if it can be represented by a KB (finite or infinite) that uses only finitely many standard names.

   (a) Prove that the representation theorem works for quasi-finite epistemic states, and hence that these are closed under **TELL**.

   (b) Prove that the sentence π of Theorem 6.5.1 is not satisfied by a quasi-finite epistemic state, and hence that the logic of $\mathcal{KL}$ requires epistemic states that are not quasi-finite.

6. Give an example where RES would return an incorrect value if the standard names used in the definition ranged only over those in the KB, but not over those in the first argument.

7. Prove that for a finite KB, **DESCRIBE**$[n, \Re[\![\mathrm{KB}]\!]]$ is always finite.

8 Only-Knowing

In previous chapters, we covered in detail the language $\mathcal{KL}$ and how it could be used as the interface language for **TELL** and **ASK** operations. We also saw how its objective fragment, $\mathcal{L}$, could always be used to represent what was known. In this chapter, we begin the examination of a third use for a logical language: as a specification of the behaviour of a knowledge base under the **TELL** and **ASK** operations.

Since we already have a semantic definition of these operations and, as a result of the Representation Theorem of the previous chapter, an equivalent symbolic characterization, why do we need yet another specification? The answer is simply that this logical specification will allow us to generalize very nicely the **TELL** and **ASK** operations in a way that will make a close connection to some of the work in nonmonotonic reasoning, explored in chapters 10 and 11.

8.1 The logic of answers

Suppose we start with an epistemic state e represented by $P(^{\#}1)$. In this state, we have, for example, that

$$\mathbf{ASK}[\exists x\, P(x), e] = \mathit{yes}.$$

One question we can ask about this answer is this:

What property of the logic of $\mathcal{KL}$ tells us that this answer is correct?

By looking at the definition of **ASK**, we can see that all the world states in e satisfy $\exists x\, P(x)$. In other words, we answer *yes* because any w that satisfies the KB must also satisfy $\exists x\, P(x)$. This is just another way of saying that we will answer *yes* for any α for which

$$\models (\mathrm{KB} \supset \alpha),$$

as expected.

But there is clearly more to the story of **ASK**. For example, we also have that

$$\mathbf{ASK}[\boldsymbol{K}\exists x\, P(x), e] = \mathit{yes}$$

and even

$$\mathbf{ASK}[\exists x\, \boldsymbol{K} P(x), e] = \mathit{yes},$$

where the α here is not implied by the KB. In this case, the answer arises due to introspection: *knowing* the α is implied by knowing the KB. Thus, we will answer *yes* for any α for which

$$\models (\boldsymbol{K}\mathrm{KB} \supset \boldsymbol{K}\alpha).$$

The reason is that since $e \models \boldsymbol{K}\text{KB}$, we get that $e \models \boldsymbol{K}\alpha$ and thus **ASK** must return *yes*. This also subsumes the previous case since if KB implies α, then $\boldsymbol{K}\text{KB}$ implies $\boldsymbol{K}\alpha$.

Although this explanation handles positive introspection properly, it does not work for negative introspection. For example, we also have that

$$\textbf{ASK}[\neg \boldsymbol{K} P(^{\#}2), e] = \mathit{yes}.$$

What property of the logic of $\mathcal{KL}$ explains this? In this case, knowing KB does not imply knowing α. In fact there is nothing in KB that suggests anything one way or another about $P(^{\#}2)$. Just because $P(^{\#}1)$ is known, $P(^{\#}2)$ may or may not be known. If $P(^{\#}2)$ is in fact known, then so will be $\boldsymbol{K}P(^{\#}2)$, by positive introspection; if it is not known, then $\neg \boldsymbol{K}P(^{\#}2)$ will be known by negative introspection.

How then did **ASK** come to settle on the second case? Informally, the answer is that because the negation of $P(^{\#}2)$ is consistent with what is known, that is, because

$$\not\models (P(^{\#}1) \supset P(^{\#}2)),$$

$P(^{\#}2)$ is not known. In other words, although there is nothing about $P(^{\#}2)$ implied by knowing $P(^{\#}1)$, if this is *all* that is known, then we can say something about $P(^{\#}2)$, namely that it is not known.

To make this distinction, we need to clearly separate the difference between saying that α is known and α is all that is known. Of course, we never mean that α is the unique single sentence known to be true, since at the very least we will know the logical consequences of α and other formulas by positive introspection. But when we say that a state e is represented by the sentences in KB, we are saying more than just that these sentences are known. We are implicitly saying that these represent all that is known.

The difference between the two readings shows up most clearly with objective sentences. If KB and ϕ are objective, and KB does not imply ϕ, then if KB is known, ϕ may or may not be known; but if KB is all that is known, ϕ is not known, and so $\neg \boldsymbol{K}\phi$ will be known by negative introspection, as above.

But characterizing the answer to **ASK** for non-objective sentences involving negative introspection is somewhat more complex. Rather than try to devise a complicated strategy using satisfiability (or consistency) instead of validity, we will take a very different approach: we will extend the language $\mathcal{KL}$ so that we can distinguish between saying "α is known" and "α is all that is known." As always, we will write the former as $\boldsymbol{K}\alpha$. For the latter, we will introduce a new modal operator $\boldsymbol{O}$ so that $\boldsymbol{O}\alpha$ is read as α is all that is known, or that only α is known. We will also sometimes use the expression "*only-knowing* α." What we will end up establishing is that **ASK** returns *yes* for question α iff

$$\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{K}\alpha),$$

that is, only-knowing the KB implies knowing α. This will subsume the previous two cases above since if the KB is all that is known, then the KB is known. Thus we will have

characterized the behaviour of **ASK** as applied to finitely representable states completely by the valid sentences of this extended logic. And, as we saw in the last chapter, we can restrict our attention to such states when it comes to arguments for **TELL** and **ASK**.

8.2 The language $\mathcal{OL}$

The language $\mathcal{OL}$ has exactly the same syntactic formation rules as that of $\mathcal{KL}$ but with one addition:

- If α is a wff, then $\boldsymbol{O}\alpha$ is one too.

Note that the argument to $\boldsymbol{O}$ need not be objective or even a sentence of $\mathcal{KL}$. For example,

$$\boldsymbol{O}[P(^{\#}1) \wedge \neg \boldsymbol{O}(P(^{\#}2))] \vee \boldsymbol{K}\boldsymbol{O}(P(^{\#}3))$$

is a proper sentence of $\mathcal{OL}$. It is also considered to be a *subjective* sentence of $\mathcal{OL}$, since all predicate and function symbols are within the scope of a modal operator. A sentence of $\mathcal{OL}$ is called *basic* if it is also a sentence of $\mathcal{KL}$ (that is, contains no $\boldsymbol{O}$ operators.)

Turning now to the semantics of $\mathcal{OL}$, we will have the usual rules of interpretation for all the connectives from $\mathcal{KL}$. All we need to do, then, is to specify when $e \models \boldsymbol{O}\alpha$ holds, after which satisfaction, validity, and implication will be as before.

The idea of only-knowing α means knowing no more than α about the world. So α will be all that is known in e when α is known, but e has as little world knowledge as possible. Since, as we discussed in Chapter 3, more knowledge means fewer world states and less knowledge means more world states, we want e to have as many world states as possible, although it clearly cannot contain any where α comes out false. In other words, α is all that is known in e iff e consists of *exactly* the world states where α is true, no more (since α is known) and no less (since it is all that is known).

More formally, we augment the semantic specification of $\mathcal{KL}$ by a single new rule of interpretation:

- $e, w \models \boldsymbol{O}\alpha$ iff for every w', $w' \in e$ iff $e, w' \models \alpha$.

Note that this is (inductively) well-defined for any α in $\mathcal{OL}$. So whereas the semantics of $\boldsymbol{K}$ requires e to be a subset of the states where α is true, that is,

- $e, w \models \boldsymbol{K}\alpha$ iff for every w', if $w' \in e$ then $e, w' \models \alpha$,

the semantics of $\boldsymbol{O}$ requires equality. That is, an "if" has been augmented to an "iff." Thus $\boldsymbol{O}\alpha$ logically implies $\boldsymbol{K}\alpha$, but not vice versa.

Also worth noting is that because we are insisting that e be the set of *all* states where α is true, we need to be careful when it comes to equivalent epistemic states. For example, imagine that we have two completely equivalent states but that one is a subset of another, as we described in Chapter 6. Since these two states know exactly the same basic sentences,

we obviously want them to agree on all sentences of the form $\boldsymbol{O}\alpha$ as well. But by the above definition, they would not. One would only be a subset of the set of states where α was true. Rather than complicate the semantics somehow (using the equivalence relation) to handle this situation, we will simply restrict our attention to *maximal* epistemic states, as we have done in previous chapters. Indeed, with maximal sets it is straightforward to show that the basic beliefs of an epistemic state uniquely determine all beliefs at that state, including those that mention $\boldsymbol{O}$. Let a *basic belief set* Γ be defined just like a belief set in $\mathcal{KL}$, that is, $\Gamma = \{\alpha \mid \alpha \text{ is basic and } e \models \boldsymbol{K}\alpha\}$.

Lemma 8.2.1: *If e and e' are maximal sets that have the same basic belief set, then for any subjective sentence σ of $\mathcal{OL}$, $e \models \sigma$ iff $e' \models \sigma$.*

Proof: Since e and e' have the same belief set, they are equivalent. Since they are both maximal, they must be equal by Theorem 6.1.2, and so satisfy the same subjective sentences. ∎

With that we define satisfiability, validity, and logical implication in $\mathcal{OL}$ just like in $\mathcal{KL}$ except that we explicitly restrict ourselves to maximal sets of worlds only.

8.3 Some properties of $\mathcal{OL}$

The simplest and most common case of only-knowing that we will consider is when the argument is an objective sentence ϕ. Saying $\boldsymbol{O}\phi$ is simply saying that what is known can be finitely represented by ϕ. There is exactly one epistemic state where this is true:

Theorem 8.3.1: *For any objective ϕ, there is a unique maximal e such that $e \models \boldsymbol{O}\phi$.*

Proof: Let $e = \Re[\![\phi]\!]$. Clearly, $e \models \boldsymbol{O}\phi$, and no other e' can contain any other world states or fail to contain those in e. ∎

As a trivial corollary, we have

Corollary 8.3.2: *If ϕ is objective and σ is subjective, then either*

$$\models (\boldsymbol{O}\phi \supset \sigma) \quad \textit{or} \quad \models (\boldsymbol{O}\phi \supset \neg\sigma).$$

So given that only ϕ is known, everything else about the epistemic state is logically implied. Note that this is not true for $\boldsymbol{K}$. If ϕ and ψ are distinct atomic sentences, then we have that

$$\models (\boldsymbol{O}\phi \supset \neg\boldsymbol{K}\psi),$$

yet $\not\models (\boldsymbol{K}\phi \supset \neg\boldsymbol{K}\psi)$ and $\not\models (\boldsymbol{K}\phi \supset \boldsymbol{K}\psi)$. In other words, $\boldsymbol{K}\phi$ leaves open whether or not $\boldsymbol{K}\psi$. We also have:

Theorem 8.3.3: *Suppose ϕ and ψ are objective. Then*

$$\models (\boldsymbol{O}\phi \supset \boldsymbol{K}\psi) \quad \textit{iff} \quad \models (\phi \supset \psi).$$

Proof: Suppose that $\models (\boldsymbol{O}\phi \supset \boldsymbol{K}\psi)$, and that e is such that $e \models \boldsymbol{O}\phi$ as guaranteed by the previous theorem, and so $e \models \boldsymbol{K}\psi$. For any w, if $w \models \phi$, then $w \in e$, and so $w \models \psi$. Conversely, assume that $\models (\phi \supset \psi)$, and so, $\models (\boldsymbol{K}\phi \supset \boldsymbol{K}\psi)$. For any e, if $e \models \boldsymbol{O}\phi$, then $e \models \boldsymbol{K}\phi$, and so $e \models \boldsymbol{K}\psi$. ∎

Finally, notice that nothing in the proof of the theorem depends on ϕ being finite. Hence we obtain the following corollary, which, in a sense, generalizes the concept of only-knowing to arbitrary sets of sentences, a subject we will not pursue further in this book.

Corollary 8.3.4: *For any set of objective sentences Φ, there is a unique maximal e such that for any objective ψ, $e \models \boldsymbol{K}\psi$ iff $\Phi \models \psi$.*

These results give us a complete characterization of which objective sentences are believed, given that all that is known is also objective.

Turning now to only-knowing purely subjective sentences, here the situation is somewhat trivial. If we say "all that is known about the world is σ," and σ is subjective and so doesn't say anything about the world, then nothing is known about the world. So the epistemic state must be e_0. The only other possibility is the inconsistent epistemic state: in this case, for certain σ, such as $\neg\boldsymbol{K}\alpha$, we have that σ is known because every sentence is known, and nothing else need be known to arrive at this inconsistent state, since $\boldsymbol{K}\alpha$ is also true. More precisely, we have:

Theorem 8.3.5: *For any e and subjective σ, $e \models \boldsymbol{O}\sigma$ iff either $e \models \sigma$ and $e = e_0$, or $e \models \neg\sigma$ and $e = \{\}$.*

Proof: Suppose that $e \models \boldsymbol{O}\sigma$. In one case, we have $e \models \sigma$, in which case $e = e_0$ since for every world state w, $e, w \models \sigma$. In the other case, we have $e \models \neg\sigma$, and so for any world state w, $e, w \models \neg\sigma$, and thus $e \models \boldsymbol{K}\neg\sigma$. However, because we also have $e \models \boldsymbol{K}\sigma$, we must have $e = \{\}$.

Conversely, suppose that $e \models \sigma$ where $e = e_0$. Then clearly $e \models \boldsymbol{K}\sigma$, and since e_0 contains every w, $e \models \boldsymbol{O}\sigma$. Similarly, assume that $e \models \neg\sigma$ where $e = \{\}$. Then, for no w do we have $e, w \models \sigma$, and since $e \models \boldsymbol{K}\sigma$, we get that $e \models \boldsymbol{O}\sigma$. ∎

So if $\boldsymbol{O}\sigma$ satisfiable at all, it is only satisfied in trivial epistemic states like e_0 or $\{\}$, or both. For example, $\boldsymbol{O}\neg\boldsymbol{K}\psi$ is satisfied by both e_0 and $\{\}$. For a similar reason, we have the following:

Corollary 8.3.6: *Suppose that ψ is atomic. Then $\models \neg\boldsymbol{O}\boldsymbol{K}\psi$.*

Proof: Assume, to the contrary, that $e \models \boldsymbol{O}\boldsymbol{K}\psi$. Then, by the theorem, $e = e_0$ or $e = \{\}$. However, $e_0 \models \neg\boldsymbol{K}\psi$, and $\{\} \models \boldsymbol{K}\psi$, contradicting the theorem. ∎

Thus, just as there are sentences of the form $\boldsymbol{K}\alpha$ that are valid in $\mathcal{KL}$ and $\mathcal{OL}$, there are sentences $\neg\boldsymbol{O}\alpha$ that are valid in $\mathcal{OL}$. In other words, there are sentences that simply cannot be all that is known in any state, from e_0 to the inconsistent one.

The following properties provide us with criteria under which sentences can be conjoined or disjoined to what is only-known without actually changing the epistemic state.

Theorem 8.3.7: $\models (\boldsymbol{O}\alpha \wedge \boldsymbol{K}\beta \supset \boldsymbol{O}[\alpha \wedge \beta])$.

Proof: Suppose that $e \models \boldsymbol{O}\alpha \wedge \boldsymbol{K}\beta$. Then $e \models \boldsymbol{K}\alpha \wedge \boldsymbol{K}\beta$, and so $e \models \boldsymbol{K}[\alpha \wedge \beta]$. Now assume that $e, w \models [\alpha \wedge \beta]$. Then $e, w \models \alpha$, and so $w \in e$, since $e \models \boldsymbol{O}\alpha$. ∎

So in expressing all that is known, we can conjoin anything that happens to be known. The second property is:

Theorem 8.3.8: *For any subjective σ, $\models (\boldsymbol{O}\alpha \wedge \sigma \supset \boldsymbol{O}[\alpha \vee \neg\sigma])$.*

Proof: Suppose that $e \models \boldsymbol{O}\alpha \wedge \sigma$. Then $e \models \boldsymbol{K}\alpha$, and so $e \models \boldsymbol{K}[\alpha \vee \neg\sigma]$. Now assume that $e, w \models [\alpha \vee \neg\sigma]$. Then $e, w \models \alpha$, since $e \models \sigma$, and so $w \in e$. ∎

So in expressing all that is known, we can disjoin any false subjective sentence. This is just another way of saying that when it comes to only-knowing, the true subjective sentences add nothing, and the false ones take nothing away.

We will investigate many other properties of $\mathcal{OL}$ in more detail later, including an attempt at axiomatizing the logic in Chapter 9. At this point, however, we already know enough about the logic to apply it to the specification of **ASK** and **TELL** within $\mathcal{OL}$, a task we now turn to.

8.4 Characterizing ASK and TELL

Recall that **ASK** and **TELL** were specified originally in terms of what the epistemic state $\Re[\![\mathrm{KB}]\!]$ of a KB knows. With $\boldsymbol{O}$ this specification can be carried out entirely within $\mathcal{OL}$, that is, in terms of certain valid sentences.

Theorem 8.4.1: *Let KB be an objective sentence and α arbitrary. Then*

$$\mathbf{ASK}[\alpha, \Re[\![\mathrm{KB}]\!]] = \mathit{yes} \quad \text{iff} \quad \models (\boldsymbol{O}\mathrm{KB} \supset \boldsymbol{K}\alpha).$$

Proof: To prove the only-if direction, assume $e \models \boldsymbol{O}\mathrm{KB}$. Then $e = \Re[\![\mathrm{KB}]\!]$ because KB is objective. Since the answer is *yes*, we have $e \models \boldsymbol{K}\alpha$ by the definition of **ASK**.

Conversely, assume that $\models (\boldsymbol{O}\mathrm{KB} \supset \boldsymbol{K}\alpha)$. Clearly $\Re[\![\mathrm{KB}]\!] \models \boldsymbol{O}\mathrm{KB}$ and, therefore, $\Re[\![\mathrm{KB}]\!] \models \boldsymbol{K}\alpha$. So the answer is *yes*. ∎

Note that the theorem holds for any α, not just basic ones. Moreover, the use of $\boldsymbol{O}$ is essential for the theorem to go through.

The characterization of **TELL** turns out to be not quite as straightforward. One might expect that $\mathbf{TELL}[\alpha, \Re[\![\mathrm{KB}]\!]] = \Re[\![\mathrm{KB}^*]\!]$ iff $\models \boldsymbol{O}\mathrm{KB}^* \equiv \boldsymbol{O}[\mathrm{KB} \wedge \alpha]$.

While this is true for objective α, it does *not* hold if α is non-objective. For example, $\mathbf{TELL}[(\boldsymbol{K}p \vee p), e_0] = \Re[\![p]\!]$, but $\not\models \boldsymbol{O}p \equiv \boldsymbol{O}[\boldsymbol{K}p \vee p]$. To see why the equivalence fails, recall that **TELL** requires that any occurrence of $\boldsymbol{K}$ within the new sentence α be interpreted with respect to the *old* epistemic state $\Re[\![\mathrm{KB}]\!]$ (e_0 in the example). Occurrences of $\boldsymbol{K}$ within $\boldsymbol{O}[\mathrm{KB} \wedge \alpha]$, on the other hand, refer to the state(s) which only know $\mathrm{KB} \wedge \alpha$. As the example shows, these are in general different from $\Re[\![\mathrm{KB}]\!]$.

What does hold, on the other hand, is that adding an objective sentence ϕ to the KB as a result of $\mathbf{TELL}[\alpha, \Re[\![\mathrm{KB}]\!]]$ is correct just in case α is known to be equivalent to ϕ before **TELL** is performed. Formally:

Theorem 8.4.2: *Let KB and ϕ be objective, α arbitrary. Then*

$$\mathbf{TELL}[\alpha, \Re[\![\mathrm{KB}]\!]] = \Re[\![\mathrm{KB} \wedge \phi]\!] \quad \textit{iff} \quad \models (\boldsymbol{O}\mathrm{KB} \supset \boldsymbol{K}(\alpha \equiv \phi)).$$

Proof: Let $e = \Re[\![\mathrm{KB}]\!]$. Recall that $\mathbf{TELL}[\alpha, e] = e \cap \{w \mid e, w \models \alpha\}$. To show the if direction, assume that $(\boldsymbol{O}\mathrm{KB} \supset \boldsymbol{K}(\alpha \equiv \phi))$ is valid. Then clearly $e \models \boldsymbol{K}(\alpha \equiv \phi)$, that is, $\{w \in e \mid e, w \models \alpha\} = \{w \in e \mid w \models \phi\}$. Hence $\mathbf{TELL}[\alpha, e] = e \cap \{w \mid w \models \phi\} = \Re[\![\mathrm{KB} \wedge \phi]\!]$.

Conversely, assume that $\mathbf{TELL}[\alpha, e] = \Re[\![\mathrm{KB} \wedge \phi]\!]$. Then $e \cap \{w \mid e, w \models \alpha\} = e \cap \{w \mid w \models \phi\}$. Thus for any $w \in e$, $e, w \models \alpha$ iff $w \models \phi$, from which $e \models \boldsymbol{K}(\alpha \equiv \phi)$ follows immediately. ∎

Corollary 8.4.3: *For any objective KB and any basic α there is an objective ϕ such that* $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{K}(\alpha \equiv \phi))$.

Proof: The corollary follows immediately from this theorem and the Representation Theorem (Theorem 7.4.1). ∎

8.5 Determinate sentences

We have seen that objective sentences uniquely determine epistemic states, that is, for any ϕ, there is exactly one e such that $e \models \boldsymbol{O}\phi$. Let us call such sentences *determinate*. Notice that there is nothing in the definition that requires a determinate sentence to be objective and it seems worthwhile to look at the more general case. In this section, we will therefore consider arbitrary determinate sentences. The main result will be that the Representation Theorem, which we obtained in Chapter 7 for objective knowledge bases, carries over nicely to the case of arbitrary determinate knowledge bases. We begin by showing that determinate sentences indeed deserve their name, that is, they leave no doubt about what is and is not believed.

Theorem 8.5.1: *A sentence δ is determinate iff for every basic α, exactly one of* $(\boldsymbol{O}\delta \supset \boldsymbol{K}\alpha)$ *and* $(\boldsymbol{O}\delta \supset \neg\boldsymbol{K}\alpha)$ *is valid.*

Proof: First, suppose that δ is determinate, and that e is the unique maximal set of worlds satisfying $\boldsymbol{O}\delta$. Then, as in Corollary 8.3.2 either $(\boldsymbol{O}\delta \supset \boldsymbol{K}\alpha)$ or $(\boldsymbol{O}\delta \supset \neg\boldsymbol{K}\alpha)$ is valid according to whether $e \models \boldsymbol{K}\alpha$ or not.

Now suppose that exactly one of $(\boldsymbol{O}\delta \supset \boldsymbol{K}\alpha)$ or $(\boldsymbol{O}\delta \supset \neg\boldsymbol{K}\alpha)$ is valid for every basic α. $\boldsymbol{O}\delta$ must be satisfiable since otherwise it would imply every sentence. Moreover, for any e that satisfies it, we have that $e \models \boldsymbol{K}\alpha$ iff $(\boldsymbol{O}\delta \supset \boldsymbol{K}\alpha)$ is valid, because either $\boldsymbol{K}\alpha$ or $\neg\boldsymbol{K}\alpha$ is implied by $\boldsymbol{O}\delta$. Thus, if e and e' satisfy $\boldsymbol{O}\delta$, then $e \models \boldsymbol{K}\alpha$ iff $e' \models \boldsymbol{K}\alpha$ for every basic α. So e and e' are equivalent and, by Theorem 6.1.2, $e = e'$. Thus, δ is determinate. ∎

Thus, determinate sentences not only tell us exactly what is and what is not believed, they are also the only sentences to do so. As such, they can be used as *representations of knowledge*, since they implicitly specify a complete epistemic state.

To see that there are interesting determinate sentences beyond the objective ones, let γ be the closed world assumption from Chapter 5, saying that all instances of predicate P are known,

$$\forall x(P(x) \supset \boldsymbol{K}P(x)),$$

and consider $\mathrm{KB}_1 = \{P(^{\#}1),\ P(^{\#}2),\ \gamma\}$. As the following lemma shows, there is a unique epistemic state which only knows KB_1:

Lemma 8.5.2: *Let* $e = \Re[\![\forall x(P(x) \equiv [(x = {}^{\#}1) \vee (x = {}^{\#}2)])]\!]$. *Then for any* e^*, $e^* \models \boldsymbol{O}\mathrm{KB}_1$ *iff* $e = e^*$.

Proof: We begin by showing that $e \models \boldsymbol{O}\mathrm{KB}_1$. Clearly, $e \models \boldsymbol{K}\mathrm{KB}_1$. Now let $w \notin e$. There are two cases. If $w \not\models P(^{\#}1) \wedge P(^{\#}2)$, then $e, w \not\models \mathrm{KB}_1$. Otherwise, $w \models P(n)$ for some $n \notin \{^{\#}1, {}^{\#}2\}$. Then $e, w \models P(n) \wedge \neg \boldsymbol{K}P(n)$, from which $e, w \not\models \mathrm{KB}_1$ follows. Therefore, $e \models \boldsymbol{O}\mathrm{KB}_1$.

Now let e^* be any epistemic state such that $e^* \models \boldsymbol{O}\mathrm{KB}_1$. Consider any world $w \in e$, that is, $w \models P(n)$ iff $n \in \{^{\#}1, {}^{\#}2\}$. Since $e^* \models \boldsymbol{K}P(^{\#}1) \wedge \boldsymbol{K}P(^{\#}2)$, we obtain $e^*, w \models \mathrm{KB}_1$ and, hence, $w \in e^*$. Since this is true for all $w \in e$, we obtain $e \subseteq e^*$. Since e itself is maximal, no proper superset of e only-knows KB_1. Hence $e^* = e$. ∎

Note that KB_1 makes the closed world assumption just for P. In particular,

$\models \boldsymbol{O}\mathrm{KB}_1 \supset \boldsymbol{K}\neg P(^{\#}3)$, yet
$\models \boldsymbol{O}\mathrm{KB}_1 \supset \neg \boldsymbol{K}Q(^{\#}1)$
$\models \boldsymbol{O}\mathrm{KB}_1 \supset \neg \boldsymbol{K}\neg Q(^{\#}1)$
$\models \boldsymbol{O}\mathrm{KB}_1 \supset \neg \boldsymbol{K}\neg Q(^{\#}2)$
etc.

In general, we have that

$$\models \boldsymbol{O}\mathrm{KB}_1 \equiv \boldsymbol{O}\forall x(P(x) \equiv [(x = {}^{\#}1) \vee (x = {}^{\#}2)]).$$

We get the similar behaviour in the case where our knowledge about P is infinite. For example, let $\mathrm{KB}_2 = \{\forall x((x \neq 3) \supset P(x)), \gamma\}$. We leave it to the reader to prove that $\boldsymbol{O}\mathrm{KB}_2 \equiv \boldsymbol{O}[\forall x(P(x) \equiv (x \neq 3))]$ is valid.

If the knowledge base has incomplete information about P, applying the closed world assumption may not lead to a determinate knowledge base. For example, if $\mathrm{KB}_3 = \{P(^{\#}1) \vee P(^{\#}2), \gamma\}$, then there are two corresponding epistemic states, one where $^{\#}1$ is the only P and another one where $^{\#}2$ is the only P. Formally,

$$\boldsymbol{O}\mathrm{KB}_3 \equiv \boldsymbol{O}\forall x[P(x) \equiv (x = {}^{\#}1)] \vee \boldsymbol{O}\forall x[P(x) \equiv (x = {}^{\#}2)]$$

is valid. We will not prove this here, but in Chapter 10, we will have a lot more to say about such nondeterminate sentences. What we will show here is that the only way something like $(\phi \wedge \gamma)$ can lead to more than one epistemic state is when it is already known from ϕ that γ is false (as above).

First we need the following lemmas:

Lemma 8.5.3: *If* $e_1 \subseteq e_2$ *and* $e_2, w \models \gamma$, *then* $e_1, w \models \gamma$.

Proof: Suppose n is any name and $w \models P(n)$. Since $e_2, w \models \gamma$, we have that $e_2 \models \boldsymbol{K}P(n)$, from which it follows that $e_1 \models \boldsymbol{K}P(n)$, since $e_1 \subseteq e_2$. Consequently, for any n, $e_1, w \models (P(n) \supset \boldsymbol{K}P(n))$. ∎

Lemma 8.5.4: *Suppose* ϕ *is objective, and* $\Re[\![\phi]\!] \models \neg\boldsymbol{K}\neg\gamma$. *Further suppose that* e *is an epistemic state such that* $e \models \boldsymbol{O}(\phi \wedge \gamma)$. *Then for any* w, $e, w \models \gamma$ *iff* $\Re[\![\phi]\!], w \models \gamma$.

Proof: As with Lemma 5.6.1, we show that for any n, $e \models \boldsymbol{K}P(n)$ iff $\Re[\![\phi]\!] \models \boldsymbol{K}P(n)$. If $\Re[\![\phi]\!] \models \boldsymbol{K}P(n)$, then $e \models \boldsymbol{K}P(n)$, since $e \subseteq \Re[\![\phi]\!]$. Conversely, if $\Re[\![\phi]\!] \models \neg\boldsymbol{K}P(n)$, then since $\Re[\![\phi]\!] \models \neg\boldsymbol{K}\neg\gamma$, there is a w in $\Re[\![\phi]\!]$ such that $w \models \phi$, $\Re[\![\phi]\!], w \models \gamma$ and thus for which, $w \models \neg P(n)$. However, by the lemma above, we then have that $e, w \models (\phi \wedge \gamma)$, and so $w \in e$. Thus, there is a $w \in e$ such that $w \models \neg P(n)$, and so $e \models \neg\boldsymbol{K}P(n)$. ∎

Theorem 8.5.5: *Suppose* ϕ *is objective, and* $\Re[\![\phi]\!] \models \neg\boldsymbol{K}\neg\gamma$. *Then* $(\phi \wedge \gamma)$ *is determinate and* **TELL**$[\gamma, \Re[\![\phi]\!]]$ *is the unique epistemic state satisfying* $\boldsymbol{O}(\phi \wedge \gamma)$.

Proof: Let $e' = \textbf{TELL}[\gamma, \Re[\![\phi]\!]] = \{w \mid \Re[\![\phi]\!], w \models (\phi \wedge \gamma)\}$. We first show that $e' \models \boldsymbol{O}(\phi \wedge \gamma)$. By the definition of **TELL**, $e' \models \boldsymbol{K}\phi$, and by Theorem 5.6.2, $e' \models \boldsymbol{K}\gamma$, and so $e' \models \boldsymbol{K}(\phi \wedge \gamma)$. Now suppose that for some w, e', $w \models (\phi \wedge \gamma)$. By Lemma 5.6.1, $\Re[\![\phi]\!], w \models (\phi \wedge \gamma)$, and so $w \in e'$.

Next we need to show that if $e \models \boldsymbol{O}(\phi \wedge \gamma)$ then $e = e'$. So suppose that $e \models \boldsymbol{O}(\phi \wedge \gamma)$. Then for any w, we have that $w \in e$ iff $e, w \models (\phi \wedge \gamma)$ iff (by the lemma immediately above) $\Re[\![\phi]\!], w \models (\phi \wedge \gamma)$ iff (by Lemma 5.6.1) $e', w \models (\phi \wedge \gamma)$ iff $w \in e'$. ∎

So as long as γ is not already known to be false given ϕ, $(\phi \wedge \gamma)$ will be determinate. Moreover, from the point of view of **TELL**, we can see that if we start with an objective KB and assert that γ is true, we not only end up in a state where γ is known (as already established in Theorem 5.6.2), we also have that (KB $\wedge \gamma$) is *all* that is known.

The previous examples of determinate knowledge bases have in common that they can always be converted into equivalent objective knowledge bases. The main result of this section is that this is true in general, that is, it is always possible to represent determinate knowledge in objective terms. Although *believing* does not reduce to believing objective sentences (Theorem 4.6.2), *only believing* does, at least as far as determinate sentences are concerned.

Definition 8.5.6: Let ϕ be an objective formula, e an epistemic state, and δ a determinate sentence such that $e \models \boldsymbol{O}\delta$. Suppose that $n_1, \ldots, n_k$, are all the names in ϕ or in δ, and that n' is some name that does not appear in ϕ or in δ.
$\mathrm{RES}_{\boldsymbol{K}}[\![\phi, e]\!]$ is defined by:

1. If ϕ has no free variables, then $\mathrm{RES}_{\boldsymbol{K}}[\![\phi, e]\!]$ is
   $\forall z(z = z)$, if $e \models \boldsymbol{K}\phi$, and
   $\neg\forall z(z = z)$, otherwise.
2. If x is a free variable in ϕ, then $\mathrm{RES}_{\boldsymbol{K}}[\![\phi, e]\!]$ is
   $[((x = n_1) \wedge \mathrm{RES}_{\boldsymbol{K}}[\![\phi^x_{n_1}, e]\!]) \vee \ldots$
   $((x = n_k) \wedge \mathrm{RES}_{\boldsymbol{K}}[\![\phi^x_{n_k}, e]\!]) \vee$
   $((x \neq n_1) \wedge \ldots \wedge (x \neq n_k) \wedge \mathrm{RES}_{\boldsymbol{K}}[\![\phi^x_{n'}, e]\!]^{n'}_x)]$.

$\mathrm{RES}_{\boldsymbol{O}}[\![\phi, e]\!]$ is defined by:

1. If ϕ has no free variables, then $\mathrm{RES}_{\boldsymbol{O}}[\![\phi, e]\!]$ is
   $\forall z(z = z)$, if $e \models \boldsymbol{O}\phi$, and
   $\neg\forall z(z = z)$, otherwise.
2. If x is a free variable in ϕ, then $\mathrm{RES}_{\boldsymbol{O}}[\![\phi, e]\!]$ is
   $[((x = n_1) \wedge \mathrm{RES}_{\boldsymbol{O}}[\![\phi^x_{n_1}, e]\!]) \vee \ldots$
   $((x = n_k) \wedge \mathrm{RES}_{\boldsymbol{O}}[\![\phi^x_{n_k}, e]\!]) \vee$
   $((x \neq n_1) \wedge \ldots \wedge (x \neq n_k) \wedge \mathrm{RES}_{\boldsymbol{O}}[\![\phi^x_{n'}, e]\!]^{n'}_x)]$.

Note that the definition of $\mathrm{RES}_{\boldsymbol{K}}$ and $\mathrm{RES}_{\boldsymbol{O}}$ are exactly like the old RES except that the implication KB $\models \phi$ is replaced by $e \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{O}\phi$, respectively.

Given a determinate sentence δ, an epistemic state e with $e \models \boldsymbol{O}\delta$, and an arbitrary wff α of $\mathcal{OL}$, $\|\alpha\|_e$ is the objective wff defined by

1. $\|\alpha\|_e = \alpha$, when α is objective
2. $\|\neg\alpha\|_e = \neg\|\alpha\|_e$
3. $\|(\alpha \wedge \beta)\|_e = (\|\alpha\|_e \wedge \|\beta\|_e)$
4. $\|\forall x\alpha\|_e = \forall x\|\alpha\|_e$
5. $\|\boldsymbol{K}\alpha\|_e = \mathrm{RES}_{\boldsymbol{K}}[\![\|\alpha\|_e, e]\!]$
6. $\|\boldsymbol{O}\alpha\|_e = \mathrm{RES}_{\boldsymbol{O}}[\![\|\alpha\|_e, e]\!]$

Again, the definition of $\|\cdot\|$ is exactly like the old one except that now we also reduce formulas of the form $\boldsymbol{O}\alpha$.

Theorem 8.5.7: *For every determinate sentence δ there is an objective sentence ϕ such that $\models \boldsymbol{O}\delta \equiv \boldsymbol{O}\phi$.*

Proof: The proof is exactly analogous to the proof of the Representation Theorem. In particular, all the results of Section 7.2 and 7.3 carry over using the new definitions of RES and $\|\cdot\|$ in a straightforward way. Finally choose $\phi = \|\alpha\|_e$ where $e \models \boldsymbol{O}\delta$. ∎

We will see in Theorem 10.5.5 of Chapter 10 that this property does not hold in general for non-determinate sentences.

In this chapter we have seen how introducing the concept of only-knowing allows us to fully characterize **ASK** and **TELL** within the logic itself. The logic $\mathcal{OL}$ has many other uses, which we will explore in subsequent chapters.

***

This then ends the first part of the book, which can be thought of as providing the basic concepts of a logic of knowledge bases. In the remaining chapters, we will touch on various more specialized topics which all build on the foundations we have laid out so far.

8.6 Bibliographic notes

As we have seen, the key feature which distinguishes only-knowing from knowing (at least) is that both accessible and inaccessible worlds (e and its complement) are involved. This idea was independently developed by Humberstone [70] and later extended by Ben-David and Gafni [6]. Pratt-Hartmann [155] proposed what he calls *total knowledge*, which shares many of its properties with only-knowing. The semantics is based on sets of world states identical to ours except that beliefs are required to be true, that is, the actual world state is always considered possible. A sentence α is said to be total knowledge if α is known and every objective sentence which does not follow from knowing α is not known. As far as objective sentences are concerned, only-knowing and total knowledge basically coincide. In the general case, however, there are differences. Since these refer to properties of only-knowing treated in Chapters 10 and 11, we defer any further discussion of total knowledge to the end of Chapter 11.

Going back to the earlier work by Humberstone, Ben-David and Gafni, while they restrict themselves to the propositional case, they are in some sense more general than we are because they do not make the assumption of an underlying set of *all* worlds, that is, having a world for every interpretation of the atomic formulas. In fact, they allow general Kripke structures and consider modal logics other than K45. Allowing models with arbitrary sets of worlds, however, is problematic for only-knowing on intuitive grounds. For consider the case where we have just one world w where both p and q are true and w is the only accessible world. Then we have, for example, that both $\boldsymbol{O}p$ and $\boldsymbol{K}q$ hold.

This seems rather strange since only knowing p and, at the same time, knowing q seems incompatible with the intuitive reading of only-knowing.

To give only-knowing the right properties, then, it seems essential that the underlying models be large enough and contain worlds for every conceivable state of affairs. Note, however, that it is not at all obvious what constitutes a particular state of affairs. In our framework where we consider a single, fully introspective agent, it just so happens that a state of affairs can be identified with a world state. This is no longer the case when the agent is not fully introspective or when there are multiple agents. For example, consider the case of two agents A and B. From A's point of view, a state of affairs consists not just of facts about the world but also of B's beliefs about the world. This is because, as far as A is concerned, B's beliefs are just as objective for A as, say, the fact that Tina teaches Sue. Not surprisingly, modeling only-knowing for multiple agents is a complicated matter. Several approaches are discussed in [58, 87, 60]. In [18] Chen considers a specialized logic of only-knowing for two agents which allows the author to capture Gelfond's notion of *epistemic specifications* [51] within the framework of only-knowing.

In [61] Halpern and Moses define a concept of *minimal knowledge* in the propositional case which bears a striking resemblance to our notion of only-knowing. Roughly, given a sentence α, they define the corresponding epistemic state that only-knows α as the union of all sets of world states where α is known. α is called honest just in case α itself is known in this epistemic state. It is easily seen that every objective ϕ is honest. Indeed, for objective propositional formulas our notion of only-knowing coincides with that of Halpern and Moses. Also, just as there are sentences that cannot be only-known there are sentences that are dishonest, for example $\boldsymbol{K}p \vee \boldsymbol{K}q$. Despite those similarities there are differences as well. An obvious difference is that Halpern and Moses consider knowledge instead of belief, that is, $(\boldsymbol{K}\alpha \supset \alpha)$ comes out valid or, equivalently, the real world is always assumed to be among the accessible world states. A much more surprising difference has to do with complexity. In [35] it was discovered that reasoning about minimal knowledge is actually harder than reasoning about only-knowing (again, restricted to the propositional case). See also [104] for more discussion on the difference between only-knowing and minimal knowledge in both the single and multi-agent case.

In [85] only-knowing was extended to a notion of only-knowing about a subject matter, where a subject matter consisted of a set of atomic propositions. Given a KB consisting of $(\mathit{cows} \supset \mathit{mammals}) \wedge (\mathit{mammals} \supset \mathit{animals})$ one would then have that all the KB knows about cows is captured by $(\mathit{cows} \supset \mathit{mammals}) \wedge (\mathit{cows} \supset \mathit{animals})$, leaving out the more general information that mammals are animals. These ideas were later extended to the multi-agent case [88] and also served as the basis for a certain kind of logical relevance [90, 92]. The interested reader may wish to consult the first edition of this book [119] for a chapter dedicated to only-knowing about a subject matter.

Halpern and Moses' logic of minimal knowledge is only one example of a wide range of formalisms called nonmonotonic logics. Only-knowing has intimate connections to a number of these besides the one just mentioned. Since we will study one such connection in much more depth in chapters 10 and 11, we defer a discussion of the related literature to Section 11.6. Similarly, see Section 9.5 for literature on proof-theoretic and complexity issues regarding $\mathcal{OL}$.

8.7 Exercises

1. Generalize Theorem 8.3.1 to the case of non-finitely representable states as follows: Show that for any set of objective sentences Φ there is a unique epistemic state e such that for any objective ψ, $e \models \boldsymbol{K}\psi$ iff Φ logically implies ψ.
2. Consider the statement "Only-knowing is closed under logical consequence." State this precisely as a theorem, and prove that it is true.
3. Give an example of a subjective σ such that $\boldsymbol{O}\sigma$ is only satisfied by e_0, and another that is only satisfied by the inconsistent epistemic state.
4. Show that only-knowing is closed under introspection in the following sense: for any subjective σ,
   $$\models \boldsymbol{O}\alpha \wedge \sigma \supset \boldsymbol{O}(\alpha \wedge \sigma).$$
   Give an example of where this fails when $\boldsymbol{O}$ is replaced by $\boldsymbol{K}$.
5. Show that for no α is it the case that $\boldsymbol{O}\alpha$ is valid. Hint: consider finitely and infinitely representable states.
6. Show that for any falsifiable objective ϕ, $\models \neg\boldsymbol{O}[\phi \vee \boldsymbol{K}\phi]$.
7. Show that for any determinate KB and basic α, there is an objective ϕ with $\models (\boldsymbol{O}\mathrm{KB} \supset \boldsymbol{K}(\alpha \equiv \phi))$.
8. Let $\mathrm{KB} = \{\forall x((x \neq 3) \supset P(x)), \forall x(P(x) \supset \boldsymbol{K}P(x))\}$. Show the validity of $\boldsymbol{O}\mathrm{KB} \equiv \boldsymbol{O}[\forall x(P(x) \equiv (x \neq 3))]$.

Part II

9 On the Proof Theory of $\mathcal{OL}$

We already saw proof theoretic characterizations of $\mathcal{L}$ and $\mathcal{KL}$ and found them useful because they gave us an independent account of the valid sentences of the respective logics. In this chapter, we consider an axiom system for $\mathcal{OL}$. We demonstrate its usefulness by syntactically deriving some of the conclusions about the closed world assumption discussed in the previous chapter and prove its completeness for the propositional case. Unfortunately, it is not complete for the whole language, and we will discuss why this is so. In addition, the axioms are not even recursively enumerable, that is, we do not have a proof theory in the traditional sense. We will see that this feature is inescapable in $\mathcal{OL}$.

9.1 Knowing at least and at most

To better analyze only-knowing and for this chapter only, it is convenient to consider $\boldsymbol{O}$ not as a primitive notion but to define it in terms of $\boldsymbol{K}$ and a new operator $\boldsymbol{N}$ in the following way. One way to read $\boldsymbol{O}\alpha$ is to say that α is believed and nothing more, whereas $\boldsymbol{K}\alpha$ says that α is believed, and perhaps more. In other words, $\boldsymbol{K}\alpha$ means that α *at least* is believed to be true. A natural dual to this is to say that α *at most* is believed to be false, which we write $\boldsymbol{N}\alpha$. The idea behind introducing this operator is that $\boldsymbol{O}\alpha$ would then be *definable* as $(\boldsymbol{K}\alpha \wedge \boldsymbol{N}\neg\alpha)$, that is, at least α is believed and at most α is believed.[1] So, *exactly* α is believed. In other words, we are taking $\boldsymbol{K}$ to specify a lower bound on what is believed (since there may be other beliefs) and $\boldsymbol{N}$ to specify an upper bound on beliefs (since there may be fewer beliefs).[2] What is actually believed must lie between these two bounds.

These bounds can be seen most clearly when talking about objective sentences. Given an epistemic state as specified by a maximal set of world states e, to say that $\boldsymbol{K}\phi$ is true wrt e is to say that e is a subset of the states where ϕ is true. By symmetry then, $\boldsymbol{N}\neg\phi$ will be true when the set of states satisfying ϕ are a subset of e. The fact that e must contain all of these states means that nothing else can be believed that would eliminate any of them. This is the sense in which no more than ϕ is known. Finally, as before, $\boldsymbol{O}\phi$ is true iff both conditions hold and the two sets coincide.

This leads us to the precise definition of $\boldsymbol{N}\alpha$:

$6'.\ e, w \models \boldsymbol{N}\alpha$ iff for every w', if $e, w' \not\models \alpha$ then $w' \in e$.

from which the original constraint 6 on $\boldsymbol{O}\alpha$ follows trivially.

---

1 Although using negation with the $\boldsymbol{N}$ operator appears perhaps to be needlessly complex, it will indeed simplify matters later.

2 A slightly simplistic way of saying this is that $\boldsymbol{K}\alpha$ means that what is actually believed is of the form $(\alpha \wedge \beta)$, whereas $\boldsymbol{N}\alpha$ means what is actually believed is of the form $(\neg\alpha \vee \beta)$.

So what are the properties of believing at most that α is false? It is very easy to show that if α is valid, then $\boldsymbol{N}\alpha$ will be valid too, if $\boldsymbol{N}\alpha$ and $\boldsymbol{N}(\alpha \supset \beta)$ are both true, then so is $\boldsymbol{N}\beta$, and if some subjective σ is true, then so is $\boldsymbol{N}\sigma$. In other words, remarkably enough, $\boldsymbol{N}$ behaves like an ordinary belief operator: it is closed under logical implication and exhibits perfect introspection. This is most clearly seen by rephrasing very slightly the definition of $\boldsymbol{N}$ and comparing it to that of $\boldsymbol{K}$:

5. $e, w \models \boldsymbol{K}\alpha$ iff for every $w' \in e$, $e, w' \models \alpha$.

6′. $e, w \models \boldsymbol{N}\alpha$ iff for every $w' \notin e$, $e, w' \models \alpha$.[3]

Letting $\overline{e}$ stand for the set of states not in e, we have

6′. $e, w \models \boldsymbol{N}\alpha$ iff for every $w' \in \overline{e}$, $e, w' \models \alpha$.

So $\boldsymbol{N}$ is like a belief operator with one important difference: we use the complement of e. In possible-world terms, we range over the *inaccessible* possible world states. In other words, $\boldsymbol{K}$ and $\boldsymbol{N}$ give us two belief-like operators: one, with respect to e, and one with respect to $\overline{e}$.

In these terms, the relation between the two belief operators is that the accessible world states they range over must be *disjoint* (empty intersection) and *exhaustive* (universal union). As it turns out, only the exhaustiveness property is used in the axiomatization. In fact, we will show as part of the propositional completeness proof that there is an equivalent semantics where the set of world states considered for $\boldsymbol{K}$ and $\boldsymbol{N}$ may overlap.

In the following, a *subjective* sentence is understood as in $\mathcal{KL}$ except that any occurrence of $\boldsymbol{K}$ (but not necessarily all) can be replaced by $\boldsymbol{N}$. For example, the sentence $(\forall x \forall z(x = y) \supset \boldsymbol{KN}P(x, y))$ is considered subjective since its truth depends only on the epistemic state and its complement. The axioms for $\mathcal{OL}$ are then given as follows:[4]

1. $\mathbf{L}\alpha$, where α is an instance of an axiom of $\mathcal{L}$ (with the proviso on specialization).
2. $\mathbf{L}(\alpha \supset \beta) \supset \mathbf{L}\alpha \supset \mathbf{L}\beta$.
3. $\forall x \mathbf{L}\alpha \supset \mathbf{L}\forall x\alpha$.
4. $\sigma \supset \mathbf{L}\sigma$, where σ is subjective.
5. The $\boldsymbol{N}$ vs. $\boldsymbol{K}$ axiom:
   $(\boldsymbol{N}\phi \supset \neg\boldsymbol{K}\phi)$, where ϕ is any objective sentence such that $\not\models \phi$.
6. The definition of $\boldsymbol{O}$: $\boldsymbol{O}\alpha \equiv (\boldsymbol{K}\alpha \wedge \boldsymbol{N}\neg\alpha)$.

The first thing to notice is that $\boldsymbol{N}$, taken by itself, has precisely the same properties as $\boldsymbol{K}$, that is, $\boldsymbol{K}$ and $\boldsymbol{N}$ can both be thought of as ordinary belief operators except, of course, that there is a strong connection between the two. For one, Axiom 4 expresses that both $\boldsymbol{K}$

3 Note that phrased this way, α rather than $\neg\alpha$ is required to be true. This is why negation was used for $\boldsymbol{N}$.

4 In the following axioms, $\mathbf{L}$ is used as a modal operator standing for either $\boldsymbol{K}$ or $\boldsymbol{N}$. Multiple occurrences of $\mathbf{L}$ in the same axiom should be uniformly replaced by $\boldsymbol{K}$ or by $\boldsymbol{N}$.

and $\boldsymbol{N}$ are perfectly and mutually introspective. For example, $(\boldsymbol{K}\alpha \supset \boldsymbol{NK}\alpha)$ is an instance of 4. If we think of $\boldsymbol{K}$ and $\boldsymbol{N}$ as two agents, then this says that each agent has perfect knowledge of what the other knows. The other and more interesting connection between the two is, of course, Axiom 5, which is valid because $\boldsymbol{K}$ and $\boldsymbol{N}$ together range over all possible world states.

It is not hard to show that all these axioms are sound:

Theorem 9.1.1: *If a sentence of $\mathcal{OL}$ is derivable, then it is valid.*

Proof: The proof proceeds by a standard induction on the length of a derivation, just like in the case $\mathcal{KL}$. Here we only show the base case for the $\boldsymbol{N}$ vs. $\boldsymbol{K}$ axiom. Thus let ϕ be an objective sentence such that $\not\models \phi$ and e a maximal set of world states such that $e \models \boldsymbol{N}\phi$. Then for all world states not in e, $w \models \phi$. Since $\not\models \phi$, there must be a world state w' such that $w' \models \neg\phi$. Moreover, w' must be in e, from which $e \models \neg\boldsymbol{K}\phi$ follows. ∎

9.2 Some example derivations

Before turning to the issue of completeness, let us go through several derivations in order to get a better feel for the axioms. The examples are taken from Section 8.5 involving the closed world assumption. Syntactic derivations of the closed world assumption are particularly instructive, since they exhibit very nicely the use of Axiom 5 and, more generally, the power of the proof theory.

In the derivations to follow, we will use a natural-deduction-style argument with three justifications: (1) the definition of $\boldsymbol{O}$ in terms of $\boldsymbol{N}$ and $\boldsymbol{K}$, (2) the axiom relating $\boldsymbol{N}$ to $\boldsymbol{K}$ for objective sentences, and (3) $\mathcal{KL}$, which we normally do not analyze further. When writing $\mathcal{KL}$ as a justification we really mean Axioms 2–4, which include the axioms of $\mathcal{KL}$ with $\boldsymbol{K}$ replaced by $\boldsymbol{N}$.

Recall from Section 8.5 that we were able to capture the closed world assumption for a unary predicate P by adding the sentence

$$\gamma = \forall x(P(x) \supset \boldsymbol{K}P(x))$$

to the knowledge base. We saw that only knowing that P holds of $^{\#}1$ and $^{\#}2$ together with γ logically implies that $P(^{\#}3)$ is known to be false. With the axioms above we can now give a syntactic derivation.

Example 9.2.1: If $\mathrm{KB} = \{P(^{\#}1) \wedge P(^{\#}2)\}$ then $(\boldsymbol{O}(\mathrm{KB} \wedge \gamma) \supset \boldsymbol{K}\neg P(^{\#}3))$ is a theorem.

Proof:

1. $\boldsymbol{O}(\mathrm{KB} \wedge \gamma)$	Assumption.
2. $\boldsymbol{K}(\mathrm{KB} \wedge \gamma)$	1;defn. of $\boldsymbol{O}$.
3. $(\boldsymbol{K}\neg\boldsymbol{K}P(^{\#}3) \supset \boldsymbol{K}\neg P(^{\#}3))$	2;$\mathcal{KL}$.
4. $(\neg\boldsymbol{K}P(^{\#}3) \supset \boldsymbol{K}\neg P(^{\#}3))$	3;$\mathcal{KL}$.
5. $\boldsymbol{N}\neg(\mathrm{KB} \wedge \gamma)$	1;defn. of $\boldsymbol{O}$.
6. $\boldsymbol{N}(\mathrm{KB} \supset \exists x\, P(x))$	5;$\mathcal{KL}$.
7. $\neg\boldsymbol{K}(\mathrm{KB} \supset \exists x\, P(x))$	6;$\boldsymbol{N}$ vs. $\boldsymbol{K}$.
8. $\neg\boldsymbol{K}P(^{\#}3)$	7;$\mathcal{KL}$.
9. $\boldsymbol{K}\neg P(^{\#}3)$	4,8;$\mathcal{KL}$.

Most of the uses of $\mathcal{KL}$ here are direct: lines 6 and 8 involve propositional reasoning within a modal operator; line 9 is the result of ordinary propositional logic; line 3 requires distributing the $\boldsymbol{K}$ over a conjunction, then over an implication; and line 4 uses the $\mathcal{KL}$ axiom $(\neg\boldsymbol{K}P(^{\#}3) \supset \boldsymbol{K}\neg\boldsymbol{K}P(^{\#}3))$ to obtain the final formula. Note that line 7 is the only place where we need to make use of the special connection between $\boldsymbol{K}$ and $\boldsymbol{N}$ (Axiom 5). The derivation is valid because $(\mathrm{KB} \supset \exists x\, P(x))$ clearly is falsifiable. ∎

Example 9.2.2: Let

$$\mathrm{KB} = \{(P(^{\#}1) \vee P(^{\#}2))\}$$

The theorem to derive is $\boldsymbol{O}[\mathrm{KB} \wedge \gamma] \equiv (\boldsymbol{O}[\mathrm{KB} \wedge \phi_1)] \vee \boldsymbol{O}[\mathrm{KB} \wedge \phi_2])$,
where $\phi_1 = \forall x(P(x) \equiv x = {}^{\#}1)$ and $\phi_2 = \forall x(P(x) \equiv x = {}^{\#}2)$.

Proof: In the following syntactic derivation we will occasionally write a line σ and then follow it by $\boldsymbol{K}\sigma$ or $\boldsymbol{N}\sigma$. This is *not* using the rule: from α infer $\boldsymbol{K}\alpha$ which, in general, is only sound when α is valid. Rather it is an application of *modus ponens*, together with the axiom $(\sigma \supset \boldsymbol{K}\sigma)$, which holds whenever σ is subjective. Also, for clarity, we will expand some of the steps that depend only on properties of $\mathcal{KL}$.

The first part of the proof (only-if direction) has two stages: we first establish that $\boldsymbol{K}P(^{\#}1) \vee \boldsymbol{K}P(^{\#}2)$ is provable given the assumption $\boldsymbol{O}[\mathrm{KB} \wedge \gamma]$; then we show that each of the disjuncts of the desired conclusion is derivable from either $\boldsymbol{K}P(^{\#}1)$ or $\boldsymbol{K}P(^{\#}2)$.

1. $\boldsymbol{O}[\mathrm{KB} \wedge \gamma]$	Assumption.
2. $\boldsymbol{N}[\mathrm{KB} \supset \exists x(P(x) \wedge \neg\boldsymbol{K}P(x))]$	1;defn. of $\boldsymbol{O}$.
3. $\boldsymbol{K}[\mathrm{KB} \wedge \forall x(P(x) \supset \boldsymbol{K}P(x))]$	1;defn. of $\boldsymbol{O}$.
4. $\boldsymbol{K}(P(^{\#}1) \vee P(^{\#}2))$	3;$\mathcal{KL}$.
5. $\boldsymbol{K}(\boldsymbol{K}P(^{\#}1) \vee \boldsymbol{K}P(^{\#}2))$	3,4;$\mathcal{KL}$.

6. $\boldsymbol{K}P(^{\#}1) \vee \boldsymbol{K}P(^{\#}2)$ 5;$\mathcal{KL}$.

Now what we will do is show that

$$\boldsymbol{K}P(^{\#}1) \supset \boldsymbol{O}[\mathrm{KB} \wedge \forall x(P(x) \equiv (x = {}^{\#}1))]$$

is a theorem, and so, by an analogous derivation, we have that

$$\boldsymbol{K}P(^{\#}2) \supset \boldsymbol{O}[\mathrm{KB} \wedge \forall x(P(x) \equiv (x = {}^{\#}2))]$$

is a theorem, in which case, the required disjunction follows immediately from Line 6.

7. $\boldsymbol{K}P(^{\#}1)$ Assumption.
8. $(\boldsymbol{K}\forall x(x = {}^{\#}1 \supset P(x)))$ 7;$\mathcal{KL}$.
9. $\boldsymbol{N}\boldsymbol{K}P(^{\#}1)$ 7;$\mathcal{KL}$.
10. $\boldsymbol{N}[\mathrm{KB} \supset \exists x(P(x) \wedge (x \neq {}^{\#}1))]$ 2,9;$\mathcal{KL}$.
11. $\neg\boldsymbol{K}[\mathrm{KB} \supset \exists x(P(x) \wedge (x \neq {}^{\#}1))]$ 10;$\boldsymbol{N}$ vs. $\boldsymbol{K}$.
12. $(\forall x(x \neq {}^{\#}1) \supset \neg\boldsymbol{K}P(x))$ 11;$\mathcal{KL}$.
13. $\boldsymbol{K}[\forall x(x \neq {}^{\#}1) \supset \neg\boldsymbol{K}P(x)]$ 12;$\mathcal{KL}$.
14. $\boldsymbol{K}[\forall x(x \neq {}^{\#}1) \supset \neg P(x)]$ 3,13;$\mathcal{KL}$.
15. $\boldsymbol{K}[\mathrm{KB} \wedge \forall x(P(x) \equiv (x = {}^{\#}1))]$ 8,14;$\mathcal{KL}$.
16. $\boldsymbol{N}[\mathrm{KB} \supset \exists x(P(x) \wedge (x \neq {}^{\#}1))]$ 2,7;$\mathcal{KL}$.
17. $\boldsymbol{N}[\mathrm{KB} \supset \neg\forall x(P(x) \equiv (x = {}^{\#}1))]$ 16;$\mathcal{KL}$.
18. $\boldsymbol{O}[\mathrm{KB} \wedge \forall x(P(x) \equiv (x = {}^{\#}1))]$ 15,17;defn. of $\boldsymbol{O}$.

This completes the first part of the proof. Note that except for line 11 all the reasoning involved requires only the axioms of $\mathcal{KL}$ (for both $\boldsymbol{K}$ and $\boldsymbol{N}$). Line 11 requires Axiom 5, which is applicable because $(\mathrm{KB} \supset \exists x(P(x) \wedge (x \neq {}^{\#}1)))$ is not valid.

We now proceed to the if direction of the proof. What we will do is show that

$$\boldsymbol{O}[\mathrm{KB} \wedge \forall x(P(x) \equiv (x = {}^{\#}1))] \supset \boldsymbol{O}[\mathrm{KB} \wedge \gamma]$$

is a theorem. Because this proof applies equally well to $^{\#}2$, the disjunction of the two possibilities gives us the desired conclusion.

19. $\boldsymbol{O}[\mathrm{KB} \wedge \forall x(P(x) \equiv (x = {}^{\#}1))]$ Assumption.
20. $\boldsymbol{K}[\mathrm{KB} \wedge \forall x(P(x) \equiv (x = {}^{\#}1))]$ 19;defn. of $\boldsymbol{O}$.
21. $\boldsymbol{K}P(^{\#}1)$ 20;$\mathcal{KL}$.
22. $(\boldsymbol{K}\forall x(P(x) \supset \boldsymbol{K}P(x)))$ 20,21;$\mathcal{KL}$.
23. $\boldsymbol{K}[\mathrm{KB} \wedge \gamma]$ 20,22;$\mathcal{KL}$.
24. $\boldsymbol{N}[\mathrm{KB} \supset \neg\forall x(P(x) \equiv (x = {}^{\#}1))]$ 19;defn. of $\boldsymbol{O}$.
25. $\boldsymbol{N}[\mathrm{KB} \supset \exists x P(x)]$ $\mathcal{KL}$.
26. $\boldsymbol{N}[\mathrm{KB} \supset \exists x(x \neq {}^{\#}1 \wedge P(x))]$ 24,25;$\mathcal{KL}$.

27. $\neg \boldsymbol{K}[\text{KB} \supset \exists x(x \neq {}^{\#}1 \wedge P(x))]$	26;$\boldsymbol{N}$ vs. $\boldsymbol{K}$.
28. $\neg \boldsymbol{K}[\exists x(x \neq {}^{\#}1 \wedge P(x))]$	27;$\mathcal{KL}$.
29. $(\forall x(x \neq {}^{\#}1 \supset \neg \boldsymbol{K} P(x)))$	28;$\mathcal{KL}$.
30. $\boldsymbol{N}\forall x(x \neq {}^{\#}1 \supset \neg \boldsymbol{K} P(x))$	29;$\mathcal{KL}$.
31. $\boldsymbol{N}[\text{KB} \supset \exists x(P(x) \wedge \neg \boldsymbol{K} P(x))]$	26,30;$\mathcal{KL}$.
32. $\boldsymbol{O}[\text{KB} \wedge \gamma]$	23,31;defn. of $\boldsymbol{O}$.

Note that there is again only one place in the proof where we need Axiom 5 (line 27), and it is exactly the same as in the first part. This completes the proof. ∎

9.3 Propositional completeness

We now turn to the issue of completeness of the axiom system in the propositional case. The proof uses the standard technique of constructing satisfying models for maximally consistent sets. As in the case of $\mathcal{KL}$, not every maximally consistent set is satisfiable. In $\mathcal{KL}$ this was due mainly to technical reasons, having to do with quantification and standard names. Interestingly, the reasons here are not only entirely different, which is not surprising, but they also lead to a deeper understanding of the underlying semantics. As we already mentioned, it is sufficient for Axiom 5 to be sound if $\boldsymbol{K}$ and $\boldsymbol{N}$ cover all world states. Whether or not the two sets are disjoint or not seems to be neither sufficient nor necessary. In fact, it turns out that there are maximally consistent sets which are only satisfiable if we allow the set of world states considered by $\boldsymbol{K}$ to overlap with the set of world states considered by $\boldsymbol{N}$. In a moment, we will consider just such an "overlapping" semantics. Not only are all of the axioms of $\mathcal{OL}$ (restricted to the propositional case) sound, but the valid sentences of the overlapping semantics are precisely the same as in our original semantics. What this tells us is that, while the disjointness property maybe more intuitively appealing, it cannot be captured axiomatically. Indeed, insisting on disjointness needlessly complicates the completeness proof since we would have to show that we can restrict ourselves to a subset of the maximally consistent sets. While this can be done, we will instead prove completeness for the equivalent overlapping semantics, which is straightforward given the known techniques for modal logics. Finally, the overlapping semantics has another revealing feature in that it does not require maximal sets of world states. So, as a corollary, we get that our original semantics yields the same valid sentences if we allow arbitrary epistemic states, not just maximal ones.

We begin with the definition of the overlapping semantics for propositional $\mathcal{OL}$. Let e and e' be two sets of world states. If $e \cup e' = e_0$, then we call (e, e') an *exhaustive pair*. Define a new satisfaction relation $\models^x$ that is exactly like $\mathcal{OL}$'s except for $\boldsymbol{K}$- and $\boldsymbol{N}$-formulas.

Let (e_K, e_N) be an exhaustive pair. Then

1. $e_K, e_N, w \models^x \boldsymbol{K}\alpha$ if $e_K, e_N, w' \models^x \alpha$ for all $w' \in e_K$
2. $e_K, e_N, w \models^x \boldsymbol{N}\alpha$ if $e_K, e_N, w' \models^x \alpha$ for all $w' \in e_N$.

Note that $\boldsymbol{K}$ and $\boldsymbol{N}$ are now treated in a completely symmetric way.

A sentence α is called *x-valid* ($\models^x \alpha$) iff $e_K, e_N, w \models^x \alpha$ for all world states w and all exhaustive pairs (e_K, e_N). α is called *x-satisfiable* iff $\neg\alpha$ is not x-valid.

Theorem 9.3.1: *The axioms are sound with respect to* $\models^x$.

The proof is straightforward and omitted. Note that for Axiom 5 to be sound it suffices that e_K and e_N together cover all world states. In particular, it does not matter whether or not the two sets overlap.

It is clear that the notions of satisfiability and validity coincide for both semantics when restricted to objective formulas. To prove that this is true in general, we need the following lemmas.

Lemma 9.3.2: *Let ϕ and ψ be objective formulas such that $\phi \wedge \neg\psi$ is satisfiable. Then $(\boldsymbol{K}\neg\phi \supset \neg\boldsymbol{N}\psi)$ and $(\boldsymbol{N}\neg\phi \supset \neg\boldsymbol{K}\psi)$ are both valid and x-valid.*

Proof: The proof relies only on the fact that the world states considered for $\boldsymbol{K}$ together with those for $\boldsymbol{N}$ cover all world states, which holds in either semantics. Here we only consider the nonstandard case.

Let $e_K, e_N, w \models^x \boldsymbol{K}\neg\phi$. Then e_N contains every world state that satisfies ϕ. Hence, by assumption, there is a $w' \in e_N$ such that $w' \models^x \phi \wedge \neg\psi$, that is, $e_K, e_N, w \models^x \neg\boldsymbol{N}\psi$.

$(\boldsymbol{N}\neg\phi \supset \neg\boldsymbol{K}\psi)$ is handled in a completely symmetric way. ∎

Next we show that every propositional sentence is provably equivalent to one without nested modal operators. Let us write $\mathbf{L}$ to stand for either $\boldsymbol{K}$ or $\boldsymbol{N}$. As before, we write $\vdash \alpha$ for α is provable from the axioms.

Lemma 9.3.3: $\vdash \mathbf{L}(\phi \vee \sigma) \equiv (\mathbf{L}\phi \vee \sigma)$.

Proof: Since $\vdash (\sigma \supset \mathbf{L}(\phi \vee \sigma))$ because σ is subjective, and $\vdash (\mathbf{L}\phi \supset \mathbf{L}(\phi \vee \sigma))$, then clearly $\vdash ((\mathbf{L}\phi \vee \sigma) \supset \mathbf{L}(\phi \vee \sigma))$. Conversely, since $\vdash (\mathbf{L}(\alpha \supset \beta) \supset (\mathbf{L}\alpha \supset \mathbf{L}\beta))$ in general, we have that $\vdash (\mathbf{L}(\phi \vee \sigma) \supset (\mathbf{L}\phi \vee \neg\mathbf{L}\neg\sigma))$. But since σ is subjective, $\vdash (\neg\mathbf{L}\neg\sigma \supset \sigma)$. Thus, $\vdash (\mathbf{L}(\phi \vee \sigma) \supset (\mathbf{L}\phi \vee \sigma))$. ∎

Lemma 9.3.4: *Every sentence is provably equivalent to one where* $\boldsymbol{K}$ *or* $\boldsymbol{N}$ *operators apply only to objective sentences.*

Proof: Consider a subformula $\mathbf{L}\alpha$. Using the rules of standard propositional logic, put α into conjunctive normal form so that $\vdash (\alpha \equiv \bigwedge(\phi_i \vee \sigma_i))$, where we have separated the subjective and objective parts. By induction, assume that each σ_i is in the correct form. Then,[5] $\vdash (\mathbf{L}\alpha \equiv \bigwedge \mathbf{L}(\phi_i \vee \sigma_i))$ and so, by the above lemma, $\vdash (\mathbf{L}\alpha \equiv \bigwedge(\mathbf{L}\phi_i \vee \sigma_i))$. One level of nesting of $\mathbf{L}$ has been eliminated. By applying this to all subformulas, the correct sentence will be obtained. ∎

Theorem 9.3.5: *For all propositional sentences* α, α *is x-satisfiable iff* α *is satisfiable.*

Proof: The only-if direction is straightforward. If $e, w \models \alpha$ then simply let $e_{\mathrm{K}} = e$ and let e_{N} be the complement of e.

To prove the if direction, let $e_{\mathrm{K}}, e_{\mathrm{N}}, w \models^x \alpha$. We need to find a maximal set of world states e such that $e, w \models \alpha$. By Theorem 9.3.1 and Lemma 9.3.4, we can assume, without loss of generality, that $\alpha = \bigvee \alpha_m$, where each α_m has the form

$$\alpha_m = \phi \wedge \bigwedge \boldsymbol{K} r_i \wedge \bigwedge \neg \boldsymbol{K} s_j \wedge \bigwedge \boldsymbol{N} t_k \wedge \bigwedge \neg \boldsymbol{N} u_l$$

for objective ϕ, r_i, s_j, t_k, and u_l. Then $e_{\mathrm{K}}, e_{\mathrm{N}}, w \models^x \alpha_m$ for some m. It suffices to find a maximal set e such that $e, w \models \alpha_m$.

Let $\alpha^* = (\bigwedge t_k \supset (p \wedge \bigwedge r_i))$ for some atom p not occurring in α and let $e = \{w \mid w \models \alpha^*\}$. e is obviously maximal and $e \models O\alpha^*$ or, equivalently, $e \models \boldsymbol{K}\alpha^* \wedge \boldsymbol{N}\neg\alpha^*$. To show that $e, w \models \alpha_m$ we need to prove that each of the conjuncts is satisfied.

$e, w \models \phi$: Follows immediately because ϕ is objective and $e_{\mathrm{K}}, e_{\mathrm{N}}, w \models^x \alpha_m$.

$e, w \models \boldsymbol{K} r_i$: Let $w' \in e$. Then $w' \models \neg \bigwedge t_k \vee (p \wedge r_i)$. If $w' \models \bigwedge t_k$, then $w' \models r_i$. Otherwise, $w' \models \neg \bigwedge t_k$. Since $\boldsymbol{N} \bigwedge t_k \wedge \boldsymbol{K} r_i$ is x-satisfiable, by Lemma 9.3.2, $\models (\neg \bigwedge t_k \supset r_i)$. Hence $w' \models r_i$.

$e, w \models \neg \boldsymbol{K} s_j$: Since $(e_{\mathrm{K}}, e_{\mathrm{N}}, w) \models^x \boldsymbol{K} \bigwedge r_i \wedge \neg \boldsymbol{K} s_j$, $\not\models (\bigwedge r_i \supset s_j)$. Since p does not occur in any of the r_i and s_j, $\not\models ((p \bigwedge r_i) \supset s_j)$. Thus, by Lemma 9.3.2, $\models (\boldsymbol{N}\neg(p \bigwedge r_i) \supset \neg \boldsymbol{K} s_j)$, from which $e \models \neg \boldsymbol{K} s_j$ follows.

$e, w \models \boldsymbol{N} t_k$: Follows immediately from $e \models \boldsymbol{N}\neg\alpha^*$.

$e, w \models \neg \boldsymbol{N} u_l$: Since $e_{\mathrm{K}}, e_{\mathrm{N}}, w \models^x \boldsymbol{N} \bigwedge t_k \wedge \neg \boldsymbol{N} u_l$, $\not\models (\bigwedge t_k \supset u_l)$ and, therefore, $\not\models ((\bigwedge t_k \wedge \neg p) \supset u_l)$, which implies $(\not\models \neg\alpha^* \supset u_l)$. Thus, by Lemma 9.3.2, $\models (\boldsymbol{K}\alpha^* \supset \neg \boldsymbol{N} u_l)$, from which $e \models \neg \boldsymbol{N} u_l$ follows. ∎

5 We repeatedly use the fact here and later that $\vdash \bigwedge \mathbf{L}\alpha_i \equiv \mathbf{L} \bigwedge \alpha_i$.

Since a sentence is valid (x-valid) iff its negation is not satisfiable (x-satisfiable) we immediately get:

Corollary 9.3.6: $\models \alpha$ *iff* $\models^x \alpha$.

An interesting consequence of the theorem is that whether or not we use maximal sets of world states has no effect on the valid sentences, at least as far as the propositional subset of $\mathcal{OL}$ is concerned.

Corollary 9.3.7: $\models \alpha$ *iff* $e, w \models \alpha$ *for all world states* w *and epistemic states* e *(including nonmaximal* e*).*

Proof: The if direction holds immediately since maximal sets of world states are special epistemic states.

For the converse, assume $\models \alpha$ and let w be a world state and e an arbitrary epistemic state. Let $\overline{e}$ denote the complement of e. By Theorem9.3.6, $\models^x \alpha$ and hence $e, \overline{e}, w \models^x \alpha$, from which $e, w \models \alpha$ follows immediately. ∎

Theorem 9.3.8: *For all propositional sentences* α, *if* $\models^x \alpha$ *then* $\vdash \alpha$.

Proof: The approach is very similar to the one taken in the completeness proof of $\mathcal{KL}$ (Theorem 4.5.1). Recall that to establish completeness it suffices to show that every consistent sentence is satisfiable, which in turn is established by showing that any maximally consistent set that contains α is satisfiable. Of course, (maximal) consistency now refers to the axioms of $\mathcal{OL}$.

Let $\mathcal{C}_0$ be the set of all maximally consistent sets of propositional sentences of $\mathcal{OL}$. For $\Gamma \in \mathcal{C}_0$, define $\Gamma/K = \{\alpha \mid \boldsymbol{K}\alpha \in \Gamma\}$ and $\Gamma/N = \{\alpha \mid \boldsymbol{N}\alpha \in \Gamma\}$. We then define

- $\mathcal{C}_K^\Gamma = \{\Gamma' \in \mathcal{C}_0 \mid \Gamma/K \subseteq \Gamma'\}$,
- $\mathcal{C}_N^\Gamma = \{\Gamma' \in \mathcal{C}_0 \mid \Gamma/N \subseteq \Gamma'\}$.

If we view maximally consistent sets as world states, then $\mathcal{C}_K^\Gamma$ and $\mathcal{C}_N^\Gamma$ represent the world states accessible from Γ for $\boldsymbol{K}$ and $\boldsymbol{N}$, respectively. The following lemma reflects the fact that $\boldsymbol{K}$ and $\boldsymbol{N}$ are both fully and mutually introspective (Axiom 4).

Lemma 9.3.9: *If* $\Gamma' \in \mathcal{C}_K^\Gamma \cup \mathcal{C}_N^\Gamma$, *then* $\mathcal{C}_K^{\Gamma'} = \mathcal{C}_K^\Gamma$ *and* $\mathcal{C}_N^{\Gamma'} = \mathcal{C}_N^\Gamma$.

Proof: We prove the lemma for $\Gamma' \in \mathcal{C}_K^\Gamma$. The case $\Gamma' \in \mathcal{C}_N^\Gamma$ is completely symmetric. To show that $\mathcal{C}_K^\Gamma = \mathcal{C}_K^{\Gamma'}$, it clearly suffices to show that $\Gamma/K = \Gamma'/K$. Let $\alpha \in \Gamma/K$. Then

$\boldsymbol{K}\alpha \in \Gamma$ and also $\boldsymbol{K}\boldsymbol{K}\alpha \in \Gamma$ by Axiom 4. Thus $\boldsymbol{K}\alpha \in \Gamma'$ (since $\Gamma' \in \mathcal{C}_K^\Gamma$ implies that $\Gamma/\boldsymbol{K} \subseteq \Gamma'$) and, hence, $\alpha \in \Gamma'/\boldsymbol{K}$.

For the converse, let $\alpha \in \Gamma'/K$. Thus, $\boldsymbol{K}\alpha \in \Gamma'$. Assume that $\alpha \notin \Gamma/K$. Then $\neg\boldsymbol{K}\alpha \in \Gamma$ (since Γ is a maximally consistent set) and, therefore, $\boldsymbol{K}\neg\boldsymbol{K}\alpha \in \Gamma$, from which $\neg\boldsymbol{K}\alpha \in \Gamma'$ follows, a contradiction.

The proof that $\mathcal{C}_N^{\Gamma'} = \mathcal{C}_N^\Gamma$ proceeds the same way, that is, we show that $\Gamma/N = \Gamma'/N$. Let $\alpha \in \Gamma/N$. Then $\boldsymbol{N}\alpha \in \Gamma$ and also $\boldsymbol{K}\boldsymbol{N}\alpha \in \Gamma$ by Axiom 4. Hence $\boldsymbol{N}\alpha \in \Gamma'$, so $\alpha \in \Gamma'/N$.

For the converse, let $\alpha \in \Gamma'/N$. Thus, $\boldsymbol{N}\alpha \in \Gamma'$. Assume that $\alpha \notin \Gamma/N$. Then $\neg\boldsymbol{N}\alpha \in \Gamma$ and also $\boldsymbol{K}\neg\boldsymbol{N}\alpha \in \Gamma$, from which $\neg\boldsymbol{N}\alpha \in \Gamma'$ follows, a contradiction. ∎

As in the completeness proof for $\mathcal{KL}$, each maximally consistent set Γ is mapped into the world state w_Γ with $w[p] = 1$ iff $p \in \Gamma$ for all atomic propositions p.

For any $\Gamma \in \mathcal{C}_0$, let $e_K^\Gamma = \{w_{\Gamma'} \mid \Gamma' \in \mathcal{C}_K^\Gamma\}$ and $e_N^\Gamma = \{w_{\Gamma'} \mid \Gamma' \in \mathcal{C}_N^\Gamma\}$.

Lemma 9.3.10:

(a) (e_K^Γ, e_N^Γ) *is an exhaustive pair.*

(b) For all α, *we have* $\alpha \in \Gamma$ *iff* $e_K^\Gamma, e_N^\Gamma, w_\Gamma \models^x \alpha$.

Proof: For part (a), to show that (e_K^Γ, e_N^Γ) is an exhaustive pair, we must show that $e_K^\Gamma \cup e_N^\Gamma$ consists of all world states. By way of contradiction, suppose there is a world state w not in $e_K^\Gamma \cup e_N^\Gamma$. Let $F_w = \{p \mid p \text{ is an atom and } w \models p\} \cup \{\neg p \mid p \text{ is an atom and } w \models \neg p\}$. $F_w \cup \Gamma/K$ cannot be consistent, for otherwise there would be some $\Gamma' \in \mathcal{C}_K^\Gamma$ that contains F_w, which would mean that $w \in e_K^\Gamma$. Similarly $F_w \cup \Gamma/N$ cannot be consistent. Thus, there must be formulas $\phi_1, \phi_2, \phi_3, \phi_4$ such that ϕ_1 and ϕ_2 are both conjunctions of a finite number of formulas in F_w, ϕ_3 is the conjunction of a finite number of formulas in Γ/L, and ϕ_4 is the conjunction of a finite number of formulas in Γ/N, and both $\phi_1 \wedge \phi_3$ and $\phi_2 \wedge \phi_4$ are inconsistent. Thus, we have $\vdash (\phi_3 \supset \neg\phi_1)$ and $\vdash (\phi_4 \supset \neg\phi_2)$. Using standard modal reasoning, we have $\vdash (\boldsymbol{K}\phi_3 \supset \boldsymbol{K}\neg\phi_1)$ and $\vdash (\boldsymbol{N}\phi_4 \supset \boldsymbol{N}\neg\phi_2)$. Since $\boldsymbol{K}\psi \in \Gamma$ for each conjunct ψ of ϕ_3, standard modal reasoning shows that $\boldsymbol{K}\phi_3 \in \Gamma$. Similarly, we have $\boldsymbol{N}\phi_4 \in \Gamma$. Since Γ is a maximally consistent set, both $\boldsymbol{K}\neg\phi_1$ and $\boldsymbol{N}\neg\phi_2$ are in Γ. Since $\vdash (\boldsymbol{K}\neg\phi_1 \supset \boldsymbol{K}(\neg\phi_1 \vee \neg\phi_2))$ and $\vdash (\boldsymbol{N}\neg\phi_2 \supset \boldsymbol{N}(\neg\phi_1 \vee \neg\phi_2))$, it follows that both $\boldsymbol{K}(\neg\phi_1 \vee \neg\phi_2)$ and $\boldsymbol{N}(\neg\phi_1 \vee \neg\phi_2)$ are in Γ. But this contradicts Axiom 5, since $\phi_1 \wedge \phi_2$ is a propositionally consistent objective formula.

For part (b), the proof proceeds by induction on the structure of α. The statement holds trivially for atomic propositions, conjunctions, and negations. In the case of $\boldsymbol{K}\alpha$, we proceed by the following chain of equivalences:

$\boldsymbol{K}\alpha \in \Gamma$
iff for all $\Gamma' \in \mathcal{C}_K^\Gamma$, we have $\alpha \in \Gamma'$
iff for all $\Gamma' \in \mathcal{C}_K^\Gamma$, we have $e_K^{\Gamma'}, e_N^{\Gamma'}, w_{\Gamma'} \models^x \alpha$ (by induction)
iff for all $w_{\Gamma'} \in e_K^\Gamma$, we have $e_K^\Gamma, e_N^\Gamma, w_{\Gamma'} \models^x \alpha$ (by Lemma 9.3.9)
iff $(e_K^\Gamma, e_N^\Gamma, w_\Gamma) \models^x \boldsymbol{K}\alpha$.
The case $\boldsymbol{N}\alpha$ is completely symmetric. ∎

The completeness result now follows easily. Let α be a consistent formula and Γ a maximally consistent set of sentences containing α. $e_K^\Gamma, e_N^\Gamma, w_\Gamma \models^x \alpha$ then follows immediately from Lemma 9.3.10. ∎

Finally, since, by Corollary 9.3.6, validity and x-validity are one and the same, we obtain

Corollary 9.3.11: *For all propositional sentences α, if $\models \alpha$ then $\vdash \alpha$.*

9.4 Incompleteness

As already mentioned in the introduction, the axioms are incomplete for the full language, that is, there are sentences which are valid in $\mathcal{OL}$ but which cannot be derived. We will not prove this result here, but rather discuss the ideas behind it in an informal way.

The reasons for the incompleteness can essentially all be traced back to Axiom 5:

$(\boldsymbol{N}\phi \supset \neg\boldsymbol{K}\phi)$, where ϕ is any objective sentence such that $\not\models \phi$.

To begin with, note that the axiom is already problematic for reasons other than completeness. Proof theories normally require axioms to be recursive, that is, it should be decidable whether any given sentence is an instance of an axiom. Clearly, Axiom 5 violates this requirement since the set of non-valid sentences in $\mathcal{L}$ is not even recursively enumerable. As we will see below, this deficiency is really inescapable in $\mathcal{OL}$, that is, no complete axiom system can be recursive.

The second and perhaps more serious problem with Axiom 5 is that it is simply too weak. In a nutshell, only considering objective non-valid sentences is just not good enough in a logic where nested beliefs do not reduce to non-nested ones, a property of $\mathcal{KL}$ and hence of $\mathcal{OL}$, which we proved in Chapter 4.[6] In fact, a sentence quite similar to the one which we used to show irreducibility can be employed to prove incompleteness. Let

$\zeta = \exists x[P(x) \wedge \neg\boldsymbol{K}P(x)] \vee \exists x[\neg P(x) \wedge \boldsymbol{K}P(x)]$.

6 In the propositional completeness proof this problem did not arise since propositional nested beliefs indeed reduce.

Thus ζ says that either there is a P which is not believed to be a P, or there is a non-P which is believed to be a P. We then obtain the following result.

Lemma 9.4.1: $(N\zeta \supset \neg K\zeta)$ *is valid.*

Proof: Suppose $e \models N\zeta$. Let $A = \{n \mid n$ is a standard name and $e \models KP(n)\}$ and let w be a world state such that $w \models P(n)$ iff $n \in A$. It is easy to see that $e, w \models \neg\zeta$. Since $e \models N\zeta$, it must be the case that $w \in e$. Thus, $e \models \neg K\zeta$. ∎

While valid, it can be shown that the sentence is not derivable from the axioms by devising a slightly different semantics where all of the axioms of $\mathcal{OL}$ are sound, but where $(N\zeta \supset \neg K\zeta)$ is not valid. Incompleteness then follows immediately because no non-valid sentence can be derived from sound axioms.

Of course, the question remains what a complete axiomatization would look like. We already remarked that the set of instances of Axiom 5 is not recursively enumerable (r.e.). Even without knowing what a complete axiomatization might look like, it is easy to see that it cannot be recursive.

Lemma 9.4.2: *Every complete axiomatization of* $\mathcal{OL}$ *is non-recursive.*

Proof: Suppose there were a recursive complete axiomatization of $\mathcal{OL}$. Then the set of non-valid objective formulas would be r.e., since we could generate them by generating all the objective formulas ϕ such that $(N\phi \supset \neg K\phi)$ is provable. Since the set of non-valid objective formulas is co-r.e., this is a contradiction. ∎

Given the non-recursive nature of the axioms, there is, in a sense, a trivial solution to the problem simply by declaring every valid sentence an axiom. Of course, such a "proof theory" is completely useless since it does not give us any new insights into the logic. Instead we would expect axioms to be natural in that they at least have a compact representation. We do not know whether there is such a natural proof-theoretic account of the logic, at least within first-order modal logic. As the following results suggest, if there is one, it may be hard to find.

Recall that the incompleteness proof proceeds by showing that, for a particular basic formula ζ, the formula $(N\zeta \supset \neg K\zeta)$ is valid yet not provable from the axioms. The latter formula almost looks like an instance of Axiom 5. It is not, of course, since 5 would apply only if the formula ζ were objective. The obvious idea, namely to strengthen Axiom 5 by allowing it to range over all non-valid basic sentences, can easily be dismissed. For example, consider the subjective sentence $KP(n)$ for some predicate P and standard

name n. $\boldsymbol{K}P(n)$ is obviously not valid, yet $(\boldsymbol{NK}P(n) \supset \neg\boldsymbol{KK}P(n))$ is not valid. In fact, $\boldsymbol{NK}P(n) \equiv \boldsymbol{KK}P(n)$ is easily derivable from the axioms (using 2) and is therefore valid.

But what about basic sentences that are not subjective like the sentence ζ used above? In other words, do we obtain a complete axiomatization if we replace Axiom 5 by the following Axiom 5′?

(5′.) $(\boldsymbol{N}\alpha \supset \neg\boldsymbol{K}\alpha)$, where α is a basic non-subjective sentence such that $\not\models \alpha$.

Since ζ is basic, non-subjective, and not valid, the offending sentence $(\boldsymbol{N}\zeta \supset \neg\boldsymbol{K}\zeta)$ would now come out trivially as a theorem. Unfortunately, 5′ does not solve the problem either, since restricting the axiom to non-subjective basic sentences is still unsound. To see this, consider the formula

$$\xi = \forall x(P(x) \supset \boldsymbol{K}P(x)),$$

which is obviously not valid. However,

Lemma 9.4.3: $(\boldsymbol{N}\xi \supset \neg\boldsymbol{K}\xi)$ *is not valid.*

Proof: Let e_P consist of all world states w such that $w \models \forall x\, P(x)$. Clearly e_P is maximal. We now show that $e_P \models \boldsymbol{K}\xi \wedge \boldsymbol{N}\xi$. It is easy to see that $e_P, w \models (\boldsymbol{K}(\forall x\, P(x)) \supset \xi)$ for all world states w. Since $e_P \models \boldsymbol{K}(\forall x\, P(x))$, it follows that $e_P, w \models \xi$ for all world states w. This means that $e_P \models \boldsymbol{K}\xi \wedge \boldsymbol{N}\xi$. Hence $(\boldsymbol{N}\xi \supset \neg\boldsymbol{K}\xi)$ is not valid. ∎

Although, as we just showed, $(\boldsymbol{N}\xi \supset \neg\boldsymbol{K}\xi)$ is not valid, there is a sense in which it just misses being valid. As we now show, the only time it fails to be valid is when every standard name is known not to satisfy P (as was the case for the set e_P of world states considered in Lemma 9.4.3).

Lemma 9.4.4: $(\neg\boldsymbol{K}(\forall x\, P(x)) \supset (\boldsymbol{N}\xi \supset \neg\boldsymbol{K}\xi))$ *is valid.*

Proof: Let e be any maximal set of world states such that $e \models \neg\boldsymbol{K}(\forall x\, P(x)) \wedge \boldsymbol{N}\xi$. Since $e \models \neg\boldsymbol{K}(\forall x\, P(x))$, there is a standard name n^* such that $e \models \neg\boldsymbol{K}P(n^*)$. Since $e \models \boldsymbol{N}\xi$, it follows that for all $w' \notin e$, we have $e, w' \models (\forall x(P(x) \supset \boldsymbol{K}P(x)))$. In particular, this means that for all $w' \notin e$, we must have $w' \models \neg P(n^*)$. Thus, there must be some $w \in e$ such that $w \models P(n^*)$. Clearly $e, w \models \neg\xi$, so $e \models \neg\boldsymbol{K}\xi$, as desired. ∎

These lemmas suggest that it may not be easy to find an extension of Axiom 5 that would cover the counterexample, let alone lead to a complete axiomatization.

9.5 Bibliographic notes

In [117] an alternative completeness proof for the propositional case is given, which uses the original semantics of $\mathcal{OL}$. Rosati [163, 164] investigated the computational complexity of only-knowing in the propositional case. He presents an algorithm which decides satisfiability in propositional $\mathcal{OL}$ in nondeterministic polynomial time using an NP-oracle for propositional satisfiability. In other words, the problem is in Σ_2^p (the second level of the polynomial hierarchy). Rosati also shows that the problem is in fact Σ_2^p-complete, which follows easily because of the close connection between $\mathcal{OL}$ and AEL (discussed in the next chapter) and a result by Gottlob [54], who showed that determining whether a formula has a stable expansion is Σ_2^p-hard.

The alternative semantics for $\mathcal{OL}$ with overlapping sets of world states was first introduced in [60]. The proof that the axioms are incomplete in the first-order case appeared in [59]. Most of the material of Section 9.4 is taken from there.

9.6 Where do we go from here?

The investigations regarding a proof theory of $\mathcal{OL}$ make up the first chapter of Part II of this book, which covers special topics extending our basic logic of knowledge bases in various ways. Naturally, there are many open research issues which are only touched upon lightly or not at all. For this reason, we include a paragraph or more in this and the following chapters to draw attention to some of the open issues.

The most immediate open question remaining regarding the proof theory is whether there is a finite set of axioms which is complete for all of $\mathcal{OL}$. Besides that, it seems also interesting to ask whether there are classes of sentences for which the given axioms are actually complete. Clearly, this is true for sentences of $\mathcal{KL}$ or those mentioning only $\boldsymbol{N}$. But what about sentences involving both $\boldsymbol{K}$ and $\boldsymbol{N}$ (or $\boldsymbol{O}$). For example, what about sentences of the form $(\boldsymbol{O}\text{KB} \supset \boldsymbol{K}\alpha)$, where KB is objective or, more generally, where KB is a determinate sentence?

9.7 Exercises

1. Let KB be any sentence such that $\not\models \text{KB} \supset b$ for some atomic sentence b. Show that $(\boldsymbol{O}[\text{KB} \wedge (\neg \boldsymbol{K} b \supset \neg b)] \supset \boldsymbol{K}\neg b)$ follows from the axioms. (This is a formalization of the "older-brother example" considered in the next chapter.)
2. In the propositional case, the soundness of Axiom 5 depends on the fact that there are *infinitely many* atomic propositions.

(a) Give an example where Axiom 5 fails in the case where there are only finitely many atomic propositions in the language.

(b) Show that in this case $\boldsymbol{O}$ can be expressed using $\boldsymbol{K}$ alone.
Hint: Use the fact that each world state can be represented completely by a sentence as long as there are only finitely many propositions.

10 Only-Knowing and Autoepistemic Logic

Up to now we have mainly focussed on knowledge bases (or arguments of $\boldsymbol{O}$, for that matter) that uniquely determine an epistemic state. We saw that this restricted use of $\boldsymbol{O}$ is sufficient to completely characterize the interaction routines **ASK** and **TELL** developed earlier. While our intuitions about $\boldsymbol{O}$ are certainly strongest in the case of determinate sentences, going beyond them not only helps us deepen our understanding of $\boldsymbol{O}$, but it also allows us to demonstrate a close connection between only-knowing and autoepistemic logic (AEL), originally introduced by R. Moore to capture certain forms of nonmonotonic or default reasoning.

Defaults are assumptions which are not based on facts about the world but rather on conventions, statistical information, and the like. For example, most people agree that birds generally fly. So, if presented with a particular bird called Tweety, it seems perfectly reasonable to assume that Tweety flies. Of course, later information may contradict this assumption, for example, if we find out that Tweety is a stuffed bird or an ostrich. In this case, we are more than willing to retract our previous belief about Tweety's flying ability. Notice that the use of defaults has the effect that the set of beliefs may grow *nonmonotonically* with the information obtained about the world. In other words, by adding new facts to our knowledge base we may be forced to retract beliefs held previously about the world. This is why reasoning by default is generally referred to as nonmonotonic reasoning, a term which also stands for the whole research area which has investigated the fundamental principles underlying this type of reasoning since the 1980s.

AEL represents one branch of this endeavour. The idea, in a nutshell, is to interpret defaults such as *birds generally fly* epistemically. Roughly, one is willing to assume that a particular bird flies provided one's own knowledge about the world does not conflict with this assumption.

In the following, we will first demonstrate, by way of example, how autoepistemic reasoning is modeled in $\mathcal{OL}$. Then we will show how $\mathcal{OL}$ not only captures Moore's original ideas in a precise sense but also extends it substantially, mainly because we are using a more expressive language.

10.1 Examples of autoepistemic reasoning in $\mathcal{OL}$

To begin let us consider the following simple example, originally due to Moore.

> Suppose Bob is the oldest child in his family and someone asks him whether he has an older brother. Naturally he would answer no and, asked to explain

his reasoning, Bob may answer as follows: "If I had an older brother, I would certainly know about it. And since I do not know that I have an older brother, I conclude I do not have one."

Note what is happening here. Bob draws a conclusion not based on factual knowledge about the world but based on his ignorance (not knowing about an older brother), which is why this form of reasoning is called autoepistemic. The first sentence of Bob's explanation really expresses a default assumption. It is a quite reasonable one to make, but it can be defeated by new information.[1]

The logic $\mathcal{OL}$ allows us to formalize the example in a natural way. Let b stand for "Bob has an older brother" and let KB be Bob's knowledge base consisting of objective sentences such that $\not\models (\mathrm{KB} \supset b)$. We can express the default as $\delta = (\neg \boldsymbol{K} b \supset \neg b)$. If we then assume $\mathrm{KB} \wedge \delta$ is all Bob knows then we get the desired result, that is, $\neg b$ is believed.

Example 10.1.1: $(\boldsymbol{O}(\mathrm{KB} \wedge \delta) \supset \boldsymbol{K}\neg b)$ is a valid sentence.

Proof: Let $e = \{w \mid w \models \mathrm{KB} \wedge \neg b\}$. Clearly, $e \models \boldsymbol{K}\neg b$. Let e^* be any set of worlds such that $e^* \models \boldsymbol{O}(\mathrm{KB} \wedge \delta)$. It suffices to show that $e^* = e$. To show that $e \subseteq e^*$, let $w \in e$. Then $w \models \mathrm{KB} \wedge \neg b$ and, hence, $e^*, w \models \mathrm{KB} \wedge \delta$, from which $w \in e^*$ follows. Conversely, let $w \in e^*$. Then $w \models \mathrm{KB}$. Note that $e^* \models \neg \boldsymbol{K} b$. (For assume otherwise and let $w^* \notin e^*$ such that $w^* \models \mathrm{KB} \wedge \neg b$. Then $e^*, w^* \models \mathrm{KB} \wedge \delta$ and, hence, $w^* \in e^*$, a contradiction.) Then, since $e^*, w \models \delta$, we have $w \models \neg b$ and, hence, $w \in e$. ∎

An important characteristic of defaults is that the objective beliefs of an agent may change nonmonotonically if new information is added to the knowledge base. In the example, Bob initially knows KB and, by default, $\neg b$. When Bob's mom finally tells Bob the truth about the extent of his immediate family, Bob may add b to his knowledge base overriding his previous default belief. Formally, $(\boldsymbol{O}(\mathrm{KB} \wedge b \wedge \delta) \supset \boldsymbol{K} b)$ is valid, which can be derived using ordinary reasoning about $\boldsymbol{K}$.

Let us now turn to more complex cases of defaults with quantifiers. Actually, we already saw examples of those in our previous discussion of determinate sentences (Section 8.5). Recall that we formalized the closed world assumption for a particular predicate P using the sentence $\gamma = (\forall x P(x) \supset \boldsymbol{K} P(x))$. Note that we can rewrite γ as $(\forall x \neg \boldsymbol{K} P(x) \supset \neg P(x))$, which is more suggestive of a default saying that P is a assumed to be false unless known otherwise. In this sense, the older-brother example is nothing more than an instance of the closed world assumption. Closed world reasoning, as it is commonly applied in databases, is perhaps the simplest form of default reasoning. But,

1 In fact, one of the authors has two older brothers whose existence was not revealed to him for a long time.

of course, defaults can also be used to derive positive facts about the world. The classic example is about birds and their ability to fly. In particular, one would like to conclude that any bird such as the infamous *Tweety* can fly unless known otherwise. One way to express the appropriate default is by using sentences like

$$\forall x[Bird(x) \wedge \neg \boldsymbol{K} \neg Fly(x) \supset Fly(x)]$$

within the scope of an $\boldsymbol{O}$ operator. If we let δ stand for this sentence, we obtain the following:

Example 10.1.2: Assume that KB $= \{Bird(tweety)\}$. Then the following sentences are valid:

1. $\boldsymbol{O}(\text{KB} \wedge \neg Fly(tweety) \wedge \delta) \supset \boldsymbol{K} \neg Fly(tweety)$
2. $\boldsymbol{O}(\text{KB} \wedge Fly(tweety) \wedge \delta) \supset \boldsymbol{K} Fly(tweety)$
3. $\boldsymbol{O}(\text{KB} \wedge \delta) \supset \boldsymbol{K} Fly(tweety)$.

Proof: (1) and (2) follow easily using the fact that $(\boldsymbol{O}\alpha \supset \boldsymbol{K}\alpha)$ is valid.

To show (3), let $e \models \boldsymbol{O}(\text{KB} \wedge \delta)$. We first show that $e \models \neg \boldsymbol{K} \neg Fly(tweety)$. Let w^* be any world state such that $w^* \models Bird(tweety) \wedge \forall x Fly(x)$. Then $e, w^* \models \text{KB} \wedge \delta$ and, hence, $w^* \in e$. Since $w^* \models Fly(tweety)$, we obtain $e \models \neg \boldsymbol{K} \neg Fly(tweety)$.

Now we show that $e \models \boldsymbol{K} Fly(tweety)$, that is, for every $w \in e$, $w \models Fly(tweety)$. Let $w \in e$. Then $e, w \models (Bird(tweety) \wedge \neg \boldsymbol{K} \neg Fly(tweety) \supset Fly(tweety))$. Clearly, $w \models Bird(tweety)$. By the above, we also have $e \models \neg \boldsymbol{K} \neg Fly(tweety)$. Therefore, $w \models Fly(tweety)$. ∎

This example shows that Tweety's flying is indeed the default: if his flying ability is specified explicitly, then this works out properly (cases 1 and 2); otherwise, flying is taken as the default (case 3).

Note, however, that the proof uses the fact that there are worlds where all things fly, which is certainly true when KB $= \{Bird(tweety)\}$. However, this condition is too strong. We should be able to get the default for Tweety even if there are some flightless birds. So what happens if the KB implies of some bird that it is flightless? The answer is that the default still works properly, but for a slightly different reason. In the following, both *tweety* and *chilly* are meant to be distinct standard names.

Example 10.1.3: Assume that

$$\text{KB} = \{Bird(tweety), Bird(chilly), \neg Fly(chilly)\}.$$

Then

$$\boldsymbol{O}(\text{KB} \wedge \delta) \supset \boldsymbol{K} \forall x[Bird(x) \wedge (x \neq chilly) \supset Fly(x)]$$

is valid and thus, $(\boldsymbol{O}(\text{KB} \wedge \delta) \supset \boldsymbol{K}Fly(tweety))$ is valid.

Proof: We leave the proof as an exercise. ∎

Note that the default belief that Tweety flies is based on the default belief that any bird other than Chilly flies. In fact,with no information at all, it will be assumed that all birds fly: the sentence $(\boldsymbol{O}\delta \supset \boldsymbol{K}\forall x[Bird(x) \supset Fly(x)])$ is valid. In many applications this is too strong; we might not want to infer anything about distal unknown birds. One way to do this is to write the default as

$$\forall x[\boldsymbol{K}Bird(x) \wedge \neg\boldsymbol{K}\neg Fly(x) \supset Fly(x)].$$

This makes the default apply only to the *known* birds. However, it does have disadvantages compared to the previous form of default. For example, if $\text{KB} = \{\exists x(Bird(x) \wedge Yellow(x))\}$, then using this form of default, we would *not* conclude by default that there was a yellow bird that flies. Although its *existence* is known, the bird in question is not (yet) a known bird.

A nice property of the examples considered so far is that the KB together with the default is *determinate* as defined previously, that is, there is a unique corresponding epistemic state. In particular, the effect of the default is to add information to the KB, in this case, information about birds' flying ability. This "filling in the blanks" is precisely what one would expect from a default. Unfortunately the desired effect does not always obtain, and to see why, let us go back to our original birds-fly default δ, which applies to all birds, not just the known ones (although the same applies to the more restricted form). Note that so far it has always been the case that (1) the KB implies that Tweety is a bird and (2) it does not imply that Tweety is flightless. Unfortunately, these conditions are not sufficient for the default to go through. For suppose that

$$\text{KB} = \{Bird(tweety), Bird(chilly), (\neg Fly(chilly) \vee \neg Fly(tweety))\}.$$

Then the KB does not imply that Tweety is flightless, but it would be inappropriate to assume by default that it can fly, since by symmetry we could infer the same of Chilly, contradicting the fact that one of them is flightless. A similar complication occurs if

$$\text{KB} = \{Bird(tweety), \exists x(Bird(x) \wedge \neg Fly(x))\}.$$

Again, if we are prepared to infer that Tweety flies, by symmetry, we should be able to do likewise for any bird, and thus come to the conclusion that all birds fly, again contradicting the belief. The trouble with the two KBs above is that the default δ is actually believed to be false, that is, the sentence $(\boldsymbol{O}\text{KB} \supset \boldsymbol{K}\neg\delta)$ is valid. In both cases the KB implies that there is a flightless bird but it does not specify which; so, if this is *all* that is known, then it *is* believed that there is a flightless bird whose identity is not known, which is $\neg\delta$. So what happens in these cases if we insist that $\boldsymbol{O}[\text{KB} \wedge \delta]$ is true? That is, what happens when we

believe KB and δ and nothing else, even though believing KB alone implies believing $\neg\delta$? The answer, in short, is that (KB $\wedge\ \delta$) is no longer determinate, that is, it fails to specify completely what is and is not believed. More specifically, we have:

Example 10.1.4: Let KB $= \{Bird(tweety), Bird(chilly), (\neg Fly(chilly) \vee \neg Fly(tweety))\}$ and let $Exc(x)$ be an abbreviation for $Bird(x) \wedge \neg Fly(x)$. Then the sentence $\boldsymbol{O}$[KB $\wedge\ \delta$] is logically equivalent to

$$\boldsymbol{O}[\text{KB} \wedge \forall x(Exc(x) \equiv x = tweety)] \ \vee\ \boldsymbol{O}[\text{KB} \wedge \forall x(Exc(x) \equiv x = chilly)].$$

Proof: To prove the if direction, let $e \models \boldsymbol{O}[\text{KB} \wedge \forall x(Exc(x) \equiv x = tweety)]$. (The other case is handled the same way.) We need to show that $e \models \boldsymbol{O}[\text{KB} \wedge \delta]$. If $w \in e$, then clearly $w \models$ KB. Also, $e, w \models \delta$ follows because all birds other than Tweety fly in w and Tweety is known not to fly. Conversely, let $w \notin e$ and assume that $w \models$ KB. Thus $w \not\models \forall x(Exc(x) \equiv x = tweety)$, that is, either there is an exceptional bird n other than Tweety or Tweety is not an exceptional bird. In the first case, $w \models Bird(n) \wedge \neg\boldsymbol{K}\neg Fly(n) \wedge \neg Fly(n)$. In the second case, we have the same with n replaced by Chilly because KB requires one of Tweety and Chilly not to fly. In either case, $e, w \not\models \delta$, and we are done.

For the only-if direction, let $e \models \boldsymbol{O}[\text{KB} \wedge \delta]$. First note that $e \models \boldsymbol{K}\neg Fly(tweety) \vee \boldsymbol{K}\neg Fly(chilly)$ which follows by the default and the fact that either Tweety or Chilly does not fly. Thus let us assume that $e \models \boldsymbol{K}\neg Fly(tweety)$. It suffices to show that $e = e^*$ with $e^* = \{w \mid w \models \text{KB} \wedge \forall x(Exc(x) \equiv x = tweety)\}$. (The other case is handled the same way with Tweety replaced by Chilly.) Let $w \in e^*$. Since all birds other than Tweety fly at w and Tweety is known not to fly by assumption, we obtain $e, w \models \text{KB} \wedge \delta$ and, hence, $w \in e$. Conversely, suppose $w \in e$. By assumption, Tweety is known to be an exceptional bird at e and is therefore exceptional at w. Any bird n other than Tweety flies at w because of δ and the fact that $e \models \neg\boldsymbol{K}\neg Fly(n)$, which follows from $e^* \subseteq e$. Hence $w \models \forall x(Exc(x) \equiv x = tweety)$ and, therefore, $w \in e^*$. ∎

This says that only-knowing KB and the default is the same as only-knowing KB and that Tweety is the only flightless bird *or* only-knowing KB and that Chilly is the only flightless bird. But the KB and the default together are not sufficient to specify exactly what is believed; they *describe* what is believed, but do not determine it. They do, however, determine what is *common* to both epistemic states. For example,

$$\boldsymbol{O}[\text{KB} \wedge \delta] \supset \boldsymbol{K}\forall x[Exc(x) \supset (x = tweety \vee x = chilly)]$$

is valid.

It is not hard to see that the default may even lead to an infinite number of compatible epistemic states. For instance, let KB $= \{\exists x Exc(x)\}$. The result here is that the sentence

$\boldsymbol{O}[\text{KB} \wedge \delta] \equiv \exists y \boldsymbol{O}[\text{KB} \wedge \forall x(Exc(x) \equiv (x = y))]$ is valid, by an argument similar to the one above. In other words, only knowing that birds fly by autoepistemic default and that there is an exceptional bird does not determine exactly what is believed; however, it only happens if, for some bird, all that is known is that this bird is the only exceptional one. In this case, there is a different epistemic state for each standard name.

Finally, instead of multiple epistemic states there is also the case, as we saw already in Chapter 8, that a sentence does not correspond to any epistemic state at all, that is, it can never be all that is known like $\boldsymbol{K}\phi$, since $\models \neg \boldsymbol{O}[\boldsymbol{K}\phi]$ (see Corollary 8.3.6).

10.2 Stable sets and stable expansions

We now turn to the close relationship between only-knowing and Moore's original formulation of autoepistemic logic. There are two notions central to AEL, *stable sets* and *stable expansions*. We will give precise definitions below, but let us first look at these notions informally. Both have in common that they are syntactic characterizations of an agent's beliefs. A stable set simply states three basic conditions the beliefs of an ideal rational agent should satisfy: closure under logical consequence, positive and negative introspection. Stable expansions then define those sets of beliefs that are stable and in some sense derive from a set of assumptions A. In other words, a stable expansion describes the beliefs an ideal rational agent might hold provided her knowledge base consists of the sentences in A. The following properties will be established relating AEL and only knowing:

- Belief sets and stable sets coincide.
- The stable expansions of a sentence α are precisely those belief sets which result from only-knowing α.
- While AEL was originally only defined for a propositional language, $\mathcal{OL}$ provides a natural quantificational generalization.

The notion of stability depends on a definition of first-order consequence, so we should be clear about this concept first. The idea is simple: α is a first-order consequence of Γ when Γ implies α by virtue of the rules of ordinary first-order logic alone, that is, without using the rules for $\boldsymbol{K}$ or for $\boldsymbol{O}$ even over sentences containing these operators. One way to formalize this is to think of sentences like $\boldsymbol{K}\alpha$ and $\boldsymbol{O}\alpha$ as new atomic sentences so that there is no forced relationship between the truth value of, for example, $\boldsymbol{K}\alpha$ and $\neg\boldsymbol{K}\neg\neg\alpha$. Although the conjunction of these two is not satisfiable, this depends on the semantics of $\boldsymbol{K}$, and so we want to say that it is first-order satisfiable.

More precisely, let θ be any function from sentences of the form $\boldsymbol{K}\alpha$ or $\boldsymbol{O}\alpha$ to $\{0, 1\}$, and w any world. We will say that a pair θ and w *first-order satisfies* a sentence α, which we write $\theta, w \models_{\text{FOL}} \alpha$ according to these rules:

1. $\theta, w \models_{\text{FOL}} P(t_1, \ldots, t_k)$ iff $w[P(n_1, ...n_k)] = 1$, where $n_i = w(t_i)$;
2. $\theta, w \models_{\text{FOL}} t_1 = t_2$ iff $w(t_1)$ is the same name as $w(t_2)$;
3. $\theta, w \models_{\text{FOL}} \neg\alpha$ iff $\theta, w \not\models_{\text{FOL}} \alpha$;
4. $\theta, w \models_{\text{FOL}} \alpha \wedge \beta$ iff $\theta, w \models_{\text{FOL}} \alpha$ and $\theta, w \models_{\text{FOL}} \beta$;
5. $\theta, w \models_{\text{FOL}} \exists x\alpha$ iff for some n, $\theta, w \models_{\text{FOL}} \alpha^x_n$;
6. $\theta, w \models_{\text{FOL}} \boldsymbol{K}\alpha$ iff $\theta(\boldsymbol{K}\alpha) = 1$;
7. $\theta, w \models_{\text{FOL}} \boldsymbol{O}\alpha$ iff $\theta(\boldsymbol{O}\alpha) = 1$.

We will say that Γ is *first-order satisfiable* iff some θ and w first-order satisfies it. Finally, we will say that Γ *first-order implies* α, which we write $\Gamma \models_{\text{FOL}} \alpha$, iff $\Gamma \cup \{\neg\alpha\}$ is not first-order satisfiable. Clearly satisfiability implies first-order satisfiability, but not the converse.

We are now in a position to formally introduce stable sets and expansions and relate them to only-knowing. Since AEL only deals with basic sentences, we focus on those first. In Section 10.6, we will see that all definitions and results carry over naturally if we allow non-basic sentences as well.

10.3 Relating epistemic states to stable sets and expansions

With first-order consequence the definition of a stable set is now very simple.

Definition 10.3.1: A set of basic sentences Γ is *stable* iff

1. If $\Gamma \models_{\text{FOL}} \alpha$, then $\alpha \in \Gamma$.[2]
2. If $\alpha \in \Gamma$, then $\boldsymbol{K}\alpha \in \Gamma$.
3. If $\alpha \notin \Gamma$, then $\neg\boldsymbol{K}\alpha \in \Gamma$.

Stability merely states in a rigorous way that beliefs are closed under perfect logical reasoning and introspection. Since we have been making these assumptions all along, it is clear that every basic belief set is also a stable set. Below we will show that the converse is also true, that is, stable sets correspond exactly to basic belief sets. But first we need a result stating that for certain sets of sentences satisfiability and first-order satisfiability coincide.

Definition 10.3.2: A set Σ is an *adjunct* of a set Γ iff $\Sigma = \{\boldsymbol{K}\alpha \mid \alpha \text{ is basic and } \alpha \in \Gamma\} \cup \{\neg\boldsymbol{K}\alpha \mid \alpha \text{ is basic and } \alpha \notin \Gamma\}$.

2 In other words we are requiring Γ to be closed under first-order implication. Moore used propositional logical consequence since he only dealt with a propositional language.

Lemma 10.3.3: *If Σ is an adjunct of a basic belief set Γ, then for any subjective sentence σ, either $\Sigma \models \sigma$ or $\Sigma \models \neg\sigma$.*

Proof: Suppose Γ is a basic belief set for some maximal e and suppose that $e \models \sigma$. Then any maximal e' such that $e' \models \Sigma$ must have the same basic belief set as e. By Lemma 8.2.1, $e' \models \sigma$ follows and, consequently, $\Sigma \models \sigma$. The case with $\neg\sigma$ is analogous. ∎

Theorem 10.3.4: *Suppose Δ is a set of basic sentences that contains an adjunct to a stable set. Then Δ is satisfiable iff it is first-order satisfiable.*

Proof: The only-if direction is immediate. So suppose that Δ contains an adjunct to a stable set Γ and is first-order satisfiable, and that $\theta, w \models_{\text{FOL}} \Delta$. Define e as $\{w' \mid \theta, w' \models_{\text{FOL}} \Gamma\}$. We will show by induction that for any w' and any basic α, $e, w' \models \alpha$ iff $\theta, w' \models_{\text{FOL}} \alpha$.

This clearly holds for atomic sentences, equalities, and by induction, for negations, conjunctions, and quantifications. Now suppose that $\theta(\boldsymbol{K}\alpha) = 1$. Therefore, $\neg\boldsymbol{K}\alpha \notin \Delta$, and so $\alpha \in \Gamma$. Thus, for every $w' \in e$, $\theta, w' \models_{\text{FOL}} \alpha$ and so by induction, $e, w' \models \alpha$ and so, $e \models \boldsymbol{K}\alpha$. Conversely, suppose that $\theta(\boldsymbol{K}\alpha) = 0$. Therefore, $\boldsymbol{K}\alpha \notin \Delta$, and so $\alpha \notin \Gamma$. But Γ is closed under first-order implication, so $\Gamma \cup \{\neg\alpha\}$ is first-order satisfiable. Therefore, there must be some θ^* and some w' such that $\theta^*, w' \models_{\text{FOL}} \Gamma \cup \{\neg\alpha\}$. But θ and θ^* can only differ on non-basic sentences since for every basic α, either $\boldsymbol{K}\alpha \in \Gamma$ or $\neg\boldsymbol{K}\alpha \in \Gamma$. Thus, $\theta, w' \models_{\text{FOL}} \Gamma \cup \{\neg\alpha\}$. This means that $w' \in e$, and so there is a $w' \in e$ such that $\theta, w' \models_{\text{FOL}} \neg\alpha$, and by induction $e, w' \models \neg\alpha$. Therefore, $e \models \neg\boldsymbol{K}\alpha$.

Thus, for every w', $e, w' \models \alpha$ iff $\theta, w' \models_{\text{FOL}} \alpha$. This establishes that $e, w \models \Delta$, and so Δ is satisfiable. ∎

One simple consequence of this theorem is that it is not necessary to use first-order implication when dealing with (supersets of) adjuncts to stable sets:

Corollary 10.3.5: *Suppose Δ is a set of basic sentences that contains an adjunct to a stable set. Then for any basic α, $\Delta \models \alpha$ iff $\Delta \models_{FOL} \alpha$.*

Proof: Immediate from the theorem. ∎

Now we can show that stable sets and basic belief sets are one and the same.

Theorem 10.3.6: *Suppose Γ is a set of basic sentences. Then Γ is stable iff Γ is a basic belief set.*

Proof: The if direction is straightforward: the first condition is a result of the logical properties of a reasoner, and the last two are a result of its introspective capabilities.

Conversely, suppose Γ is stable. There are two cases. If Γ is satisfiable, then some $e, w \models \Gamma$. For any basic α, if $\alpha \in \Gamma$, then $\boldsymbol{K}\alpha \in \Gamma$, and so $e \models \boldsymbol{K}\alpha$; if $\alpha \notin \Gamma$, then $\neg\boldsymbol{K}\alpha \in \Gamma$, and so $e \models \neg\boldsymbol{K}\alpha$. Thus, $\alpha \in \Gamma$ iff $e \models \boldsymbol{K}\alpha$, and so Γ is a basic belief set for e. Suppose on the other hand that Γ is unsatisfiable. By properties (2) and (3) of stability, Γ must contain the adjunct to Γ. Then by Theorem 10.3.4, Γ is not first-order satisfiable. So for every basic α, $\Gamma \models_{\text{FOL}} \alpha$, and by definition of stability, $\alpha \in \Gamma$. Thus, Γ contains every basic sentence. It is therefore the basic belief set of the empty set of worlds. ∎

It has long been known that stable sets, when restricted to propositional sentences, are uniquely determined by their objective subsets. With quantifiers and, in particular, quantifying-in, this is no longer the case.[3]

Theorem 10.3.7: *Stable sets are in general not uniquely determined by their objective subsets.*

Proof: The result follows easily from Theorem 4.6.2, which says that there are two epistemic states e_1 and e_2 whose corresponding basic belief sets agree on all objective sentences but disagree on $\boldsymbol{K}\exists x[P(x) \wedge \neg\boldsymbol{K}P(x)]$. Since, by the previous theorem, stable sets and basic belief sets are one and the same, the theorem follows. ∎

Let us now turn to stable expansions. Roughly, a sentence γ belongs to a stable expansion of a set of basic sentences A if it follows from A using logical reasoning and introspection. Of course, we need to be clear about what we mean by introspection here. The trick is to assume we already know what the stable expansion is and use its adjunct as the characterization of the beliefs that can be inferred by introspection. γ is then simply a logical consequence of A and the adjunct. Formally, we obtain the following fixed-point definition.

Definition 10.3.8: A set of sentences Γ is a *stable expansion* of a set of basic sentences A iff Γ satisfies the fixed-point equation:

$$\Gamma = \{\gamma \mid \gamma \text{ is basic and } A \cup \{\boldsymbol{K}\beta \mid \beta \in \Gamma\} \cup \{\neg\boldsymbol{K}\beta \mid \beta \notin \Gamma\} \models_{\text{FOL}} \gamma\}.$$

The main result of this chapter says that the stable expansion of a sentence α and the basic belief sets that result from only-knowing α are one and the same.

3 If we disallow quantifying-in, we obtain the same results as in the propositional case.

Theorem 10.3.9: *For any basic α and any maximal set of worlds e, $e \models \boldsymbol{O}\alpha$ iff the basic belief set of e is a stable expansion of $\{\alpha\}$.*

Proof: Let e be any maximal set of worlds with Γ as its basic belief set and Σ as the adjunct to Γ. Thus, $e \models \Sigma$. What we want to show is that $e \models \boldsymbol{O}\alpha$ iff Γ is the set of basic sentences that are first-order implied by $\{\alpha\} \cup \Sigma$. Moreover, by Corollary 10.3.5, we can use full logical implication instead of first-order implication since Γ is a stable set. Thus, we need to show that

$$e \models \boldsymbol{O}\alpha \quad \text{iff} \quad \text{for every basic } \beta,\ e \models \boldsymbol{K}\beta \text{ iff } \{\alpha\} \cup \Sigma \models \beta.$$

First assume that $e \models \boldsymbol{O}\alpha$. For the if part, assume that $\{\alpha\} \cup \Sigma \models \beta$. Now let w be any element of e. Since $e \models \boldsymbol{O}\alpha$, $e, w \models \{\alpha\} \cup \Sigma$, and therefore, $e, w \models \beta$. Thus, for any $w \in e$, we have that $e, w \models \beta$, and so $e \models \boldsymbol{K}\beta$.

For the only-if part, assume that $e \models \boldsymbol{K}\beta$. To show that $\{\alpha\} \cup \Sigma \models \beta$, let e' be any maximal set of worlds and w be any world. If $e', w \models \{\alpha\} \cup \Sigma$, then $e' = e$ since Σ is an adjunct of the basic belief set for e by Lemma 10.3.3. Thus, $e, w \models \alpha$ and so $w \in e$, because $e \models \boldsymbol{O}\alpha$. But if $w \in e$, then $e, w \models \beta$, since $e \models \boldsymbol{K}\beta$. Thus for any e' and w, if $e', w \models \{\alpha\} \cup \Sigma$, then $e', w \models \beta$, and so $\{\alpha\} \cup \Sigma \models \beta$.

Now assume that $e \models \boldsymbol{K}\beta$ iff $\{\alpha\} \cup \Sigma \models \beta$. First we need to show that $e \models \boldsymbol{K}\alpha$, but this is immediate since clearly $\{\alpha\} \cup \Sigma \models \alpha$. Next we need to establish that if $e, w \models \alpha$ then $w \in e$. If $e, w \models \alpha$ then $e, w \models \{\alpha\} \cup \Sigma$, since $e \models \Sigma$. Now consider any β such that $e \models \boldsymbol{K}\beta$. We have that $\{\alpha\} \cup \Sigma \models \beta$, and so $e, w \models \beta$. Therefore, by Theorem 6.1.1, we have that $e \approx (e + w)$ and so, because e is maximal, $w \in e$. Thus, for any w, if $e, w \models \alpha$ then $w \in e$, and so $e \models \boldsymbol{O}\alpha$. ∎

So only-knowing a sentence means that what is believed is a stable expansion of that sentence (or, more intuitively, what is believed is derivable from that sentence using first-order logic and introspection alone). This theorem provides for the first time a semantic account for the notion of stable expansion. In addition, we have generalized the notion of a stable expansion to deal with a quantificational language with equality. To summarize, we have the following correspondences:

semantic	**syntactic**
believing	membership in a stable set
basic belief sets	stable sets
only believing	stable expansions

One easy corollary to this theorem relates the number of stable expansions of a sentence to the number of sets of worlds where that sentence is all that is known.

Corollary 10.3.10: *A sentence α has exactly as many stable expansions as there are maximal sets of worlds where $\boldsymbol{O}\alpha$ is true.*

Proof: By Theorem 8.2.1, the mapping between maximal sets of worlds and basic belief sets is bijective. By Theorem 10.3.6, beliefs sets are the stable sets. The correspondence then follows from the above theorem. ∎

What this says, among other things, is that our previous discussions of determinate and non-determinate sentences applies equally well to stable expansions.

10.4 Computing stable expansions

In the previous section, we saw that there was a one-to-one correspondence between the stable expansions of a formula and the epistemic states where the formula is all that is known. In this section, we examine a procedure for calculating these stable expansions or epistemic states in the propositional case. Specifically, we will return a set of objective formulas that represent each of the epistemic states where the given propositional formula is all that is known.

In the following, we will show that for any propositional $\beta \in \mathcal{OL}$, the formula $\boldsymbol{O}\beta$ is equivalent to a finite disjunction of formulas of the form $\boldsymbol{O}\psi$ where ψ is objective. In the process, we will need to substitute subwffs of the form $\boldsymbol{K}\gamma$ or $\boldsymbol{O}\gamma$ in β by either TRUE or FALSE. We begin by enumerating all subwffs $\boldsymbol{K}\gamma_1, \ldots, \boldsymbol{K}\gamma_k$, and $\boldsymbol{O}\gamma_{k+1}, \ldots \boldsymbol{O}\gamma_n$ that appear in β. In the proof below, we will let $\mathcal{OL}_\beta$ mean the subset of $\mathcal{OL}$ whose $\boldsymbol{K}\gamma$ or $\boldsymbol{O}\gamma$ subwffs appear in this list.

Definition 10.4.1: Let $v \in \{0, 1\}^n$. Then for any $\alpha \in \mathcal{OL}_\beta$, $\|\alpha\|_v$ is the objective formula that results from replacing a subwff $\boldsymbol{K}\gamma_i$ or $\boldsymbol{O}\gamma_i$ in α by TRUE if $v_i = 1$ and FALSE if $v_i = 0$.

Lemma 10.4.2: *Let e be an epistemic state, and suppose that $v \in \{0, 1\}^n$ satisfies $v_i = 1$ iff $e \models \boldsymbol{K}\gamma_i$ (or $e \models \boldsymbol{O}\gamma_i$). Then for any w, and any $\alpha \in \mathcal{OL}_\beta$, $e, w \models \alpha$ iff $w \models \|\alpha\|_v$, and consequently, $e \models \boldsymbol{O}\alpha$ iff $e \models \boldsymbol{O}\|\alpha\|_v$.*

Proof: By induction on the length of α. ∎

Lemma 10.4.3: *Suppose e and v are as above, and that for some $\alpha \in \mathcal{OL}_\beta$, we have that $e \models \boldsymbol{O}\alpha$. Then for all $1 \leq i \leq k$, $v_i = 1$ iff $\models (\|\alpha\|_v \supset \|\gamma_i\|_v)$, and for all $k + 1 \leq i \leq n$,*

$v_i = 1$ *iff* $\models (\|\alpha\|_v \equiv \|\gamma_i\|_v)$.

Proof: For the first part with $i \leq k$, in the only-if direction, assume that $v_i = 1$. Now suppose that for any w, $w \models \|\alpha\|_v$. By the lemma above, $e \models \boldsymbol{O}\|\alpha\|_v$, and so $w \in e$. Since $v_i = 1$, we have that $e \models \boldsymbol{K}\gamma_i$, and therefore, $e, w \models \gamma_i$, and again by the same lemma, $w \models \|\gamma_i\|_v$. So $\models (\|\alpha\|_v \supset \|\gamma_i\|_v)$.

In the if-direction, assume that $\models (\|\alpha\|_v \supset \|\gamma_i\|_v)$, and suppose that w is any element of e. Since we have that $e \models \boldsymbol{O}\|\alpha\|_v$, we get that $w \models \|\alpha\|_v$ and so $w \models \|\gamma_i\|_v$, and then by the above lemma, $e, w \models \gamma_i$. Thus, $e \models \boldsymbol{K}\gamma_i$, and so $v_i = 1$.

The second part of the proof with $i > k$ is analogous. ∎

Lemma 10.4.4: *Assume that* $v \in \{0,1\}^n$ *and that for the given* β *we have that for all* $1 \leq i \leq k$, $v_i = 1$ *iff* $\models (\|\beta\|_v \supset \|\gamma_i\|_v)$, *and for all* $k+1 \leq i \leq n$, $v_i = 1$ *iff* $\models (\|\beta\|_v \equiv \|\gamma_i\|_v)$. *Let* $e = \Re[\![\|\beta\|_v]\!]$. *Then for any* $\alpha \in \mathcal{OL}_\beta$ *and any* w, *we have that* $e, w \models \alpha$ *iff* $w \models \|\alpha\|_v$, *and so* $e \models \boldsymbol{O}\beta$.

Proof: By induction on the length of α. For atoms, negations and conjunctions, the argument is clear. If α is $\boldsymbol{K}\gamma_i$, then $e, w \models \alpha$ iff for every $w' \in e$, we have that $e, w' \models \gamma_i$ iff (by induction) for every $w' \in e$, we have that $w' \models \|\gamma_i\|_v$. Since $e = \Re[\![\|\beta\|_v]\!]$, this happens iff $\models (\|\beta\|_v \supset \|\gamma_i\|_v)$, iff $v_i = 1$ iff $\|\alpha\|_v = \text{TRUE}$ iff $w \models \|\alpha\|_v$. The final case with $\boldsymbol{O}\gamma_i$ is analogous. ∎

Theorem 10.4.5: *For any formula* $\beta \in \mathcal{OL}$ *and any epistemic state* e, $e \models \boldsymbol{O}\beta$ *iff there is a* $v \in \{0,1\}^n$ *such that* $e = \Re[\![\|\beta\|_v]\!]$ *and where for all* $1 \leq i \leq k$, $v_i = 1$ *iff* $\models (\|\beta\|_v \supset \|\gamma_i\|_v)$, *and for all* $k+1 \leq i \leq n$, $v_i = 1$ *iff* $\models (\|\beta\|_v \equiv \|\gamma_i\|_v)$.

Proof: In the if direction, we can define the v using e as in Lemma 10.4.2, and then apply Lemma 10.4.3. The only-if direction is an immediate consequence of Lemma 10.4.4. ∎

Corollary 10.4.6: *For any* $\beta \in \mathcal{OL}$, *there are objective wffs* $\psi_1, \ldots, \psi_m, m \geq 0$ *such that* $\models \boldsymbol{O}\beta \equiv (\boldsymbol{O}\psi_1 \vee \cdots \vee \boldsymbol{O}\psi_m)$.

Proof: Let S be the set of all objective wffs of the form $\|\beta\|_v$ where $v \in \{0,1\}^n$ and for all $1 \leq i \leq k$, $v_i = 1$ iff $\models (\|\beta\|_v \supset \|\gamma_i\|_v)$, and for all $k+1 \leq i \leq n$, $v_i = 1$ iff $\models (\|\beta\|_v \equiv \|\gamma_i\|_v)$. Then by the theorem, if $e \models \boldsymbol{O}\beta$, then for some $\psi \in S$, $e \models \boldsymbol{O}\psi$. Furthermore, if $\psi \in S$, and $e = \Re[\![\psi]\!]$, then again by the theorem, $e \models \boldsymbol{O}\beta$. Thus we

Input: any propositional formula $\beta \in \mathcal{OL}$;
Output: a set of objective formulas $\psi_1, \ldots, \psi_m$ satisfying
$\models \boldsymbol{O}\beta \equiv (\boldsymbol{O}\psi_1 \vee \cdots \vee \boldsymbol{O}\psi_m)$.

Procedure
/* Assume that β has subwffs $\boldsymbol{K}\gamma_1, \ldots, \boldsymbol{K}\gamma_k, \boldsymbol{O}\gamma_{k+1}, \ldots, \boldsymbol{O}\gamma_n$. */
$S \leftarrow \{\}$
for $v \in \{0,1\}^n$ **do**
if for all $1 \leq i \leq k$, $v_i = 1$ iff $\models (\|\beta\|_v \supset \|\gamma_i\|_v)$
and for all $k+1 \leq i \leq n$, $v_i = 1$ iff $\models (\|\beta\|_v \equiv \|\gamma_i\|_v)$
then $S \leftarrow S \cup \{\|\beta\|_v\}$
end
return S
end

Figure 10.1: Calculating stable expansions

have that $\boldsymbol{O}\beta$ is logically equivalent to $\vee\{\boldsymbol{O}\psi \mid \psi \in S\}$. ∎

We can also see looking at the proof of this corollary that the m in question can be no larger that 2^n where n is the number of subwffs of the form $\boldsymbol{K}\gamma$ or $\boldsymbol{O}\gamma$ that appear in β.

This then suggests a procedure for generating the epistemic states that satisfy a given propositional formula $O\beta$ by generating a finite set of objective formulas that represent all that is known in each of these states.[4] The procedure appears in Figure 10.1. Because of Corollary 10.3.10, this procedure also generates the stable expansions of any propositional formula. More precisely, the objective formulas returned by the procedure represent the epistemic states whose basic belief sets are the stable expansions.

Finally, the theorem leads us to the conclusion that propositional $\mathcal{OL}$ is reducible, in the sense of Section 4.6: it is possible to reduce any propositional formula involving perhaps nested $\boldsymbol{K}$ or $\boldsymbol{O}$ operators to an equivalent one where the $\boldsymbol{K}$ and the $\boldsymbol{O}$ only dominate objective formulas. The proof of this is left as an exercise.

10.5 Non-reducibility of $\mathcal{OL}$

We have seen in Chapter 4 that $\mathcal{KL}$ is irreducible, that is, there are sentences such as $\boldsymbol{K}[\exists x\, P(x) \wedge \neg \boldsymbol{K} P(x)]$ with nested occurrences of $\boldsymbol{K}$ which are not equivalent to any sentence without nested $\boldsymbol{K}$'s. Of course, the same holds in $\mathcal{OL}$ as well since it subsumes $\mathcal{KL}$. But what about sentences of the form $\boldsymbol{O}\alpha$? Do we obtain reducibility at least for this special case? We saw in the previous section that we do get reducibility when α has no quantifiers. With quantifiers, the answer, in short, is no, but finding an appropriate irreducible

4 This will not work in the first-order case since as in the example on page 164, a formula $\boldsymbol{O}\beta$ can be satisfied by an infinite set of epistemic states.

1. $\forall xyz[R(x, y) \wedge R(y, z) \supset R(x, z)]$
   R is transitive.
2. $\forall x \neg R(x, x)$
   R is irreflexive.
3. $\forall x[\boldsymbol{K}P(x) \supset \exists y.R(x, y) \wedge \boldsymbol{K}P(y)]$
   For every known instance of P, there is another one that is R related to it.
4. $\forall x[\boldsymbol{K}\neg P(x) \supset \exists y.R(x, y) \wedge \boldsymbol{K}\neg P(y)]$
   For every known non-instance of P, there is another one that is R related to it.
5. $\exists x \boldsymbol{K}P(x) \wedge \exists x \boldsymbol{K}\neg P(x)$
   There is at least one known instance and known non-instance of P.
6. $\exists x \neg \boldsymbol{K}P(x)$
   There is something that is not known to be an instance of P.
7. $\forall x \boldsymbol{K}P(x) \supset P(x)$.
   Every known instance of P is a P.
8. $\forall x \boldsymbol{K}\neg P(x) \supset \neg P(x)$.
   Every known non-instance of P is not a P.

Figure 10.2: A sentence unsatisfiable in finite states

sentence is not as straightforward as one might think. For example, obvious candidates like $\boldsymbol{O}[\exists x P(x) \wedge \neg \boldsymbol{K}P(x)]$ and $\boldsymbol{O}[\exists x P(x) \wedge \neg \boldsymbol{O}P(x)]$ are both equivalent to $\boldsymbol{O}\exists x P(x)$ and hence reducible (see Exercise 6).

To show that only-knowing does not reduce, we choose a sentence which is almost identical to the sentence π in Figure 6.1 on page 105, which was used to show that finitely representable epistemic states are not sufficient to capture $\mathcal{KL}$. Let ζ be the conjunction of the sentences of Figure 10.2, which differs from π only in that there are two additional conjuncts (7) and (8).

Our first task is to show that ζ can be all that is known. To this end, let Ω be the set $\{{}^{\#}1, {}^{\#}3, {}^{\#}5, \ldots\}$ and let us call a standard name *odd* if it is in Ω and *even* otherwise. Let e be the set of world states w which satisfy the following conditions:

a) w satisfies all of the following objective sentences:
$$\{P({}^{\#}1), \neg P({}^{\#}2), P({}^{\#}3), \neg P({}^{\#}4), \ldots.\};$$
b) w satisfies conjuncts (1) and (2) stating that R is transitive and irreflexive;
c) for every even n there are infinitely many even standard names m which are R-related to n, that is, for which $w \models R(n, m)$;
d) for every odd n there are infinitely many odd standard names which are R-related to n.

Lemma 10.5.1: $e \models \boldsymbol{O}\zeta$.

Proof: It is easy to see that for every $w \in e$, $e, w \models \zeta$. Given our particular choice

of R (conditions (b)–(d)), the argument is very similar to the one used to show the satisfiability of π. Note also that the conjuncts (7) and (8) are clearly satisfied because both $\boldsymbol{K}[\forall x \boldsymbol{K} P(x) \supset P(x)]$ and $\boldsymbol{K}[\forall x \boldsymbol{K} \neg P(x) \supset \neg P(x)]$ are valid sentences. Now consider an arbitrary world state w not in e. Then it violates one of the conditions (a)–(d). We will show that, in each case, one of the conjuncts of ζ is falsified by w. If w violates condition (a), then w does not satisfy an even P or w satisfies an odd P, that is, either conjunct (7) or (8) turns out false. If condition (b) is violated, then clearly either (1) or (2) is false. Now consider the case where (c) is violated. Then there is an even n and at most finitely many even $m_1, \ldots, m_k$ such that $w \models R(n, m_i)$, and for all other even m, $w \not\models R(n, m)$. We claim that there must be some $m^* \in \{m_1, \ldots, m_k\}$ such that $w \not\models R(m^*, m)$ for all even m. For assume otherwise, that is, for every m_i there is an even m_i' such that $w \models R(m_i, m_i')$. Then, by the transitivity of R, we also have $w \models R(n, m_i')$. Hence $m_i' \in \{m_1, \ldots, m_k\}$. However, this is only possible if there is a cycle, that is, w satisfies all of $\{R(m_i, m_{j_1}), R(m_i, m_{j_2}), \ldots, R(m_i, m_{j_k})\}$ and $m_{j_k} = m_i$ for some i. But then $w \models R(m_i, m_i)$, contradicting the irreflexivity of R. Given that there is an even name m^* such that for all even names m, $w \not\models R(m^*, m)$, conjunct (3) of ζ is clearly not satisfied. Similarly, the case where condition (d) is violated implies that conjunct (4) is not satisfied. Therefore, for every $w \notin e$, $e, w \not\models \zeta$ and $e \models \boldsymbol{O}\zeta$ follows. Finally, note that e is also a maximal set because any $w \notin e$ falsifies a known basic sentence, namely ζ. ∎

Lemma 10.5.2: *For any e such that $e \models \boldsymbol{O}\zeta$, both the set of known instances of P and the set of known non-instances are infinite.*

Proof: If we assume that there are only finitely many known instances of P, say, $m_1, \ldots, m_k$, then the assumption that conjunct (3) of ζ is satisfied at every $w \in e$ leads to a contradiction with the irreflexivity of R, using an argument similar to the one in the previous proof. The case of the known non-instances of P is symmetric. ∎

Lemma 10.5.3: *Let e be any epistemic state such that $e \models \boldsymbol{O}\zeta$. Then for all objective ϕ, $e \not\models \boldsymbol{O}\phi$.*

Proof: Assume otherwise, that is, suppose there is an objective ϕ such that $e \models \boldsymbol{O}\phi$. Then e is finitely represented by ϕ. By Lemma 6.5.3, either the set of known instances of P is finite or the set of known non-instances, contradicting Lemma 10.5.2. ∎

Lemma 10.5.4: *Let $e \models \boldsymbol{O}\zeta$ and let q be a 0-ary predicate symbol not occurring in ζ. Let*

$e^* = e \cap \{w \mid w \models q\}$. *Then for any* w *and any* α *which does not mention* q *and whose subformulas* $\boldsymbol{O}\phi$ *are restricted to objective* ϕ, $e, w \models \alpha$ *iff* $e^*, w \models \alpha$.

Proof: The proof is by induction on the structure of α. The lemma clearly holds for objective sentences and, by induction, for $\neg$, $\vee$, and $\exists$. Let $e \models \boldsymbol{K}\alpha$. Then for all $w \in e$, $e, w \models \alpha$ and, by induction, for all $w \in e$, $e^*, w \models \alpha$. Since $e^* \subseteq e$, $e^* \models \boldsymbol{K}\alpha$ follows. Now suppose $e \not\models \boldsymbol{K}\alpha$. Then there is a $w \in e$ such that $e, w \not\models \alpha$. By induction, $e^*, w \not\models \alpha$. If $w \in e^*$ we are done. Otherwise, by the construction of e^*, $w \models \neg q$. Since q does not appear in ζ, it is easy to see that there must be a $\overline{w} \in e$ which is exactly like w except that $\overline{w} \models q$. Then $\overline{w}$ is in e^* and $e^*, \overline{w} \not\models \alpha$ because q does not appear in α. Hence $e^* \not\models \boldsymbol{K}\alpha$. Finally, let us consider $\boldsymbol{O}\phi$. Since ϕ is objective, $e^* \models \neg\boldsymbol{O}\phi$ because e^* knows q and q does not occur in ϕ. Also, $e \models \neg\boldsymbol{O}\phi$ because of Lemma 10.5.3. Hence, $e \models \boldsymbol{O}\phi$ iff $e^* \models \boldsymbol{O}\phi$. ∎

Theorem 10.5.5: *There is no* α *without nested modal operators such that* $\boldsymbol{O}\zeta$ *is logically equivalent to* α.

Proof: Assume, to the contrary, that there is an α without nested modal operators such that $\models \boldsymbol{O}\zeta \equiv \alpha$. Let e be any maximal set of world states such that $e \models \boldsymbol{O}\zeta$ and let $e^* = e \cap \{w \mid w \models q\}$, where q is a 0-ary predicate symbol occurring nowhere in ζ or α.

First we show that e^* itself is maximal. For that it suffices to show that for any $w \notin e^*$, $e^*, w \not\models \gamma$ for some basic γ such that $e^* \models \boldsymbol{K}\gamma$. If $w \not\models q$ we are done because $e^* \models \boldsymbol{K}q$. Otherwise $w \models q$ and, since $w \notin e^*$, $w \notin e$. Since $e \models \boldsymbol{O}\zeta$ by assumption, $e, w \not\models \zeta$. Also, since ζ does not mention q, $e^*, w \not\models \zeta$ follows from Lemma 10.5.4. By the same lemma, $e^* \models \boldsymbol{K}\zeta$ and we are done.

Continuing with the main proof of the theorem, since $\models \boldsymbol{O}\zeta \equiv \alpha$ and $e \models \boldsymbol{O}\zeta$ by assumption, we have $e, w \models \alpha$ for any w. Then, since any occurence of $\boldsymbol{O}$ in α applies only to an objective formula, $e^*, w \models \alpha$ by Lemma 10.5.4. Now consider a w in e which is not in e^*. (Such w clearly exists.) Then $e, w \models \zeta$ and thus, by Lemma 10.5.4, $e^*, w \models \zeta$. Therefore, $e^* \not\models \boldsymbol{O}\zeta$, contradicting our assumption that α and $\boldsymbol{O}\zeta$ are equivalent. ∎

10.6 Generalized stability

So far, the two main results relating $\mathcal{OL}$ to Moore's autoepistemic logic, Theorem 10.3.6 and Theorem 10.3.9, have only dealt with basic sentences or sentences like $\boldsymbol{O}\alpha$, where α is basic. However, the generalization to deal with arbitrary sentences is not difficult. First

define the *generalized belief set* of e to be the set of all sentences α (basic or not) such that $e \models \boldsymbol{K}\alpha$. Then we have the following:

Theorem 10.6.1: *A set of sentences* Γ *is a generalized belief set iff* Γ *is a* generalized stable set*, that is, it satisfies the following conditions:*

1. *If* $\Gamma \models \alpha$, *then* $\alpha \in \Gamma$.[5]
2. *If* $\alpha \in \Gamma$, *then* $\boldsymbol{K}\alpha \in \Gamma$.
3. *If* $\alpha \notin \Gamma$, *then* $\neg\boldsymbol{K}\alpha \in \Gamma$.

Proof: The proof is identical to that of Theorem 10.3.6, except without the diversion via Theorem 10.3.4 to handle first-order implication. ∎

So to convert a belief set to a generalized belief set, we need only close it under implication (rather than just first-order implication).

Dealing with $\boldsymbol{O}\alpha$ in general is also straightforward:

Theorem 10.6.2: *For any* α *and any maximal* e, $e \models \boldsymbol{O}\alpha$ *iff the generalized belief set of* e *is a* generalized stable expansion *of* α, *that is, the generalized belief set* Γ *satisfies*

$$\Gamma \textit{ is the set of implications of } \{\alpha\} \cup \{\boldsymbol{K}\beta \mid \beta \in \Gamma\} \cup \{\neg\boldsymbol{K}\beta \mid \beta \notin \Gamma\}.$$

Proof: The proof is the same as that of Theorem 10.3.9, except again without the diversion through Theorem 10.3.4 to handle first-order implication, and a generalized belief set is used here. However, by Lemma 10.3.3, belief sets completely determine the generalized belief sets. ∎

This, then, is perhaps the most succinct characterization of only-knowing: α is all that is known iff every belief follows logically from α and the basic subjective facts.

10.7 Bibliographic notes

Since the next chapter addresses other forms of nonmonotonic reasoning besides AEL, we defer bibliographic notes to that chapter.

5 This closure under full implication is the only change to the definition of stability.

10.8 Where do we go from here?

As the next chapter addresses weaknesses of AEL with regards to default reasoning, we defer comments on where to go from here also to that chapter.

10.9 Exercises

1. Show that $(\boldsymbol{K}p \supset p)$ has two stable expansions.
2. Let KB $= \{Bird(tweety), Bird(chilly), \neg Fly(chilly)\}$. We assume that both *tweety* and *chilly* are standard names. Show that $(\boldsymbol{O}(\text{KB} \wedge \delta) \supset \boldsymbol{K}Fly(tweety))$ is valid, where δ is the birds-fly default.
   Hint: Use the proof of Example 10.1.2, part 3, but with $\forall x(x \neq chilly) \supset Fly(x)$ instead of $\forall x Fly(x)$.
3. Let KB $= \{\exists x Exc(x)\}$, where $Exc(x)$ stands for $Bird(x) \wedge \neg Fly(x)$ as before. Show that $\boldsymbol{O}[\text{KB} \wedge \delta] \equiv \exists y \boldsymbol{O}[\text{KB} \wedge \forall x(Exc(x) \equiv (x = y))]$ is valid, that is, there are infinitely many epistemic states compatible with only-knowing KB.
4. Let $\delta_{\boldsymbol{K}} = \forall x[\boldsymbol{K}Bird(x) \wedge \neg \boldsymbol{K}\neg Fly(x) \supset Fly(x)]$, that is, the default about flying birds only applies to known birds, as discussed on page 162. Let
   KB $= \{Bird(tweety), Bird(best\_friend(tweety)), \neg Fly(best\_friend(tweety))\}$.
   Show whether $(\boldsymbol{O}(\text{KB} \wedge \delta_{\boldsymbol{K}}) \supset \boldsymbol{K}Fly(tweety))$ is valid.
5. Show that for any propositional formula α there is an equivalent one α' where the $\boldsymbol{K}$ and the $\boldsymbol{O}$ only dominate objective formulas..
6. Show that both $\boldsymbol{O}[\exists x P(x) \wedge \boldsymbol{K}P(x)]$ and $\boldsymbol{O}[\exists x P(x) \wedge \boldsymbol{O}P(x)]$ are logically equivalent to $\boldsymbol{O}\exists x P(x)$.

11 The Logic of Defaults

In the previous chapter, we examined a form of default reasoning in terms of autoepistemic logic. We saw how to use only-knowing to characterize what should be believed given a knowledge base consisting of a set of objective facts and a set of defaults, where the defaults were represented as non-objective sentences. As usual, we said that a sentence ϕ would be believed if $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{K}\phi)$. We also saw how, in the propositional case, epistemic states correspond to what are called *stable sets* of sentences, and how the epistemic states satisfying $\boldsymbol{O}\alpha$ correspond to what are called *stable expansions* of α.

So the account of default reasoning from the previous chapter depends crucially on which epistemic states are considered to only-know α. It turns out that when the sentence α is not objective, there are some reasonable alternatives to what it means to only-know it. In this chapter, we will consider some variants. We will propose a logic with three only-knowing operators: $\boldsymbol{O}_{M}$, $\boldsymbol{O}_{K}$, and $\boldsymbol{O}_{R}$. The first of these is a relabelling of the $\boldsymbol{O}$ operator from previous chapters, while the other two will be new. The three operators agree completely on what it means to only-know an *objective* sentence, but give different results on the non-objective sentences, and therefore lead to different forms of default reasoning.

The three subscripts on the $\boldsymbol{O}$ operator are taken from the names of three researchers whose proposals inspired the definitions: Robert **M**oore, Kurt **K**onolige, and Ray **R**eiter. All three worked on default reasoning in terms of the sets of objective sentences believed given certain defaults, which in this chapter we will call *extensions* of the KB: the Moore extensions are the objective parts of stable expansions from the previous chapter; the Konolige extensions (corresponding to $\boldsymbol{O}_{K}$) and the Reiter extensions (corresponding to $\boldsymbol{O}_{R}$) will be the new ones.

It will be convenient in what follows to restrict knowledge bases to what are called *closed default theories*. So unless stated otherwise, $\text{KB} = F \cup D$, where F is an arbitrary finite set of objective sentences, and D is a finite set of closed defaults, sentences of the form $(\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi)$. Three things to notice: First, we are using $\boldsymbol{K}\phi$ instead of ϕ as the first part of the default (called the prerequisite). This already came up in the previous chapter. This formulation does the job in many cases, but does admittedly have drawbacks when it comes to drawing default conclusions from disjunctions or existentials. Second, we are using $\boldsymbol{M}\psi$ instead of $\neg\boldsymbol{K}\neg\psi$ as the second part of the default (called the justification). For now, this $\boldsymbol{M}$ can be thought as an abbreviation for $\neg\boldsymbol{K}\neg$. Finally, we are not considering defaults that have quantifiers on the outside of the belief operators, as we did in the previous chapter. These will be taken up at the end in Section 11.5.

11.1 Varieties of only-knowing

To introduce the three forms of only-knowing, we first define the three forms of extensions from the literature. The Moore extensions derive directly from stable expansions:

Definition 11.1.1: A set E of objective sentences is a *Moore extension* of a closed default theory $\langle F, D\rangle$ iff E is the objective subset of a stable expansion of $(F \cup D)$.

As noted in the previous chapter, AEL sometimes allows too many stable expansions. Consider, for example, $\gamma = (\neg \boldsymbol{K} p \vee p)$. This has two stable expansions since there are two epistemic states that satisfy $\boldsymbol{O}\gamma$: $e_0 = W$ and $e_p = \{w : w \models p\}$. It is easy to see why e_0 should satisfy $\boldsymbol{O}\gamma$. First note that $\boldsymbol{K}(\neg \boldsymbol{K} p \vee p)$ is valid in $\mathcal{OL}$ (and in $\mathcal{KL}$), so γ is believed in every epistemic state. So e_0 believes γ and clearly e_0 knows nothing else. But why should e_p satisfy $\boldsymbol{O}\gamma$? The answer goes back to Theorem 8.3.8 of Chapter 8: if all you know is something, then disjoining a false subjective sentence does not affect anything. Since $e_p \models \boldsymbol{O} p$ and $e_p \models \boldsymbol{K} p$, we have that $e_p \models \boldsymbol{O}(p \vee \neg \boldsymbol{K} p)$.

But there is another way of looking at it. We can think of only-knowing a sentence as requiring that everything that is known be recoverable from that sentence alone. But as we said, the γ above is believed in every state. So if this is all that is known, there is no reason to believe p. Consequently, e_p must know something more than just γ.

This is the basis for the definition of $\boldsymbol{O}_{\kappa}$ in the next section. In terms of the objective sentences believed, we have the following definition:

Definition 11.1.2: A set E of objective sentences is a *Konolige extension* of a closed default theory $\langle F, D\rangle$ iff E is a minimal Moore extension of $\langle F, D\rangle$.

So the γ above has two Moore extensions but only one Konolige extension.[1] What we will get in terms of only-knowing in the logic is that $e_p \models \boldsymbol{O}_{\mathcal{M}}\gamma$ but $e_p \not\models \boldsymbol{O}_{\kappa}\gamma$. More generally, we will have that $\not\models (\boldsymbol{O}_{\mathcal{M}}\gamma \supset \neg \boldsymbol{K} p)$, but $\models (\boldsymbol{O}_{\kappa}\gamma \supset \neg \boldsymbol{K} p)$.

Turning now to $\boldsymbol{O}_{\mathcal{R}}$, we first need to consider how Reiter extensions are defined:

Definition 11.1.3: Let $\langle F, D\rangle$ be a closed default theory and let S be a set of objective sentences. $\Gamma(S)$ is defined to be the least set of objective sentences such that

1. $F \subseteq \Gamma(S)$;
2. if $\Gamma(S) \models \phi$ then $\phi \in \Gamma(S)$;
3. if $(\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi) \in D$, $\phi \in \Gamma(S)$, and $\neg\psi \notin S$ then $\chi \in \Gamma(S)$.

A set E of objective sentences is a *Reiter extension* of $\langle F, D\rangle$ iff $\Gamma(E) = E$.

1 Konolige used the term "moderately grounded stable expansion."

This definition is quite different from the ones based on stable expansions, and so relating Reiter extensions to only-knowing is somewhat more involved. In particular, it will push us to treat the $\boldsymbol{M}$ operator as different from $\neg\boldsymbol{K}\neg$. To see why, consider the default theory where F is empty and D contains the following two defaults:

$$\boldsymbol{K}p \wedge \boldsymbol{M}\text{TRUE} \supset p$$
$$\boldsymbol{K}\text{TRUE} \wedge \boldsymbol{M}\neg p \supset p.$$

If $\boldsymbol{M}$ means $\neg\boldsymbol{K}\neg$, we have that $\models \delta \equiv p$ and so it will turn out that $\models \boldsymbol{O}_{\scriptscriptstyle M}\delta \equiv \boldsymbol{O}_{\scriptscriptstyle M}p$ and $\models \boldsymbol{O}_{\scriptscriptstyle K}\delta \equiv \boldsymbol{O}_{\scriptscriptstyle K}p$. So e_p satisfies both $\boldsymbol{O}_{\scriptscriptstyle M}\delta$ and $\boldsymbol{O}_{\scriptscriptstyle K}\delta$ (and is the only epistemic state to do so).

However, it is not hard to see that this default theory has no Reiter extensions. The first default does not sanction p for the same reason ($\neg\boldsymbol{K}p \vee p$) does not sanction p (as seen above); the second default is nonsensical, since it says that we can conclude p when $\neg p$ is consistent with what is believed. In Reiter's logic, defaults are not sentences,[2] and so we cannot somehow combine the $\boldsymbol{K}p$ from the first default with the $\boldsymbol{M}\neg p$ from the second to derive p. This means that we will need to give up on the duality of $\boldsymbol{K}$ and $\boldsymbol{M}$ within the context of the $\boldsymbol{O}_{\scriptscriptstyle R}$ operator.

11.2 The logic $\mathcal{O}_{\scriptscriptstyle 3}\mathcal{L}$

Putting all the pieces into place, we define a new logic $\mathcal{O}_{\scriptscriptstyle 3}\mathcal{L}$ that is similar to $\mathcal{OL}$, but with separate $\boldsymbol{K}$ and $\boldsymbol{M}$ operators, and with the $\boldsymbol{O}$ operator replaced by three operators, $\boldsymbol{O}_{\scriptscriptstyle M}$, $\boldsymbol{O}_{\scriptscriptstyle K}$, and $\boldsymbol{O}_{\scriptscriptstyle R}$. The semantics of $\mathcal{O}_{\scriptscriptstyle 3}\mathcal{L}$ is like that of $\mathcal{OL}$ except that two epistemic states are used: one to interpret $\boldsymbol{K}$ and one to interpret $\boldsymbol{M}$ (since, as we noted, there will be contexts where the two operators are not duals).

Definition 11.2.1: A sentence α is *true* wrt epistemic state e and world w, which we write as $e, w \models \alpha$, according to whether or not $e, e, w \models \alpha$, defined as follows:

1. $e_1, e_2, w \models P(t_1, \ldots, t_k)$ iff $w[P(n_1, \ldots, n_k)] = 1$, where $n_i = w(t_i)$;
2. $e_1, e_2, w \models (n_1 = n_2)$ iff $w(t_1)$ is the same name as $w(t_2)$;
3. $e_1, e_2, w \models \neg\alpha$ iff $e_1, e_2, w \not\models \alpha$;
4. $e_1, e_2, w \models (\alpha \wedge \beta)$ iff $e_1, e_2, w \models \alpha$ and $e_1, e_2, w \models \beta$;
5. $e_1, e_2, w \models \forall x.\alpha$ iff $e_1, e_2, w \models \alpha^x_n$ for every standard name n;
6. $e_1, e_2, w \models \boldsymbol{K}\alpha$ iff $e_1, e_2, w' \models \alpha$ for every $w' \in e_1$;
7. $e_1, e_2, w \models \boldsymbol{M}\alpha$ iff $e_1, e_2, w' \models \alpha$ for some $w' \in e_2$.

2 In Reiter's original formulation, there were no belief operators, and defaults were merely triples $\langle \phi, \psi, \chi \rangle$.

8. $e_1, e_2, w \models \boldsymbol{O}_M\alpha$ iff for every $w' \in W$, $e_1, e_2, w' \models \alpha$ iff $w' \in e_1$.
9. $e_1, e_2, w \models \boldsymbol{O}_K\alpha$ iff for every e' such that $e_1 \subseteq e'$, $e', e', w \models \boldsymbol{O}_M\alpha$ iff $e' = e_1$;
10. $e_1, e_2, w \models \boldsymbol{O}_R\alpha$ iff for every e' such that $e_1 \subseteq e'$, $e', e_2, w \models \boldsymbol{O}_M\alpha$ iff $e' = e_1$;

We say that a sentence α is *valid* in $\mathcal{O}_3\mathcal{L}$ (which we write as $\models \alpha$) iff $e, w \models \alpha$ for every e and w. If α is objective, we often omit the e and write $w \models \alpha$; if α is subjective, we write $e \models \alpha$ or perhaps $e_1, e_2 \models \alpha$.

Observe that when $e_1 = e_2$ in the above, $\boldsymbol{K}$ and $\boldsymbol{M}$ behave like the usual duals. In fact, the only time they do not behave like duals is within the scope of $\boldsymbol{O}_R$:

Theorem 11.2.2: *A sentence α is valid in $\mathcal{O}_3\mathcal{L}$ iff α' is valid, where α' is α with all occurrences of $\boldsymbol{M}$ outside of an $\boldsymbol{O}_R$ operator replaced by $\neg\boldsymbol{K}\neg$.*

Proof: The only place we can get $e_1 \neq e_2$ is within the $\boldsymbol{O}_R$ operator. ∎

This means that without $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$ operators, $\mathcal{O}_3\mathcal{L}$ behaves exactly like $\mathcal{OL}$:

Theorem 11.2.3: *A sentence α not mentioning $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$ is valid in $\mathcal{O}_3\mathcal{L}$ iff α' is valid in $\mathcal{OL}$, where α' is α with all occurrences of $\boldsymbol{M}$ replaced by $\neg\boldsymbol{K}\neg$ and $\boldsymbol{O}_M$ by $\boldsymbol{O}$.*

Proof: Without $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$ operators, only the first epistemic state e_1 is ever used. ∎

Note also that the definition of $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$ differ only in one small detail: where $\boldsymbol{O}_K$ uses the e' for its second epistemic argument (thus keeping the two epistemic states identical), $\boldsymbol{O}_R$ uses the given e_2.

Before turning to defaults, let us consider the relationships among these three forms of only-knowing. It is easy to see that all three coincide on objective sentences.

Theorem 11.2.4: *If ϕ is objective, then $\models \boldsymbol{O}_M\phi \equiv \boldsymbol{O}_K\phi$ and $\models \boldsymbol{O}_K\phi \equiv \boldsymbol{O}_R\phi$.*

This is so because there is a unique e such that $e \models \boldsymbol{O}_M\phi$, namely $e = \{w \mid w \models \phi\}$, and the second epistemic state is irrelevant for objective sentences.

From the definition of the $\boldsymbol{O}$-operators it also follows immediately that $\boldsymbol{O}_M$ is the most basic of the three in the following sense:

Theorem 11.2.5: *For all sentences α, $\models (\boldsymbol{O}_K\alpha \supset \boldsymbol{O}_M\alpha)$ and $\models (\boldsymbol{O}_R\alpha \supset \boldsymbol{O}_M\alpha)$.*

The converse, however, fails in both cases. For suppose that γ is $(\neg \boldsymbol{K} p \vee p)$ as before. Then $e_p \models \boldsymbol{O}_{\boldsymbol{M}} \gamma$, but $e_p \not\models \boldsymbol{O}_{\boldsymbol{K}} \gamma$ because $e_0, e_0 \models \boldsymbol{O}_{\boldsymbol{M}} \gamma$, and $e_p \not\models \boldsymbol{O}_{\boldsymbol{R}} \gamma$ because $e_0, e_p \models \boldsymbol{O}_{\boldsymbol{M}} \gamma$.

Looking at the definition of $\boldsymbol{O}_{\boldsymbol{M}}$ and $\boldsymbol{O}_{\boldsymbol{K}}$, it is not difficult to derive necessary and sufficient conditions for when the two modalities coincide:

Lemma 11.2.6: *For any epistemic state e and basic sentence α, $e \models \boldsymbol{O}_{\boldsymbol{K}} \alpha$ iff $e \models \boldsymbol{O}_{\boldsymbol{M}} \alpha$ and for all $e' \supsetneq e$, $e' \not\models \boldsymbol{O}_{\boldsymbol{M}} \alpha$.*

Theorem 11.2.7: $\models \boldsymbol{O}_{\boldsymbol{M}} \alpha \equiv \boldsymbol{O}_{\boldsymbol{K}} \alpha$ *iff for all e, if $e \models \boldsymbol{O}_{\boldsymbol{M}} \alpha$ then for all $e' \supsetneq e$, $e' \not\models \boldsymbol{O}_{\boldsymbol{M}} \alpha$.*

As a special case of this theorem we get that $\boldsymbol{O}_{\boldsymbol{M}}$ and $\boldsymbol{O}_{\boldsymbol{K}}$ agree on α if there is a *unique* epistemic state that only-knows it:

Corollary 11.2.8: *Suppose α is definite. Then $\models \boldsymbol{O}_{\boldsymbol{M}} \alpha \equiv \boldsymbol{O}_{\boldsymbol{K}} \alpha$.*

Theorem 11.2.7 fails when we use $\boldsymbol{O}_{\boldsymbol{R}}$ instead of $\boldsymbol{O}_{\boldsymbol{K}}$. Indeed, it does not even hold in the case where there is a unique e such that $e \models \boldsymbol{O}_{\boldsymbol{M}} \alpha$, as the following example demonstrates. Let δ be the two defaults from before

$$(\neg \boldsymbol{K} p \vee \neg \boldsymbol{M} \mathrm{TRUE} \vee p) \wedge (\neg \boldsymbol{K} \mathrm{TRUE} \vee \neg \boldsymbol{M} \neg p \vee p).$$

Since $\boldsymbol{O}_{\boldsymbol{M}}$ treats $\boldsymbol{K}$ and $\boldsymbol{M}$ as duals, $\boldsymbol{O}_{\boldsymbol{M}} \delta$ is logically equivalent to $\boldsymbol{O}_{\boldsymbol{M}} p$, and so e_p is the only epistemic state e such that $e \models \boldsymbol{O}_{\boldsymbol{M}} \delta$. But $\boldsymbol{O}_{\boldsymbol{R}}$ treats $\boldsymbol{K}$ and $\boldsymbol{M}$ differently and in particular, $e_p \not\models \boldsymbol{O}_{\boldsymbol{R}} \delta$ because $e_0, e_p \models \boldsymbol{O}_{\boldsymbol{M}} \delta$.

However, one interesting property is that $\boldsymbol{O}_{\boldsymbol{R}}$ does reduce to $\boldsymbol{O}_{\boldsymbol{K}}$ when $\boldsymbol{K}$ is the only modality in α, and to $\boldsymbol{O}_{\boldsymbol{M}}$ when $\boldsymbol{M}$ is the only modality:[3]

Theorem 11.2.9: *For any $\boldsymbol{K}$-basic sentence α (that is, any sentence where $\boldsymbol{K}$ is the only modality), $\models \boldsymbol{O}_{\boldsymbol{R}} \alpha \equiv \boldsymbol{O}_{\boldsymbol{K}} \alpha$, and for any $\boldsymbol{M}$-basic sentence α (that is, any sentence where $\boldsymbol{M}$ is the only modality), $\models \boldsymbol{O}_{\boldsymbol{R}} \alpha \equiv \boldsymbol{O}_{\boldsymbol{M}} \alpha$.*

Proof: For the first part, recall that $\boldsymbol{O}_{\boldsymbol{K}}$ and $\boldsymbol{O}_{\boldsymbol{R}}$ differ only in one place, which concerns the interpretation of $\boldsymbol{M}$. A simple induction on α shows that for any e_1, e_2, e_3, w, and a $\boldsymbol{K}$-basic α, $e_1, e_2, w \models \alpha$ iff $e_1, e_3, w \models \alpha$. The equivalence of $\boldsymbol{O}_{\boldsymbol{R}} \alpha$ and $\boldsymbol{O}_{\boldsymbol{K}} \alpha$ then follows immediately from the definitions of the two operators. Turning now to the second part, the only-if direction is immediate because of Theorem 11.2.5. For the converse, suppose $e \models \boldsymbol{O}_{\boldsymbol{M}} \alpha$. Then it suffices to show that for all $e' \supsetneq e$, $e', e \not\models \boldsymbol{O}_{\boldsymbol{M}} \alpha$. As α does not mention

3 It is a happy coincidence that the first letters of the two names align with the two basic modalities.

$\boldsymbol{K}$, a simple induction shows that for any e_1, e_2, e_3, w, $e_1, e_2, w \models \alpha$ iff $e_3, e_2, w \models \alpha$. Now let $w \in e' - e$. By assumption, $e, e, w \not\models \alpha$. Therefore, $e', e, w \not\models \alpha$, from which $e', e \not\models \boldsymbol{O}_M\alpha$ follows. ∎

As a final general property of $\mathcal{O}_3\mathcal{L}$, recall Corollary 10.4.6 of the previous chapter. It says that for any propositional α, there is a finite set of objective sentences Φ_M such that

$$\models \boldsymbol{O}_M\alpha \equiv \bigvee\{\boldsymbol{O}_M\phi \mid \phi \in \Phi_M\}.$$

A similar argument can be used to show analogous properties for $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$:

Theorem 11.2.10: *For any propositional α, there are finite sets of objective sentences Φ_K and Φ_R such that*

1. $\models \boldsymbol{O}_K\alpha \equiv \bigvee\{\boldsymbol{O}_K\phi \mid \phi \in \Phi_K\}$;
2. $\models \boldsymbol{O}_R\alpha \equiv \bigvee\{\boldsymbol{O}_R\phi \mid \phi \in \Phi_R\}$.

(The proof is left as an exercise.) Using Theorem 11.2.5, it also follows that $\Phi_K \subseteq \Phi_M$ and $\Phi_R \subseteq \Phi_M$. This means that, in the propositional case, only-knowing an arbitrary basic sentence α is reducible to a sentence without nested modalities, not only for $\boldsymbol{O}_M$, but also for $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$. None of these reductions hold in the full first-order case, however, as a consequence of Theorem 10.5.5 of the previous chapter.

11.3 Handling closed defaults

The logic $\mathcal{O}_3\mathcal{L}$ defined above is quite general, and is not restricted to knowledge bases that are closed default theories. What we will show in this section, however, is that under this restriction, there is a one-to-one correspondence between the three forms of extensions defined above and the three forms of only-knowing in $\mathcal{O}_3\mathcal{L}$.

As we will see, this is fairly straightforward to prove in the case of Moore and Konolige extensions, but not so for Reiter. To prepare for that, we first need to look at Reiter extensions more closely (see Definition 11.1.3). We begin with an alternate characterization due to Reiter that avoids the use of the operator Γ:

Theorem 11.3.1: *[Reiter] Let E be a set of objective sentences and $\langle F, D\rangle$ be a closed default theory. Let*

1. $E_0 = F$;
2. $E_{i+1} = \{\chi \mid E_i \models \chi\} \cup \{\chi \mid (\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi) \in D, \phi \in E_i, \text{ and } \neg\psi \notin E\}$.

Then E is a Reiter extension of $\langle F, D\rangle$ iff $E = \bigcup_{i=0}^{\infty} E_i$.

Note that the theorem is not a recipe to construct extensions as E itself is mentioned in the definition of E_{i+1}. We remark that for finite default theories, $E = E_k$ for some $k \geq 0$.

Next we define a semantic version of Reiter extensions in terms of epistemic states:

Definition 11.3.2: Let $\langle F, D\rangle$ be a closed default theory and let e be any set of worlds. $\Delta(e)$ is defined to be the largest set of worlds such that

1. $\Delta(e) \subseteq \Re[\![F]\!]$
2. for all $\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D$, if $\Delta(e) \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$ then $\Delta(e) \models \boldsymbol{K}\chi$.

e ia called an e-extension of $\langle F, D\rangle$ if $e = \Delta(e)$.

Eventually we will use this definition to prove the exact correspondence between Reiter extensions and $\boldsymbol{O}_k$, but we first need to establish that e-extensions and Reiter extensions indeed amount to the same thing. We begin by establishing a semantic version of Theorem 11.3.1 for e-extensions:

Lemma 11.3.3: *Let e be a set worlds and $\langle F, D\rangle$ be a closed default theory. Let*

1. $e^0 = \Re[\![F]\!]$;
2. $e^{i+1} = e^i \cap \{w \mid \text{for all } \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D, \text{ if } e^i \models \boldsymbol{K}\phi \text{ and } e \models \boldsymbol{M}\psi \text{ then } w \models \chi\}$.

*Then e is an e-*extension *of $\langle F, D\rangle$ iff $e = \bigcap_{i=0}^{\infty} e^i$.*

Proof: Let $e^* = \bigcap_{i=0}^{\infty} e^i$. Note that $e^* \subseteq \Re[\![F]\!]$ and for $\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D$, if $e^* \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$ then $e^* \models \boldsymbol{K}\chi$. By the maximality of of $\Delta(e)$ it follows that $e^* \subseteq \Delta(e)$.

To prove the only-if direction, let $e = \Delta(e)$. We show that $e \subseteq e^i$ for all i, from which $e \subseteq e^*$ follows. The proof is by induction on i. Clearly, $e \subseteq e^0$ by the definition of $\Delta(e)$. Now suppose $e \subseteq e^i$ and let $w \in e$. We need to show that $w \in e^{i+1}$. Since $w \in e^i$ by induction, it suffices to show that for any $\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D$, if $e^i \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$ then $w \models \chi$. Let $e^i \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$. Since $e \subseteq e^i$ by assumption, we have $e \models \boldsymbol{K}\phi$. Since $e = \Delta(e)$ we thus have $e \models \boldsymbol{K}\chi$ and, therefore, $w \models \chi$.

Conversely, let $e = e^*$. We show that $\Delta(e) \subseteq e^i$ for all i, from which $\Delta(e) \subseteq e^*(= e)$ follows. Since we already have that $e^* \subseteq \Delta(e)$, we are done. The proof is by induction on i. Clearly, $\Delta(e) \subseteq e^0$. Suppose $\Delta(e) \subseteq e^i$ and let $w \in \Delta(e)$. We need to show that $w \in e^{i+1}$. By induction $w \in e^i$. It suffices to show for any $\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D$, if $e^i \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$ then $w \models \chi$. Let $e^i \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$. Since $\Delta(e) \subseteq e^i$ by assumption, $\Delta(e) \models \boldsymbol{K}\phi$. By the definition of $\Delta(e)$, we then have $\Delta(e) \models \boldsymbol{K}\chi$. Since $w \in \Delta(e)$, $w \models \chi$ follows. ∎

Note that for a given default theory $\langle F, D\rangle$, the E_i and e^i do not line up exactly, that is, $\Re[\![E_i]\!]$ may not be the same as e^i. A simple example is $F = \{p \wedge q\}$ and $D = \{\boldsymbol{K}p \wedge \boldsymbol{M}r \supset r\}$. While $\Re[\![E_0]\!] = e^0$, $\Re[\![E_1]\!] = \{w \mid w \models p \wedge q\} \neq e^1 = \{w \mid w \models p \wedge q \wedge r\}$. This is because the beliefs of e^0 are closed under logical entailment while E_0 is not. However, closure unter entailment is always achieved at higher levels in Reiter's case. The following two lemmas make this precise.

Lemma 11.3.4: *Let $\langle F, D\rangle$ be a default theory and E a set of objective sentences such that $E = \{\phi \mid E \models \phi\}$ and $e = \Re[\![E]\!]$. Then $e^i \subseteq \Re[\![E_i]\!]$ for all i.*

Proof: The proof is by induction on i. Clearly $e^0 \subseteq \Re[\![E_0]\!]$. Suppose $e^i \subseteq \Re[\![E_i]\!]$. We need to show that $e^{i+1} \subseteq \Re[\![E_{i+1}]\!]$. Let $w \in e^{i+1}$. Recall that $E_{i+1} = \{\phi \mid E_i \models \phi\} \cup \{\chi \mid \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D, \phi \in E_i, \text{ and } \psi \notin E\}$. Since, by assumption, $e^i \subseteq \Re[\![E_i]\!]$ and $w \in e^i$, $w \models \{\phi \mid E_i \models \phi\}$. Suppose $\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D$ such that $\phi \in E_i$ and $\neg\psi \notin E$. Then $e^i \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$. Hence, by the definition of e^{i+1}, $w \models \chi$, from which $w \in \Re[\![E_{i+1}]\!]$ follows. ∎

While the converse of the lemma does not hold, we do have the following:

Lemma 11.3.5: *Let $\langle F, D\rangle$, E, and e be as in the lemma above.*
Then $\Re[\![E_{i+2}]\!] \subseteq e^i$ for all i.

Proof: The proof is by induction on i. Since $e^0 = \Re[\![F]\!]$ and $E_2 \supseteq \{\phi \mid F \models \phi\}$, $\Re[\![E_2]\!] \subseteq e^0$. Suppose $\Re[\![E_{i+2}]\!] \subseteq e^i$ and let $w \in \Re[\![E_{i+3}]\!]$. We need to show that $w \in e^{i+1}$. By the definition of E_{i+3}, $w \models \{\phi \mid E_{i+2} \models \phi\}$. Thus $w \in \Re[\![E_{i+2}]\!]$ and since $\Re[\![E_{i+2}]\!] \subseteq e^i$ by assumption, $w \in e^i$. We are left to show that $w \in e^* = \{w \mid \text{for all } \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D, \text{ if } e^i \models \boldsymbol{K}\phi \text{ and } e \models \boldsymbol{M}\psi \text{ then } w \models \chi\}$. So suppose $e^i \models \boldsymbol{K}\phi$ and $e \models \boldsymbol{M}\psi$ for $\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi \in D$. We need to show that $w \models \chi$. Since $\Re[\![E_{i+2}]\!] \subseteq e^i$ by assumption, $\Re[\![E_{i+2}]\!] \models \boldsymbol{K}\phi$, that is, $\phi \in \{\phi \mid E_{i+2} \models \phi\}$, and since $\{\phi \mid E_{i+2} \models \phi\} \subseteq E_{i+3}$, $\phi \in E_{i+3}$. Since $\{\phi \mid E \models \phi\} = E$ by assumption and $e \models \boldsymbol{M}\psi$, $\neg\psi \notin E$. Thus $\chi \in E_{i+3}$, from which $w \models \chi$ follows. ∎

With the previous two lemmas in hand, we can now proceed to showing that e-extensions are indeed a correct semantic account of Reiter extensions.

Lemma 11.3.6: *Let $\langle F, D\rangle$ be a finite closed default theory, $E = \{\phi \mid E \models \phi\}$ and $e = \Re[\![E]\!]$. Then $E = \Gamma(E)$ iff $e = \Delta(e)$.*

Proof: To prove the only-if direction, let $E = \Gamma(E)$. By Theorem 11.3.1, $E = \bigcup_{i=0}^{\infty} E_i$. Since the default theory is finite, there is a k such that for all $j \geq k$, $E = E_j$. Now consider $e^* = \bigcap_{i=0}^{\infty} e^i$. Again, for some l and all $j \geq l$, $e^* = e^j$. Wlog. let $k \geq l$. Then we have $e = \Re[\![E]\!] = \Re[\![E_k]\!] = \Re[\![E_{k+2}]\!] = e^{k+2}$ by Lemma 11.3.4 and 11.3.5. Hence $e^* = e$ and, by Theorem 11.3.3, $e = \Delta(e)$.

Conversely, let $e = \Delta(e)$. By Theorem 11.3.3, $e = \bigcap_{i=0}^{\infty} e^i$. Now consider $E^* = \bigcup_{i=0}^{\infty} E_i$. As before, for some l, $e = e^l$ and for some k, $E^* = E_k$ with $k \geq l$. By Lemma 11.3.4 and 11.3.5, $\Re[\![E_{k+2}]\!] = e^{k+2}$. Since $e = e^{k+2}$, $\Re[\![E_{k+2}]\!] = \Re[\![E]\!]$. Since $\{\phi \mid E^* \models \phi\} = E^* = \{\phi \mid E_{k+2} \models \phi\}$ and $\{\phi \mid E \models \phi\} = E$, $E^* = E$. Hence $E = \Gamma(E)$ follows by Theorem 11.3.1. ∎

The main theorem of this chapter is the close correpondence between only-knowing as we have defined it and the three forms of extensions we have defined:

Theorem 11.3.7: *Let $\alpha = \bigwedge(F \cup D)$, where $\langle F, D\rangle$ is a closed default theory. Then*

1. *E is a Moore extension of $\langle F, D\rangle$ iff*
   there is an e such that $e \models \boldsymbol{O}_M\alpha$ and E is the objective belief set of e;
2. *E is a Konolige extension of $\langle F, D\rangle$ iff*
   there is an e such that $e \models \boldsymbol{O}_K\alpha$ and E is the objective belief set of e;
3. *E is a Reiter extension of $\langle F, D\rangle$ iff*
   there is an e such that $e \models \boldsymbol{O}_R\alpha$ and E is the objective belief set of e;

Proof:

1. Since Moore extensions uniquely determine stable expansions in the case of closed default theories, Part 1 of the theorem follows immediately from Theorem 10.3.9 together with Theorem 11.2.2.
2. Let E be a Konolige extension. Then E is a minimal Moore extension. By Part 1, for some e, $e \models \boldsymbol{O}_M\alpha$ and E is the objective belief set of e. Since E is minimal, there cannot be an epistemic state e' such that $e \subsetneq e'$ and $e' \models \boldsymbol{O}_M\alpha$.
   Conversely, if $e \models \boldsymbol{O}_K\alpha$, then the objective belief set of e is a Moore extension by Theorem 11.2.5 and Part 1. In addition, it is clearly minimal as there is no proper superset e' of e such that $e' \models \boldsymbol{O}_M\alpha$.
3. For the only-if direction, let E be a Reiter extension of α. It suffices to show that for $e = \Re[\![E]\!]$, $e \models \boldsymbol{O}_R\alpha$, that is, $e \models \boldsymbol{O}_M\alpha$ and for all $e' \supsetneq e$, $e', e \not\models \boldsymbol{O}_M\alpha$. We first prove that $e \models \boldsymbol{O}_M\alpha$. By Theorem 11.3.6, $e = \Delta(e)$. Given the definition of Δ, clearly $e \models \boldsymbol{K}\alpha$. Now let $w \notin e$. Since e is a fixpoint of Δ, $e \cup \{w\} \not\models \boldsymbol{K}(\bigwedge F)$ or for some default, $e \cup \{w\}, e \not\models \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \boldsymbol{K}\chi$. Since $F, \phi, \psi.\chi$ are objective,

it follows that either $w \not\models F$ or $e, w \not\models \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi$. Hence $e, w \not\models \alpha$. Now let $e' \supsetneq e$ and suppose $e', e \models \boldsymbol{O}_{\!M}\alpha$. Then clearly $e' \subseteq \Re[\![F]\!]$. Also for any default, $e', e \models \boldsymbol{K}(\boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi)$ and thus $e', e \models \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \boldsymbol{K}\chi)$. Since, by definition, $\Delta(e)$ is maximal, we then have $e' \subseteq \Delta(e)$, contradicting the assumption that $e = \Delta(e)$. Conversely, let $e \models \boldsymbol{O}_{\!R}\alpha$ and let E be the objective belief set of e. By the definition of $\boldsymbol{O}_{\!R}$ we have that for all $e' \supsetneq e$, $e', e \not\models \boldsymbol{O}_{\!M}\alpha$. To show that E is a Reiter extension, it suffices to show that $\Delta(e) = e$ by Theorem 11.3.6. Suppose, to the contrary, that there is an $e' \supsetneq e$ such that $e', e \models \boldsymbol{K}(\bigwedge F)$ and for all defaults, $e', e \models \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \boldsymbol{K}\chi$. Let e' be the largest superset of e with that property. Then $e', e \models \boldsymbol{K}\alpha$. Also, for any $w \notin e'$, either $w \not\models F$ or $e', e, w \not\models \boldsymbol{K}\phi \wedge \boldsymbol{M}\psi \supset \chi$ for some default. But then $e', e \models \boldsymbol{O}_{\!M}\alpha$, contradiction. ∎

Corollary 11.3.8: *Let* KB *be the encoding of a finite closed default theory and* ψ *be an objective sentence. Then* ψ *is an element of every Moore/Konolige/Reiter extension of* KB *iff* $\boldsymbol{K}\psi$ *is logically entailed by* $\boldsymbol{O}_{\!M}$KB / $\boldsymbol{O}_{\!K}$KB / $\boldsymbol{O}_{\!R}$KB.

This shows that models of only-knowing are in 1–1 correspondence with the syntactically defined extensions of default theories, for Moore, Konolige, and Reiter. Many properties of closed default theories now follow directly from general properties of belief. For example, the fact that every Reiter extension is also a Moore extension follows from our Theorem 11.2.5. The fact that the converse does not hold in general, but does hold in the case of prerequisite-free default theories (that is, ones where the defaults are of the form $\boldsymbol{K}$TRUE $\wedge \boldsymbol{M}\psi \supset \chi$) follows from Theorem 11.2.9.

What we have, in other words, is a single logic with a well-defined notion of truth where it is possible to compare what is believed in the presence of defaults according to Moore, Konolige, and Reiter. Unfortunately, this does not tell us which treatment is the best one, or indeed if any of them are any good. Rather, it appears that each of the three proposals has faults and limitations. (For Reiter's account, perhaps the only problem is its handling of open defaults, discussed in Section 11.5.)

11.4 An axiomatic account

Given that $\mathcal{O}_3\mathcal{L}$ is a classical truth-theoretic logic, we can consider looking for a set of axioms and rules of inference that will generate all and only the valid sentences of the logic. These proof-theoretic characterizations sometimes provide additional insight into the behaviour of the logic.

An axiom system will also allow us to consider step-by-step monotonic *derivations* for

skeptical default reasoning. Instead of starting with a KB made up of facts and defaults and looking for some nonmonotonic operations that will lead to a conclusion like *Fly*(*tweety*), we can start with $\boldsymbol{O}$KB as a given assumption, and then look for a sequence of classical monotonic steps that will allow us to conclude $\boldsymbol{K}$*Fly*(*tweety*).

As in previous sections, we will only consider closed default theories. Specifically, we will develop an axiomatic proof theory for $\mathcal{O}_3\mathcal{L}$ under the following restrictions:

1. we consider the propositional subset of the language only;
2. we exclude $\boldsymbol{O}$ operators within the scope of a $\boldsymbol{K}$, $\boldsymbol{M}$, or $\boldsymbol{O}$ operator;
3. we exclude $\boldsymbol{K}$ and $\boldsymbol{M}$ operators within the scope of a $\boldsymbol{K}$ or $\boldsymbol{M}$ operator.

(We will comment on the first of these restrictions in Section 11.5).

The axiomatization of $\mathcal{O}_3\mathcal{L}$ below builds on the one for $\mathcal{OL}$ of Chapter 9. All we will really need are some new axioms and rules of inference to handle $\boldsymbol{M}$, $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$.

11.4.1 Consistency of belief

It easy to characterize the $\boldsymbol{M}$ operator, since we can reduce it to the $\boldsymbol{K}$ operator:

Axiom:

$\boldsymbol{M}\alpha \equiv \neg\boldsymbol{K}\neg\alpha$.

11.4.2 Konolige

To characterize $\boldsymbol{O}_K$ we can make use of Theorem 11.2.4 and Theorem 11.2.10 to relate $\boldsymbol{O}_K$ to $\boldsymbol{O}_M$. For any propositional basic α, $\boldsymbol{O}_K\alpha$ is equivalent to a disjunction of $\boldsymbol{O}_M\phi_i$ sentences, where the ϕ_i are the ones that are "minimal." This can be captured as follows:

Axiom:

$(\boldsymbol{O}_K\alpha \supset \boldsymbol{O}_M\alpha)$

Rules of Inference:

From $(\boldsymbol{O}_M\psi \supset \boldsymbol{O}_M\alpha)$, $(\boldsymbol{O}_M\phi \supset \boldsymbol{O}_M\alpha)$, $(\boldsymbol{O}_M\psi \supset \boldsymbol{K}\phi)$, and $(\boldsymbol{O}_M\phi \supset \neg\boldsymbol{K}\psi)$, infer $(\boldsymbol{K}\psi \supset \neg\boldsymbol{O}_K\alpha)$.

From $(\boldsymbol{O}_M\alpha \supset \boldsymbol{O}_M\psi \vee \bigvee \boldsymbol{O}_M\phi_i)$, $(\boldsymbol{O}_M\psi \supset \boldsymbol{O}_M\alpha)$, and $(\boldsymbol{O}_M\psi \supset \bigwedge \neg\boldsymbol{K}\phi_i)$, infer $(\boldsymbol{O}_M\psi \supset \boldsymbol{O}_K\alpha)$.

The first rule of inference deals with the case where the ψ is not minimal (there is a competing ϕ that is weaker: only-knowing ϕ does not require knowing ψ) and so knowing it precludes $\boldsymbol{O}_K\alpha$; the second rule deals with the case where the ψ is one of the minimal disjuncts (only-knowing ψ precludes knowing any of the competitors), where the conclusion is the opposite.

11.4.3 Reiter

For the $\boldsymbol{O}_R$ operator, we first show how we can eliminate $\boldsymbol{M}$ operators, and then use Theorem 11.2.9 to reduce $\boldsymbol{O}_R$ to $\boldsymbol{O}_K$. To eliminate $\boldsymbol{M}$ operators, we use the following:

Theorem 11.4.1: *Let α be any sentence of $\mathcal{O}_3\mathcal{L}$ subject to the restrictions noted. For any epistemic state e, let α' be like α but with a subformula $\boldsymbol{M}\phi$ replaced by TRUE or FALSE according to whether or not $e \models \boldsymbol{M}\phi$. Then $e \models \boldsymbol{O}_R\alpha$ iff $e \models \boldsymbol{O}_R\alpha'$.*

Proof: We can show by induction on α that $e', e \models \alpha$ iff $e', e \models \alpha'$ for any epistemic state e'. Consequently, $e \models \boldsymbol{O}_R\alpha$ iff $e \models \boldsymbol{O}_R\alpha'$. ∎

What this says is that for Reiter's default logic, we can systematically replace every $\boldsymbol{M}\phi$ in α by either TRUE or by FALSE and then use the Konolige version of only-knowing. We can duplicate this reduction with the following axioms:

Axioms:

$(\boldsymbol{O}_R\alpha \equiv \boldsymbol{O}_K\alpha)$, when α has no $\boldsymbol{M}$ operators.

$\boldsymbol{M}\phi \supset (\boldsymbol{O}_R\alpha \equiv \boldsymbol{O}_R\alpha')$, where α' is α with $\boldsymbol{M}\phi$ replaced by TRUE.

$\neg\boldsymbol{M}\phi \supset (\boldsymbol{O}_R\alpha \equiv \boldsymbol{O}_R\alpha')$, where α' is α with $\boldsymbol{M}\phi$ replaced by FALSE.

This then completes the proof theory for $\mathcal{O}_3\mathcal{L}$. Putting all the pieces together, we get the following:

Theorem 11.4.2: *Let α be any sentence of $\mathcal{O}_3\mathcal{L}$ subject to the restrictions noted. Then α is valid iff it is derivable according to all the above axioms and rules of inference.*

To give a simple example of how this proof theory can be used, consider how to derive that Tweety flies in Reiter's default logic. In this case, we want to derive

$$\boldsymbol{O}_R\alpha \supset \boldsymbol{K}Fly(tweety)$$

where α is $(Bird(tweety) \wedge [\boldsymbol{K}Bird(tweety) \wedge \boldsymbol{M}Fly(tweety) \supset Fly(tweety)])$. (Note the use of the closed default.) This can be done in two parts: first derive

$$\boldsymbol{O}_R\alpha \wedge \boldsymbol{M}Fly(tweety) \supset \boldsymbol{K}Fly(tweety),$$

then derive

$$\boldsymbol{O}_R\alpha \wedge \neg\boldsymbol{M}Fly(tweety) \supset \boldsymbol{M}Fly(tweety),$$

and finally use propositional logic to combine the two parts to get the desired conclusion.

We sketch the two parts and leave the details as an exercise. For the first part, the derivation is as follows: we start with the antecedent and then (using the axioms for $\boldsymbol{O}_R$) get $\boldsymbol{O}_K\alpha'$ where α' is $(\mathit{Bird}(\mathit{tweety}) \wedge [\boldsymbol{K}\mathit{Bird}(\mathit{tweety}) \supset \mathit{Fly}(\mathit{tweety})])$, then (using the axiom for $\boldsymbol{O}_K$) get $\boldsymbol{O}_M\alpha'$, and then finally (using the properties of $\boldsymbol{O}_M$ as in the previous chapter) get the consequent $\boldsymbol{K}\mathit{Fly}(\mathit{tweety})$. For the second part, the derivation is as follows: we start with the antecedent and then (using the axioms for $\boldsymbol{O}_R$) get $\boldsymbol{O}_K\mathit{Bird}(\mathit{tweety})$, then (using the axiom for $\boldsymbol{O}_K$) get $\boldsymbol{O}_M\mathit{Bird}(\mathit{tweety})$, then (using the properties of $\boldsymbol{O}_M$ and the axiom for $\boldsymbol{M}$) get the consequent $\boldsymbol{M}\mathit{Fly}(\mathit{tweety})$.

11.5 The first-order case

While the proof theory above only works for the propositional subset of the language, and therefore for closed defaults only, the semantic theory allows quantified defaults in the case of both $\boldsymbol{O}_M$ and $\boldsymbol{O}_K$. For example, we can show that $\models (\boldsymbol{O}_K\alpha \supset \boldsymbol{K}\mathit{Fly}(^{\#}1))$ where α is

$$\mathit{Bird}(^{\#}1) \ \wedge \ \forall x.\boldsymbol{K}\mathit{Bird}(x) \wedge \boldsymbol{M}\mathit{Fly}(x) \supset \mathit{Fly}(x)$$

and similarly with $\boldsymbol{O}_R$. (See the exercises.) However, the way these quantified defaults are handled is not the same way Reiter suggests handling open defaults. The quantified default here would not work for an arbitrary constant like *tweety*, since the default is restricted to *known birds* and the identity of the constant need not be known. But for Reiter, an open default is understood as standing for all its ground instances. This leads to the desired conclusion for Tweety, but it has some curious anomalies.

For example, let a, b, c be constants and f a function symbol. If the only mention of a and b in a KB is the sentence $(\mathit{Bird}(a) \vee \mathit{Bird}(b))$, then Reiter's logic would not sanction deriving $(\mathit{Fly}(a) \vee \mathit{Fly}(b))$, since neither a nor b are known to be birds. However, if the KB also contains $(f(c) = a \vee f(c) = b)$, then Reiter's treatment of open defaults would sanction $(\mathit{Fly}(a) \vee \mathit{Fly}(b))$. So just the fact of giving the unknown bird a name (which need not even be a constant) is enough to cause the system to believe in its flying ability.

But whatever the advantages or disdavantages of our way of handling quantified defaults, it is unlikely that there are axioms and rules of inference that would work for them. Consider the following example. Let D consist of a single quantified default of the simplest sort, normal and prerequisite-free: $\forall x.\, \boldsymbol{M}\neg\mathit{Ab}(x) \supset \neg\mathit{Ab}(x)$. Let F consist of the following objective facts:

$$\begin{array}{l}
\forall x.\, R(x, x) \\
\forall x, y, z.\, R(x, y) \wedge R(y, z) \supset R(x, z) \\
\exists x.\, \mathit{Ab}(x) \\
\forall x.\, \mathit{Ab}(x) \supset \exists y.\, R(x, y) \wedge \mathit{Ab}(y)
\end{array}$$

This default theory has no Moore extensions: $\models \neg \boldsymbol{O}_M(F \wedge D)$. Consequently, it has no Konolige or Reiter extensions either. Intuitively, what is happening here is that D is insisting that the extension of Ab be minimal, whereas F is insisting that it be infinite (using an irreflexive, transitive relation R). No belief state e can satisfy both. It is unlikely that there are axioms and rules of inference that would lead to this conclusion, however, as they would need to confirm the impossibility of a minimal infinite set. It is interesting to note that in Reiter's treatment of open defaults, this theory would have a single extension: a theory that insists that there are infinitely many Ab individuals, but that contains $\neg Ab(t)$ for every term t.

11.6 Bibliographic notes

Moore's autoepistemic logic appeared in [148, 147]. The idea of stable sets goes back to Stalnaker and appeared in a note already in 1980, but was published only much later in [180]. There have been a number of proposals to syntactically characterize the stable expansions of a given set of assumptions, for example [176, 178, 136, 151, 185]. Ours, which we presented in the previous chapter, differs perhaps in that it yields a rather simple algorithm (Figure 10.1). Corollary 10.4.6 was independently obtained by Waaler [185] (see also [174]).

Konolige extensions were first introduced in [77]. That paper was the first in a line of work, including [126, 150, 135, 183], that investigated the connections between AEL and Reiter's default logic [159] by translating Reiter defaults into AEL. They all had in common that they required some modification of AEL in order to arrive at an exact correspondence. Gottlob [55] later established that a faithful translation from default logic into standard AEL is possible, but that it cannot be done in a modular way, that is, by translating every default separately.

Moore's AEL was inspired by earlier work on nonmonotonic modal logic by McDermott and Doyle [142, 143], which was later thoroughly investigated by Marek et al. [137]. Nonmonotonic modal logic generalizes AEL essentially by also considering modal logics other than K45 as the base logic. In [101] we showed that it is possible to reconstruct the different versions of nonmonotonic modal logic in terms of only-knowing with a possible-world semantics tailored to the requirements of the respective base modal logics. We did not include this flavour of only-knowing in this chapter, as it is technically quite challenging and would have distracted too much from the main points we have tried to convey.

It is perhaps interesting to note that much of the work on nonmonotonic reasoning was carried out in a propositional setting, or at least did not consider quantifying-in. In fact, as far as we know only Konolige [78, 79] and Lifschitz [123] address nonmonotonic

reasoning with quantifying-in apart from us. While there clearly are similarities, there are also significant differences, in particular, regarding the use of names. For example, while Lifschitz requires there to be a name for every individual, the name need not be unique. Moreover, there is no restriction on the cardinality of the domain. Konolige even allows for individuals which have no name at all. While certainly interesting, a detailed analysis of what these differences amount to remains largely open.

11.7 Where do we go from here?

What we have attempted to show is that it is possible to consider default reasoning from the standpoint of truth. We can look at a model of a default theory (a belief state e and a world state w) and ask what is true, what is believed, what is all that is believed. Default reasoning, in other words, does not need to be limited to a proof-theoretic analysis.

The exercise also reveals interesting connections among the versions of default reasoning proposed by Moore, Konolige, and Reiter. By formulating these three accounts within a monotonic logic of belief, we also get sentence-by-sentence derivations that correspond precisely to each form of default reasoning. The whole machinery of fixpoints, stable sets, and so on is still there in the background, of course, but we are no longer forced to use it.

But many questions remain. The rules of inference for multiple Konolige extensions are quite clumsy and should be reformulated. The use of quantified defaults and its connection to open defaults needs to be further investigated. Other forms of default reasoning, based on circumscription and nomonotonic modal systems, for instance, should be incorporated into some sort of grand unified theory.

11.8 Exercises

The exercises concerning derivations below are mainly questions about using the $\boldsymbol{O}_K$ and $\boldsymbol{O}_R$ operators. Consequently, any valid sentence that does not mention these two operators can be assumed to be derivable without any further justification.

1. Prove that $\models \boldsymbol{O}_K(Bird(^{\#}1) \wedge \forall x.\boldsymbol{K}Bird(x) \wedge \boldsymbol{M}Fly(x) \supset Fly(x)) \supset \boldsymbol{K}Fly(^{\#}1)$.
2. Prove that $\models \boldsymbol{O}_R(Bird(^{\#}1) \wedge \forall x.\boldsymbol{K}Bird(x) \wedge \boldsymbol{M}Fly(x) \supset Fly(x)) \supset \boldsymbol{K}Fly(^{\#}1)$.
3. Prove that the following is derivable: $\boldsymbol{O}_K(\neg\boldsymbol{K}p \vee p) \supset \neg\boldsymbol{K}p$.
4. Prove that the following is derivable: $\neg\boldsymbol{O}_R(\boldsymbol{M}p \supset \neg p)$.
5. Complete the derivation sketched in the text that Tweety flies according to Reiter's default logic with closed defaults.
6. Prove Theorem 11.2.10.

12 Tractable Representations

12.1 Introduction

So far in our considerations, the "symbol level" and symbolic representations of knowledge have played a relatively minor role. In as much as we cared at all about representations, we focused on results like the Representation Theorem of Chapter 7, which deals with the mere existence of finite representations, but more or less neglects the computational cost of manipulating them. There was one exception, however: we were glad to see that **ASK** and **TELL** could be characterized completely using first-order reasoning alone, even though the interaction language is modal. While we did not say so explicitly, one motivation for avoiding modal reasoning is that it is less well understood than non-modal reasoning. However, this is only partly good news, since it is well known that first-order reasoning itself is already undecidable. In particular, any faithful implementation of **ASK** and **TELL** is bound to run forever on certain inputs. While this may be acceptable under certain circumstances such as proving mathematical theorems, it clearly is not when it comes to things like commonsense reasoning, the main motivation behind our work.

If we look at the problem in terms of the properties of belief, then one reason for the computational difficulty is that beliefs are required to be closed under logical implication: what is believed is what follows logically from the knowledge base. Even when a knowledge base is empty, the agent must still believe every valid sentence of the logic. As already mentioned in Chapter 4, this assumption is known as *logical omniscience*, which seems clearly unacceptable for real resource-bounded agents.

Overall, there are two major ways of dealing with this intractability:

1. restrict the kinds of knowledge that can be represented, or
2. change the kinds of reasoning that needs to take place.

In this chapter, we consider the first alternative, limiting the expressiveness of the representation language used for knowledge bases or queries or both. We will see that if we limit the sorts of knowledge in a knowledge base to certain fragments of $\mathcal{L}$, calculating what is logically entailed can then be tractable or at least decidable. In the next chapter, we will consider the second option above, where we consider the reasoning needed for a less omniscient and less computationally demanding model of belief.

In the following, when we refer to reasoning or query answering we mean it in the sense of determining whether or not KB $\models \phi$, (or equivalently, whether or not the sentence (KB $\supset \phi$) is valid), where KB and ϕ are expressed in fragments of $\mathcal{L}$. As before KB should be thought of as the knowledge base (a finite set of sentences understood conjunctively) and ϕ as the query. (We put all epistemic operators on hold for now.)

The computational complexity of query answering can be measured in several ways. To distinguish them, we follow the terminology common in database theory. The obvious way is to express the complexity in terms of the size of both KB and ϕ. This is called *combined complexity*. However, for all practical purposes, KB can be assumed to be very big and ϕ very small. In fact, it makes sense to assume that the size of ϕ is bounded by a small constant, and consider only the size of the KB to be relevant for the computational effort needed to answer queries. This is called *data complexity*. In the following, it will be clear in each case which one is meant.

12.2 The propositional case

We begin our investigations by considering a KB that is restricted to sentences taken from the propositional fragment of $\mathcal{L}$, which we take to be $\mathcal{L}$ without variables, quantifiers, function symbols, and $=$. So the propositional sentences of $\mathcal{L}$ are those which can be formed using primitive atoms, $\neg$, and $\vee$. Other Boolean connectives such as $\wedge$ or $\supset$ will be used freely as their usual abbreviations.

As literals and clauses play an important role in this and the following chapters, let us briefly recall what they stand for: a *literal* is either an atom or its negation; a *clause* is a disjunction of literals, and we often identify a clause with the set of literals it contains. A formula is said to be in *conjunctive normal form* (CNF) if it is a conjunction of clauses, and we often identify a formula in CNF with the set of clauses it contains. Atoms are sometimes called *positive* literals and their negations *negative* literals. The complement of an atom p is $\neg p$ and the complement of $\neg p$ is p. For any literal ρ, we sometimes write $\overline{\rho}$ to denote its complement.

It is well known that reasoning in propositional logic is decidable but intractable, even if we restrict ourselves to formulas in CNF.

Theorem 12.2.1: *Let* KB *and ϕ be propositional formulas in CNF. Then the problem of deciding whether* KB $\models \phi$ *is co-NP complete.*

What makes this result particularly vexing is the fact that the hardness is in the size of both KB and ϕ. If we imagine the KB to be very large, then this is a real problem, any advances in satisfiability solvers notwithstanding. If we want to use full logical reasoning, and we want it to be tractable, we need to be bold and drastically restrict the expressiveness of knowledge bases.

12.2.1 Knowledge bases as consistent sets of literals

So let us consider perhaps the simplest kind of knowledge base, where a KB consists of a finite, consistent set of propositional literals. We have the following:

Lemma 12.2.2: *Let* KB *be a finite, consistent set of propositional literals and ϕ any propositional formula. Let ϕ^* be ϕ with every atom p replaced by TRUE if $p \in$ KB and by FALSE if $\neg p \in$ KB. Then for any world w with $w \models$ KB, $w \models \phi$ iff $w \models \phi^*$.*

Proof: The proof is by induction on ϕ. For the base case, let $\phi = p$ for some atom p. If neither p nor $\neg p \in$ KB, then $\phi^* = \phi$ and we are done. If $p \in$ KB then $w \models p$ and $w \models$ TRUE. If $\neg p \in$ KB then $w \not\models p$ and $w \not\models$ FALSE. In any case, $w \models p$ iff $w \models p^*$.

For the induction, we only go over the case of $\neg\phi$. Disjunctions are handled analogously. Then $w \models \neg\phi$ iff $w \not\models \phi$ iff (by induction) $w \not\models \phi^*$ iff $w \models \neg\phi^*$ iff $w \models (\neg\phi)^*$ (since $\neg\phi^* = (\neg\phi)^*$). ∎

Theorem 12.2.3: *Let* KB*, ϕ and ϕ^* be as above. Then* KB $\models \phi$ *iff* $\models \phi^*$.

Proof: For the if direction, let $\models \phi^*$ and suppose $w \models$ KB. Then $w \models \phi^*$ and, by Lemma 12.2.2, $w \models \phi$.

Now let KB $\models \phi$ and let w be any world. Since ϕ^* mentions none of the atoms mentioned in KB, consider w^*, which is like w except $w^* \models \rho$ for all $\rho \in$ KB. A simple induction shows that $w \models \phi^*$ iff $w^* \models \phi^*$. By construction, $w^* \models$ KB and hence $w^* \models \phi$. By Lemma 12.2.2, $w^* \models \phi^*$ and thus $w \models \phi^*$. ∎

This theorem gives us a direct recipe for how to evaluate a query ϕ when the KB is a consistent set of literals as above: First compute ϕ^* and then decide whether ϕ^* is valid. The first part is easy and can clearly be done in polynomial time. The second part is, in the worst case, still a co-NP complete problem, but this time only in the size of the query, which, as mentioned above can be assumed to be small compared to the KB. In the best case, when all atoms of ϕ appear in the KB, the reasoning is easy, as ϕ^* is then a Boolean combination of TRUE and FALSE, which can be evaluated in linear time.

In case we are willing to transform queries into CNF, things become even easier, and we lose the co-NP completeness over the query.

Theorem 12.2.4: *Let* KB *be a finite, consistent set of propositional literals and ϕ a propositional sentence in CNF. Then* KB $\models \phi$ *iff every clause in ϕ contains either complementary literals or an element of* KB.

Proof: Let $\phi = \bigwedge c_i$. Since KB $\models \bigwedge c_i$ iff KB $\models c_i$ for all i, it suffices to consider the case of a single clause c.

For the only-if direction, if c contains complementary literals, then $\models c$ and hence KB $\models c$. Otherwise, suppose c contains a literal ρ from KB and let $w \models$ KB. Then $w \models \rho$ and hence $w \models c$.

Conversely, assume KB $\models c$ and suppose c contains no complementary literals and none of the literals in KB. Let w be a world such that $w \models \rho$ for all $\rho \in$ KB and for all $\rho \in c$, let $w \models \overline{\rho}$. Note that this can always be done since either KB contains neither ρ nor $\overline{\rho}$ or KB contains $\overline{\rho}$. But then $w \models$ KB and $w \not\models c$, a contradiction. ∎

An obvious consequence of this is that answering CNF queries is tractable:

Corollary 12.2.5: *Let* KB *and ϕ be as in the theorem. Then computing whether* KB $\models \phi$ *can be done in polynomial time in the size of* KB *and ϕ.*

12.3 The first-order case

In the rest of this chapter, we will be concerned with reasoning using first-order sentences of $\mathcal{L}$ in the KB. (We will for now, however, put function symbols aside for simplicity.) As we will see, it is much more difficult to obtain tractability in this case as the general undecidability of first-order logic raises the bar considerably. As noted above, even in the case of an *empty* KB, asking whether KB entails ϕ is the same as asking whether ϕ is valid, and this is already undecidable. Conversion into normal form similar to what we did in Theorem 12.2.4 is no longer of any help here, as the conversion itself is computable, and thus leaves the validity problem undecidable.

12.3.1 Knowledge bases in database form

There are, however, some special cases where the entailment problem is tractable. Perhaps the simplest is a knowledge base that lists the finite extensions of a finite number of predicates. This is precisely what is done in relational databases.

Definition 12.3.1: A KB is in *database* form if it consists of a finite set of sentences of the form $\forall \vec{x}[P(\vec{x}) \equiv \vec{x} = \vec{n_1} \vee \ldots \vee \vec{x} = \vec{n_k}]$. where $\vec{x}$ is a sequence of variables, and each $\vec{n_i}$ is a sequence of standard names, both of the length of the arity of the predicate P. Each predicate P can appear at most once in the KB.

(By $(\vec{x} = \vec{n})$ we mean $\bigwedge(x_i = n_i)$.) Note that we must use the $\equiv$ operator in these sentences to ensure that the extension of the predicate is precisely the set of tuples of standard names

named in the sentence.

Here is a small example of a KB in this form for the single predicate *Teach*, inspired by an example in Chapter 5.

$$\forall x, y\ (\mathit{Teach}(x, y) \equiv x = \mathit{tom} \wedge y = \mathit{sam} \vee \\ x = \mathit{tina} \wedge y = \mathit{sue} \vee x = \mathit{ted} \wedge y = \mathit{sue} \vee \\ x = \mathit{ted} \wedge y = \mathit{sandy}).$$

A KB like this always provides complete knowledge about the predicates involved: the students Sam and Sandy each have exactly one teacher whose identity is known, and Sue has two known teachers. In the earlier example in Figure 5.1, the student Sam may or not have had additional teachers, the student Sue either had Tom or Ted as a teacher, and the student Sandy had a teacher whose identity was not known.

It is not hard to see that for a KB in database form, if $w_1 \models \text{KB}$ and $w_2 \models \text{KB}$, then w_1 and w_2 must agree on the predicates that appear in the KB: they both satisfy $P(\vec{n})$ iff $\vec{x} = \vec{n}$ appears as one of the disjuncts on the right-hand side of the equivalence for P in the KB. Moreover, a KB in this form is always consistent, in that there is always at least one world w that satisfies the KB, namely the one that assigns true to $P(\vec{n})$ iff P appears in the KB and $\vec{x} = \vec{n}$ appears as one of the disjuncts.

It follows from these facts that it will be easy to determine if a query ϕ is entailed by a KB, provided the predicates in ϕ are those in the KB. We will show this by exhibiting an evaluation method U that returns 1 when the query is entailed, and 0 otherwise. The idea is that U will handle a query by breaking it apart until it gets to a primitive atom, which it evaluates by looking at the equivalence for that predicate in the KB.

The only real complication concerns quantified queries, like $\forall x\phi$. To handle these, U will check ϕ_n^x for every name n in the KB or in the query, and one additional name. As we will see, if all of these are true, then $\forall x\phi$ must be true as well. More precisely, for any set of sentences Γ let $H^+(\Gamma)$ denote the set of all standard names contained in Γ plus one new name that does not occur in Γ. When the context is clear, we will sometimes write H^+ instead of $H^+(\Gamma)$.

Given a KB and a query ϕ, the U procedure is defined as follows:

1. $U[\text{KB}, P(\vec{n})] = \begin{cases} 1 & \text{if } (\vec{x} = \vec{n}) \text{ appears on the right-hand side for } P \text{ in the KB;} \\ 0 & \text{otherwise;} \end{cases}$
2. $U[\text{KB}, n = m] = 1$ if n is identical to m, and 0 otherwise;
3. $U[\text{KB}, \neg\phi] = 1 - U[\text{KB}, \phi]$;
4. $U[\text{KB}, \phi \wedge \psi] = \min\{U[\text{KB}, \phi], U[\text{KB}, \psi]\}$;
5. $U[\text{KB}, \forall x\phi] = \min_{n \in H^+(\text{KB} \cup \{\phi\})} U[\text{KB}, \phi_n^x]$.

First we show that U is sound and complete provided the query ϕ only uses the predicates that appear in the KB.

Theorem 12.3.2: *Let* KB *be in database form and ϕ be a sentence of $\mathcal{L}$ that mentions only the* KB *predicates. Then* KB $\models \phi$ *iff* $U[\text{KB}, \phi] = 1$.

Proof: First let us consider the soundness. We show by induction on the length of ϕ that whenever $U[\text{KB}, \phi] = 1$ then KB $\models \phi$, and whenever $U[\text{KB}, \phi] = 0$ then KB $\models \neg\phi$. The base case, where ϕ is $P(\vec{n})$ is straightforward since U returns 1 when $\vec{x} = \vec{n}$ appears on the right-hand side of the equivalence for P in the KB, in which case KB $\models P(\vec{n})$, and U returns 0 otherwise, in which case KB $\models \neg P(\vec{n})$.

For the induction, we only prove the case $\forall x\phi$. (All other cases are straightforward.) Suppose $U[\text{KB}, \forall x\phi] = 1$. Then $U[\text{KB}, \phi^x_n] = 1$ for all $n \in H^+$ and hence, by induction, KB $\models \phi^x_n$ for all $n \in H^+$. By Corollary 2.8.11, KB $\models \forall x\phi$. Now suppose conversely that $U[\text{KB}, \forall x\phi] = 0$. Then for some $n \in H^+$, $U[\text{KB}, \phi^x_n] = 0$ and, by induction, KB $\models \neg\phi^x_n$ for some $n \in H^+$. Therefore KB $\models \neg\forall x\phi$.

Finally for completeness, we need to show that if KB $\models \phi$ then $U[\text{KB}, \phi] = 1$. So suppose that KB $\models \phi$. Since KB is consistent, KB $\not\models \neg\phi$. By the above soundness, $U[\text{KB}, \phi] \neq 0$, and therefore $U[\text{KB}, \phi] = 1$. ∎

Next we show that U runs in polynomial time.

Theorem 12.3.3: *Let* KB *and ϕ be as in the previous theorem. Then $U[\text{KB}, \phi]$ has polynomial data complexity.*

Proof: We prove the theorem for queries where all the universal and existential quantifiers appear at the front of the formula. (The more general case is left as an exercise.) Suppose that the size of the KB is m, and that there are q quantifiers at the front of the query followed by a quantifier-free body whose size is b. For each atomic query, U must scan the entire KB looking for an equality on the right-hand side of an equivalence, which will take no more than m steps. So for $q = 0$, the U procedure runs in time that is no more than bm. For each quantifier, U must consider a subquery for each element of H^+, and there will be at most $(m + b + q)$ of them. So for $q = 1$, U would take no more than $(bm)(m+b+q)$ steps. In general, for a query with q quantifiers, U will take no more than $(bm)(m + b + q)^q$ steps. Taking q and b to be constant, this is polynomial in m. ∎

This is a very rough analysis of the worst-case complexity of U. (See Section 12.4 for more on this.)

Corollary 12.3.4: *Let* KB *be in database form and ϕ be a sentence of $\mathcal{L}$ that mentions only the* KB *predicates. Then computing whether* KB $\models \phi$ *can be done in polynomial time*

in data complexity.

Note that this tractability result is for queries that only use the predicates that appear in the KB. Otherwise, the U procedure would not be sound. To see this, observe that if $U[\text{KB}, Q(n)] = 0$ for some Q that does not appear in the KB, then $U[\text{KB}, \neg Q(n)] = 1$ even though $\text{KB} \not\models \neg Q(n)$.

On the other hand, the query is free to use standard names that are not mentioned in the KB. In fact, it is not hard to generalize U to deal with knowledge bases where some predicates are given an infinite extension.

Let us call a formula e an *ewff* if it is quantifier-free and contains no predicate, function, or constant symbols. So ewffs are made out of equalities over standard names and variables, negations and conjunctions. Examples of ewffs are the right-hand sides of the equivalences appearing in a KB in database form. Now we extend our KB representation to allow any ewff to appear on the right-hand side.

Definition 12.3.5: A KB is in *generalized database* form if it consists of a finite set of sentences of the form $\forall \vec{x}[P(\vec{x}) \equiv e]$, where e is an ewff. Again, each predicate P can appear at most once in the KB.

This generalizes the previous definition and allows for predicates with infinite extensions, as in sentences like $\forall x, y\,(R(x, y) \equiv x \neq y)$. A simple change to U is sufficient to deal with knowledge bases with sentences like this. First observe that U is already sound and complete for any ewff.

Lemma 12.3.6: *For any ewff sentence e and any* KB, $U[\text{KB}, e] = 1$ *iff* $\models e$.

Proof: The proof is by a simple induction on the length of e. ∎

To handle knowledge bases in generalized database form, we need only change the behaviour of U on primitive atoms:

1. $U[\text{KB}, P(\vec{n})] = \begin{cases} 1 & \text{if } \forall \vec{x}(P(\vec{x}) \equiv e) \text{ appears in KB, where } U[\text{KB}, e^{\vec{x}}_{\vec{n}}] = 1; \\ 0 & \text{otherwise;} \end{cases}$

So with this version of U, instead of looking for $(\vec{x} = \vec{n})$ in the right-hand side of the equivalence, we use U itself to check that the right-hand side of the equivalence comes out true, appealing to Lemma 12.3.6. This evaluation of an ewff can be done in linear time. This then leads to the following:

Corollary 12.3.7: *Let* KB *be in generalized database form and ϕ be a sentence of $\mathcal{L}$*

that mentions only the KB *predicates. Then computing whether* KB $\models \phi$ *can be done in polynomial time in data complexity.*

12.3.2 Proper knowledge bases

A major limitation of the previous subsection was that the knowledge represented was required to be *complete*: for every query ϕ (whose predicates appear in the KB), either KB $\models \phi$ or KB $\models \neg\phi$. It was impossible to have a query come out unknown. In this section, we relax this restriction and consider allowing a simple form of incomplete knowledge. However, as we will see, even this small move introduces significant complications.

Perhaps the simplest change to the generalized database form would be to eliminate the strict equivalences for predicates. Instead of requiring a sentence $\forall\vec{x}(P(\vec{x}) \equiv e)$, which pins down the extension of the predicate, we can use implications to list some positive and negative instances of the predicate. This would allow some of the instances of the predicate to be left unknown.

Definition 12.3.8: A *proper* KB is a finite and consistent collection of sentences of the form $\forall\vec{x}(e \supset P(\vec{x}))$ or $\forall\vec{x}(e \supset \neg P(\vec{x}))$, where e is an ewff.

This obviously extends the idea of a KB in generalized database form since $\forall\vec{x}(P(\vec{x}) \equiv e)$ is equivalent to the conjunction of $\forall\vec{x}(e \supset P(\vec{x}))$ and $\forall\vec{x}(\neg e \supset \neg P(\vec{x}))$. What it allows in addition is a KB that says something like $\forall x(x = {}^{\#}1 \supset T(x))$ and $\forall x(x = {}^{\#}2 \supset \neg T(x))$ and no more. In this case, ${}^{\#}1$ is a known instance of T, ${}^{\#}2$ is a known non-instance of T, but the status of ${}^{\#}3$, for example, is left unspecified. So with a proper KB, there is a (potentially infinite) collection of primitive atoms that are known to be true, a (potentially infinite) collection of primitive atoms that are known to be false, and all the others are unknown.

The following lemma makes this explicit. Let

$$\begin{aligned}\text{LITS(KB)} = \{&P(\vec{n}) \mid \text{there is a } \forall\vec{x}(e \supset P(\vec{x})) \text{ in KB such that } \models e^{\vec{x}}_{\vec{n}}\}\\ \cup\ \{&\neg P(\vec{n}) \mid \text{there is a } \forall\vec{x}(e \supset \neg P(\vec{x})) \text{ in KB such that } \models e^{\vec{x}}_{\vec{n}}\}.\end{aligned}$$

Lemma 12.3.9: *For any world* w, $w \models$ KB *iff* $w \models$ LITS(KB).

With this, it can be seen that query answering for propositional queries is not much harder than before.

Theorem 12.3.10: *Let* KB *be proper and* ϕ *be in propositional CNF. Then* KB $\models \phi$ *iff every clause in* ϕ *contains either complementary literals or an element of* LITS(KB).

Proof: The proof is analogous to the proof of Theorem 12.2.4 using Lemma 12.3.9. ∎

Corollary 12.3.11: *Let* KB *and* ϕ *be as in the previous theorem. Then computing whether* $\text{KB} \models \phi$ *can be done in time polynomial in the size of* KB *and* ϕ.

It is possible to generalize this idea to handle queries of the form $\forall\vec{x}\phi$ where ϕ is a quantifier-free CNF formula. This is left as an exercise.

12.3.3 An evaluation-based reasoning procedure

To deal with quantified queries over a proper knowledge base in a more general way, however, it is best to look at a new evaluation procedure V that extends the previous U procedure. The idea is that for a query ϕ, V will return 1, 0, or $\frac{1}{2}$ according to whether ϕ is known to be true (that is, $\text{KB} \models \phi$), known to be false (that is, $\text{KB} \models \neg\phi$), or neither.

The V procedure is defined as follows:

1. $V[\text{KB}, P(\vec{n})] = \begin{cases} 1 & \text{if } \forall\vec{x}(e \supset P(\vec{x})) \text{ appears in KB, where } V[\text{KB}, e^{\vec{x}}_{\vec{n}}] = 1; \\ 0 & \text{if } \forall\vec{x}(e \supset \neg P(\vec{x})) \text{ appears in KB, where } V[\text{KB}, e^{\vec{x}}_{\vec{n}}] = 1; \\ \frac{1}{2} & \text{otherwise;} \end{cases}$
2. $V[\text{KB}, n = m] = 1$ if n is identical to m, and 0 otherwise;
3. $V[\text{KB}, \neg\phi] = 1 - V[\text{KB}, \phi]$;
4. $V[\text{KB}, \phi \wedge \psi] = \min\{V[\text{KB}, \phi], V[\text{KB}, \psi]\}$;
5. $V[\text{KB}, \forall x\phi] = \min_{n \in H^+(\text{KB} \cup \{\phi\})} V[\text{KB}, \phi^x_n]$.

Compared to U the definition of V differs only in how it treats primitive formulas. Of course, the additional value $\frac{1}{2}$ also effects the value returned for the Boolean connectives and quantifiers. Note, in particular, how the min and minus operations work with $\frac{1}{2}$. In the propositional case, these can be thought of as compact representations of Kleene's three-valued truth tables:

		$p \wedge q$			$\neg p$
	q	t	u	f	
	t	t	u	f	f
p	u	u	u	f	u
	f	f	f	f	t

Here t, f, u play the role of 1, 0, $\frac{1}{2}$, respectively.

As was the case with U, the V procedure runs in time that is polynomial in the size of the KB. Moreover, the V procedure is always sound:

Theorem 12.3.12: *Let* KB *be proper and let the query* ϕ *be any sentence of* $\mathcal{L}$. *If* $V[\text{KB}, \phi] = 1$ *then* $\text{KB} \models \phi$.

Proof: We prove that if $V[\text{KB}, \phi] = 1$ then $\text{KB} \models \phi$, and if $V[\text{KB}, \phi] = 0$ then $\text{KB} \models \neg\phi$. The argument is the same as the soundness part of Theorem 12.3.2. ∎

Note that there is no longer a restriction that the query only use predicates mentioned in the KB. V will correctly return $\frac{1}{2}$ for queries involving unknown predicates.

Since V behaves the same as U when applied to an ewff, we immediately obtain that V is complete for for any ewff, that is, Lemma 12.3.6 applies to V as well:

Lemma 12.3.13: *For any ewff sentence* e *and any* KB, $V[\text{KB}, e] = 1$ *iff* $\models e$.

The general story of the completeness of V, however, is more complicated. Suppose KB is empty and the query ϕ is $(p \vee \neg p)$ for some primtive atom p. Then $\text{KB} \models \phi$, but $V[\text{KB}, \phi] = \frac{1}{2}$. In general, tautologies cannot be detected by looking in the KB the way V does. This is precisely why, in Theorem 12.3.10, we needed to also test for complementary literals in clauses to obtain completeness.

But this difficulty shows up even without tautologies. Consider this knowledge base:

$$\text{KB} = \{\forall x(x = {}^{\#}1 \supset T(x)), \forall x(x = {}^{\#}2 \supset \neg T(x))\}.$$

Now let $q = T({}^{\#}1)$, $r = T({}^{\#}2)$ and $p = T({}^{\#}3)$, and consider this query:

$$\phi = (q \wedge (\neg r \wedge p)) \vee (\neg p \wedge (\neg r \wedge q)).$$

Then again we get that $\text{KB} \models \phi$, but $V[\text{KB}, \phi] = \frac{1}{2}$. There is, however, a tautology "hidden" here: if we convert ϕ to CNF, we get $[q \wedge \neg r \wedge (p \vee \neg p)]$.

This analysis suggests two ways of getting completeness for V. In the propositional case, we can convert the formula ϕ to CNF and filter out tautologous clauses. Then if $\text{KB} \models \phi$, by Theorem 12.3.10, each clause must contain an element of LITS(KB), and so $V[\text{KB}, \phi] = 1$. It follows therefore that in the propositional case, V will be complete for queries in CNF having no tautologous clauses.

The second way of ensuring the completeness of V, this time in the presence of universal and existential quantifiers, is to ensure that we can never get both p and $\neg p$ as subformulas of a query, and therefore nothing resembling a tautology. We will prove that this idea works for perhaps the simplest case, when the query only has positive literals. To further simplify matters, we assume these first-order queries are in *negation normal form*, that is, where a $\neg$ operator only appears in front of an atom or an equality. (Every formula can be put into this form by moving negations inward, replacing $\neg\forall x\phi$ by $\exists x\neg\phi$, replacing $\neg(\phi \wedge \psi)$ by $(\neg\phi \vee \neg\psi)$, and so on.) Let us call a formula in negation normal

form *negative* if all the atomic subformulas appear negated, and *positive* if all the atomic subformulas appear unnegated. (Equalities are unconstrained.) We will prove that V is complete for positive queries in negation normal form.

Definition 12.3.14: Let S be any set of worlds. The world $min(S)$ is defined to be the w^* such that $w^*[P(\vec{n})] = 1$ iff for every $w \in S$, $w[P(\vec{n})] = 1$.

Lemma 12.3.15: *Let S be a set of worlds, with $w^* = min(S)$, and let ϕ be a positive sentence in negation normal form. If $w^* \models \phi$, then for every $w \in S$, $w \models \phi$.*

Proof: The proof is a simple induction on the length of ϕ. ∎

Lemma 12.3.16: *Let* KB *be proper and let ϕ and ψ be positive sentences in negation normal form. Then we have the following:*

1. *If* KB $\models (\phi \vee \psi)$ *then* KB $\models \phi$ *or* KB $\models \psi$.
2. *If* KB $\models \exists x \phi$ *then for some standard name n,* KB $\models \phi_n^x$.

Proof: For the disjunction, assume to the contrary that KB $\not\models \phi$ and KB $\not\models \psi$. Then there is a w_1 such that $w_1 \models$ KB and $w_1 \not\models \phi$ and a w_2 such that $w_2 \models$ KB and $w_2 \not\models \psi$. Let $w^* = min\{w_1, w_2\}$. By Lemma 12.3.9, $w_1 \models$ LITS(KB) and $w_2 \models$ LITS(KB), and so $w^* \models$ LITS(KB), and therefore $w^* \models$ KB. By the above lemma, since $w_1 \not\models \phi$ and $w_2 \not\models \psi$, we have that $w^* \not\models \phi$ and $w^* \not\models \psi$. So $w^* \models$ KB but $w^* \not\models (\phi \vee \psi)$. Therefore, KB $\not\models (\phi \vee \psi)$. The argument for the existential operator is analogous. ∎

Theorem 12.3.17: *Let* KB *be proper and let the query ϕ be a positive sentence in negation normal form. If* KB $\models \phi$, *then* $V[\text{KB}, \phi] = 1$.

Proof: The proof is by induction on the length of ϕ. The theorem holds for primitive atoms (by definition of V), and for equalities and inequalities. It also holds by induction for conjunctions and universal quantifications. Finally, for disjunctions and existential quantifications, we apply the above lemma and once again use induction. ∎

Corollary 12.3.18: *Let* KB *and ϕ be as in the theorem. Then computing whether* KB $\models \phi$ *can be done in polynomial time in the size of* KB *and ϕ.*

It is not hard to show that the completeness theorem above also holds for negative queries.

(We simply use a $max(S)$ world instead of a $min(S)$ one in the above.) It is left as an exercise to show that the completeness also holds for queries that are neither positive nor negative overall, but where each predicate in the query appears only positively or only negatively. Finally, it is left as a somewhat more cumbersome exercise to show that the completeness holds for queries that are not in negation normal form. In this case, a formula is considered positive if every atomic subformula appears within the scope of an *even* number of $\neg$ symbols, and negative if every atomic subformula appears within the scope of an *odd* number of $\neg$ symbols.

12.4 Bibliographic notes

The logical omniscience problem was first discussed by Hintikka [65], who also proposed a solution, however, without considering issues of complexity [66]. The fact that the evaluation procedure for KBs in database form is tractable (see Theorem 12.3.3) is actually not that surprising as it relates directly to the tractability of query evaluation in relational databases [184]. V was first proposed in [118]. The paper mentions the connection to Kleene's three-valued logic [74] in the propositional case and introduces a wide class of sentences $\mathcal{NF}$ (for normal form) for which V is complete and which subsumes all the cases considered in this chapter. In [96] we showed that V coincides precisely with *tautological entailment*, a fragment of relevance logic [2, 37], for proper KBs and arbitrary queries. The computational complexity of V was studied in [131], including a tractability result for a large class of first-order queries. One limitation of V is that the only terms considered in either the KB or the query are variables and standard names. In [29] it is shown how to handle unkown individuals in the form of constants without sacrificing tractability.

12.5 Where do we go from here?

This chapter was about how to keep the problem of deciding whether or not KB $\models \phi$ computationally tractable by restricting the form of KB and ϕ. As we saw, one way to do this was to restrict the KB to be in what we called proper form. However, this imposed very strong requirements on the ϕ to ensure that the entailment did not hold because of properties of ϕ itself. But another approach is to ensure that all the logical properties of ϕ, that is all the hidden entailments it may contain, have been extracted beforehand.

We can sketch what this could mean in the propositional case. Suppose we have a propositional query ϕ. Let $C(\phi)$ be the set of all minimal non-tautologous clauses entailed by ϕ. (There can only be finitely many such clauses.) It is left as an exercise to show that ϕ and $C(\phi)$ are logically equivalent. A conjunction of clauses like $C(\phi)$ is said to be in

Blake canonical form. Moreover, if KB $\models \bigwedge C(\phi)$, then by Theorem 12.3.10, each clause must contain an element of LITS(KB), and so $V[\text{KB}, \bigwedge C(\phi)] = 1$. It follows therefore that in the propositional case, V will be complete for queries in Blake canonical form.

So this is an example of a class of queries that is complete for V without requiring the atoms to be all positive or all negative. It does require preprocessing the query to convert it to Blake canonical form, however. And this is all in the propositional case. It remains open how to apply anything like this idea to first-order queries.

Turning now to proper knowledge bases, it is also worth considering whether the tractability results can be preserved for knowledge bases that are more expressive than the proper ones. One easy generalization concerns the idea of basic and defined predicates.

We can think of a proper knowledge base as representing knowledge about a set of *basic* predicates. We can now imagine a knowledge base also characterizes a set of *defined* predicates using formulas of the form $\forall\vec{x}(P(\vec{x}) \equiv \phi)$, where P is the predicate and ϕ is its definition, any formula using only basic predicates. It is not hard to extend the V procedure to answer queries involving both basic and defined predicates in this way. With a bit of care, it is also possible to let the definition ϕ include other defined predicates, so long as we avoid circular definitions.

A more complex generalization of proper knowledge bases involves trying to incorporate terms other than variables and standard names. For example, it would be desirable to be able to use constants in the knowledge base to represent individuals known to have certain properties, without having to know who those individuals are. However, knowledge bases that are in proper form except using constants instead of standard names can already lead to disjunctive reasoning. For example, if we have a constant a where

$$\text{KB} = \{\forall x(x = a \supset P(x)), \forall x(x \neq a \supset Q(x))\},$$

then we get that KB $\models (P(\#5) \vee Q(\#5))$, even though KB $\not\models P(\#5)$ and KB $\not\models Q(\#5)$. So something like the V procedure cannot be used directly in this case. There are some ideas about what can be done with constants (see the bibliographic notes), but the case with function symbols more generally remains completely open.

12.6 Exercises

1. Prove Theorem 12.3.3 as stated, where queries may contain quantifiers not at the front of the formula.
2. Let KB be proper and ϕ be a quantifier-free CNF formula whose only free variable is x. Prove that KB $\models \forall x.\phi$ iff for all $n \in H^+$, every clause in ϕ_n^x contains either complementary literals or an element of LITS(KB).
3. Show that V is complete for queries in negation normal form that are neither positive

nor negative overall, but where each predicate in the query appears only positively or only negatively.

4. Extend the previous exercise to queries which are not in negation normal form. (Recall that a formula is considered positive if every atomic subformula appears within the scope of an *even* number of ¬ symbols, and negative if every atomic subformula appears within the scope of an *odd* number of ¬ symbols.)
5. Show that every propositional formula is logically equivalent to its Blake canonical form.

13 Tractable Reasoning

In the previous chapter, we introduced the idea of *tractable belief*, where it would always be computationally feasible for an agent to decide whether or not it believed something. This was motivated by the observation that the general problem of determining whether or not something is believed in $\mathcal{KL}$ or $\mathcal{OL}$ is computationally intractable (undecidable in the first-order case, and co-NP complete in the propositional case). This is because, among other things, our model of belief includes full logical reasoning: an objective sentence ϕ is believed given an objective knowledge base KB iff that KB logically entails ϕ.

In the previous chapter, we explored how this intractability could be avoided by limiting the representation language, that is, by assuming that the KB and ϕ could be represented in a certain restricted form. In particular, we showed that when the KB is proper and ϕ is positive, it would then be computationally feasible to determine whether or not KB logically entails ϕ. What did we give up to get this result? Most obviously perhaps, we had to give up only knowing disjunctions. Although proper knowledge bases do allow for incomplete knowledge, it is incomplete knowledge of a certain form only. For example, there is no proper knowledge base where $(p \vee q)$ is all that is known.

However useful proper knowledge bases might turn out to be, there may be cases where an agent has to deal with a knowledge base that is not proper. For example, an agent might simply be told that some non-proper ϕ is true. What should the agent do in this case? Perhaps the most obvious thing is to keep ϕ in the knowledge base, but somehow limit the *reasoning* that must be done with it. In other words, for cases of this sort, we want to consider a new form of belief where the agent is not required to believe all the logical consequences of its knowledge base. This is the direction we pursue in this chapter.

13.1 The approach

Perhaps the simplest story to tell about what should be believed when an objective KB might contain arbitrary sentences of $\mathcal{L}$ is to arrange things so that ϕ is believed iff $\phi \in$ KB. This would certainly give us a sound notion of belief (in the sense that everything believed would be logically entailed by the KB). It would also be computationally tractable, in that a procedure could determine if something was believed by simply looking for it in the KB.

The problem is that this version of belief is much too *syntactic*: it checks for membership of a certain syntactic expression in the KB. If $\forall x.P(x)$ is in the KB, then $\forall x.P(x)$ will indeed be believed, but $\forall y.P(y)$ will not be believed, even though this really amounts to the same belief. Furthermore, perfectly obvious sentences like $(^{\#}7 = {}^{\#}7)$ will not be believed unless they are explicitly placed in the KB to be retrieved later.

This suggests that even with a very minimal notion of belief, we may want to go beyond membership in the KB. But what additional sentences should be believed? If we perform *all* logically permissible manipulations on the sentences in the KB, we will end up doing full logical reasoning, and belief will once again be computationally intractable.

In this chapter, we want to consider a notion of belief where what is believed goes beyond membership in the KB, but not so far as full logical entailment. To do so, we will formalize a family of belief operators $\boldsymbol{B}_0$, $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, and so on, which include more and more of those logical entailments. The idea is that $\boldsymbol{B}_0$ will include the "obvious" beliefs given the KB, including those sentences that are members of the KB. Then $\boldsymbol{B}_1$ will include some additional less obvious beliefs, $\boldsymbol{B}_2$ even more, and so on.

13.1.1 Desiderata

At the highest level, the properties of limited reasoning we are looking for are these:

- *expressiveness*: unlike in the previous chapter, for any sentence ϕ of $\mathcal{L}$, the sentence $\boldsymbol{O}\phi$ will be satisfiable, and moreover $\models(\boldsymbol{O}\phi \supset \boldsymbol{B}_0\phi)$.
- *cumulativity*: for any k and any ϕ, $\models(\boldsymbol{B}_k\phi \supset \boldsymbol{B}_{k+1}\phi)$.
- soundness: for any k, any KB and ϕ, if $\models(\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_k\phi)$, then $\models(\text{KB} \supset \phi)$.
- *eventual completeness*: for any KB and any ϕ, if $\models(\text{KB} \supset \phi)$, then there will be some k such that $\models(\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_k\phi)$.
- tractability: for any k, KB, and α, the question as to whether $\models(\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_k\alpha)$ will be decidable (and have polynomial data complexity in cases of interest).

So while it will be computationally feasible to determine if $\boldsymbol{B}_k\alpha$ is true (with an effort that depends on the k), if an agent really needs to determine whether or not α is true, it may have to look at higher and higher values of k. Because of the undecidability of first-order logic, it will be undecidable to determine if there exists a k such that $\boldsymbol{B}_k\alpha$ is true.

One possible way to satisfy the above requirements might be to start with a sound and complete logical reasoning procedure (like Resolution, say), but cut it off after k steps. In other words, we could arrange the semantics so that $\boldsymbol{B}_k\phi$ is true iff ϕ can be derived from the KB by the reasoning procedure in k or fewer steps. While this is better than the basic syntactic approach, it still has some drawbacks. For one thing, we need to worry about the fact that equality and standard names have a special status in $\mathcal{L}$ not seen in standard first-order logic (or in procedures like Resolution). For example, $({}^{\#}3 \neq {}^{\#}5)$ is valid in $\mathcal{L}$, but not in first-order logic. We might also want to define $\boldsymbol{B}_k$ in such a way that that $\boldsymbol{B}_k(\alpha \wedge \beta)$ holds iff $\boldsymbol{B}_k(\beta \wedge \alpha)$ holds, even when the reasoning procedure might need some extra steps to go from one conjunction to the other. To obtain these and other desirable closure properties like commutativity (as part of the desiderata, in other words), we prefer to define belief not in terms of a reasoning procedure, but using some notion of epistemic state, as before.

13.1.2 Two sources of intractability

If we think of how intractability arises in trying to determine whether or not KB logically entails ϕ, there are really two sources:

1. It can be too hard to make full use of the information provided by the KB.
2. It can be too hard to see if ϕ should be believed because of its own properties.

For the first item, consider, for example, a KB consisting of a set of ground clauses (that is, clauses with no variables) and where ϕ is some ground atom p. Determining whether KB $\models \phi$ in this case is the same as determining if KB $\cup \{\neg p\}$ is unsatisfiable. This task is co-NP-hard and is believed to require a number of steps that is exponential in the number of clauses in the worst case.

For the second item, consider the case where the KB is empty. Determining whether KB $\models \phi$ in this case is the same as determining if ϕ is logically valid. In the propositional case, this is not too hard when ϕ is small (relative to the size of the KB): we can convert ϕ to *CNF* and ensure that each resulting clause is a tautology. But for the full language with quantifiers, the task is unsolvable.

To deal with these two items, we will be proposing a new model of belief in this chapter with two separate mechanisms to keep the reasoning tractable. For the first item above, we will generalize the notion of epistemic state to be sets of what we will call "extended" worlds; for the second item, we will preprocess the KB and the query ϕ using Skolemization and term substitution. The exact details will be presented beginning in the next section, but here is an informal outline of those two ideas.

13.1.3 Using extended worlds

In previous chapters, when we talked about an epistemic state, we meant a set of worlds: the epistemic state where a given KB was all that is known was defined as the set of all worlds w such that $w \models$ KB. For tractable reasoning however, this notion of epistemic state is too coarse, as it lumps all logically equivalent knowledge bases together. For example, for KB $= \{p, (p \supset q)\}$, we want an epistemic state where $\boldsymbol{B}_0(q)$ is false, but for the logically equivalent KB $= \{p, (p \supset q), q\}$, we want an epistemic state where $\boldsymbol{B}_0(q)$ is true.

In this chapter, we will be using a finer-grained notion of epistemic state based on sets of *extended* worlds. An extended world will be defined as one where atomic sentences are mapped to one of three values, $\{0, 1, *\}$. A world that assigns p to $*$ is taken to support both the truth and the falsity of p. Such a world will then be able to support the truth of both p and $(p \supset q)$ without also supporting the truth of q. So the epistemic state e_1 made up of all extended worlds where p and $(p \supset q)$ are supported is a superset of an e_2 where q is also supported. In this way, in e_1 we can end up believing p and $(p \supset q)$ without

believing q, whereas in e_2, the sentence q is believed as well.

In going from belief at level k to belief at level $k+1$, we will be moving from an epistemic state e to another one, $S(e)$, that has fewer extended worlds and where more sentences are believed. As we will see, the idea is to eliminate some of the worlds where an atom is assigned $*$. In the case of e_1 above, we will end up eliminating all the worlds where p is assigned $*$, which means that $S(e_1)$ will be e_2. More generally, if the epistemic state is the set of all extended worlds that support $\{(p \vee q), (\neg p \vee r), (\neg q \vee r)\}$, then the clauses $(p \vee q)$ and $(s \vee \neg s)$ and their supersets will be believed at levels 0, the clause $(p \vee r)$ and its supersets will believed at level 1 (after one application of S), and finally, the clause r and its supersets will be believed at level 2 (after two applications of S).

13.1.4 Using Skolemization

The idea of epistemic states as sets of extended worlds works fine for beliefs that do not involve quantifiers, but as noted above, it cannot be the whole story. The logic for beliefs involving quantifiers goes further. As we will see, the semantics of $\boldsymbol{O}$ will use *Skolemization* to eliminate existential variables, and the semantics of $\boldsymbol{B}_k$ will first use the dual of Skolemization (also called Herbrandization) to eliminate universal variables, and then use a bounded form of term substitution to produce a ground sentence. (Skolemization involves replacing any existentially quantified variable in a formula by a new function symbol used nowhere else whose arguments are the universally quantified variable it appears within the scope of. *Dual-Skolemization* involves replacing any universally quantified variable in a fomula by a new function symbol whose arguments are the existentially quantified variable it appears within the scope of.) Overall, we will get reductions like the following:

1. $\boldsymbol{O}\phi$ will hold iff $\boldsymbol{O}\forall\vec{x}.\psi$ holds (where the formula ψ is a Skolemized version of ϕ with no quantifiers);
2. $\boldsymbol{B}_k\phi$ will hold iff $\boldsymbol{B}_k\exists\vec{x}\psi$ holds (where ψ is a dual-Skolemized version of ϕ with no quantifiers) iff there are terms $\vec{t_0}, \ldots, \vec{t_k}$ such that $\boldsymbol{B}_k(\psi^{\vec{x}}_{\vec{t_0}} \vee \cdots \vee \psi^{\vec{x}}_{\vec{t_k}})$ holds.

The second item above is a bounded application of what is known as Herbrand's Theorem, a way of going from unsatisfiablility in classical first-order logic to its propositional counterpart. Here is the relevant theorem from classical logic:

Proposition 13.1.1: [Herbrand] *Let Φ be a set of formulas with no quantifiers. If the set Φ is first-order unsatifiable (with the free variables interpreted universally) then so is some finite subset of $\{\phi^{\vec{x}}_{\vec{t}} \mid \phi \in \Phi$ and $\vec{t}$ is ground$\}$.*

As a special case (mirroring item 2 above), we have the following:

Corollary 13.1.2: *If ϕ is first-order valid, then there is a number k and terms $\vec{t}_0, \ldots, \vec{t}_k$ such that the sentence $(\psi^{\vec{x}}_{\vec{t}_0} \vee \cdots \vee \psi^{\vec{x}}_{\vec{t}_k})$ is first-order valid, where ψ is a dual-Skolemized version of ϕ.*

To get a sense of how these reductions avoid the problem of having to believe all classically valid sentences, consider for example $\exists x \forall y (P(y) \vee \neg P(x))$. This sentence is valid in first-order logic and so is supported by all extended worlds, as is $\exists x.\psi$, where ψ is its dual-Skolemized version, $(P(f(x)) \vee \neg P(x))$. However, there is no single term t such that ψ^x_t is supported by all extended worlds. Because of this, $\boldsymbol{B}_0 \exists x.\psi$ need not be true, according to the reduction above. (However, $\boldsymbol{B}_1 \exists x.\psi$ will be true in this case since there are two terms t and u such that $(\psi^x_t \vee \psi^x_u)$ is supported by all extended worlds, namely $t = a$ and $u = f(a)$. Other first-order valid sentences will require higher levels of belief.)

But having made this move to term substitution in beliefs, we need to do something related in the KB using Skolemization. Consider the sentence $\exists x.P(x)$. This will be believed in an epistemic state e at level 0 only if there is a t such that $P(t)$ is supported by all the extended worlds in e. This means that it is not sufficient that the extended worlds in e support an existential; they must all agree on some term t. So an epistemic state where say $\boldsymbol{O}\exists x[P(x) \wedge Q(x)]$ is true should be the set of extended worlds that support the Skolemized version of this KB, that is, something like $[P(a) \wedge Q(a)]$, for some Skolem constant a to ensure that $\exists x.P(x)$ is believed. In general, the Skolemization of the KB is needed to guarantee the existence of the terms now required for believing existentials. (It will be necessary to ensure that the choice of Skolem constants is irrelevant so that, for example, $\boldsymbol{O}\exists y[Q(y) \wedge P(y)]$ also comes out true even if it uses a different constant.)

These are the main ideas of this chapter. The technical details are somewhat involved since we are dealing with what amounts to a new form of logical entailment. For this reason, we will be taking things slowly, one step at a time. In Section 13.2, we begin by considering formulas $\boldsymbol{B}_k \exists \vec{x} \phi$ and $\boldsymbol{O}\forall \vec{x} \phi$ where $k = 0$ and ϕ is what we call a *qfree* formula, a quantifier-free objective formula. (We will write $\exists \phi$ and $\forall \phi$ to mean the existential or universal closure of ϕ, respectively.) In Section 13.3, we will consider $\boldsymbol{B}_k$ when $k > 0$. Finally, in Section 13.4, we will consider $\boldsymbol{B}_k \phi$ and $\boldsymbol{O}\phi$ when ϕ is an arbitrary objective formula. (For simplicity, this chapter deals with objective belief only. The case where a $\boldsymbol{B}_k$ operator can appear in the scope of another $\boldsymbol{B}_{k'}$ is left as an exercise.)

13.2 A first logic of limited reasoning

The language we will be using in this chapter is like $\mathcal{OL}$ except using $\boldsymbol{B}_k \alpha$ instead of $\boldsymbol{K}\alpha$. (In this section, we have $k = 0$.) We follow our usual naming conventions: α and β for

arbitrary formulas, ϕ and ψ for objective ones, σ for subjective ones. We use p to refer to ground atomic formulas (that is, atomic formulas including equalities without variables), and ρ and τ to refer to literals, with $\bar{\rho}$ as the complement of ρ. We let b and d refer to clauses, as finite sets of literals. Finally, we use θ to refer to ground substitutions. For any formula ϕ, $\phi\theta$ is the sentence that results from replacing all free variables x in ϕ by $\theta(x)$ and $GND(\phi)$ is the set of $\phi\theta$ sentences over all θ.

13.2.1 Extended worlds and epistemic states

As noted above, the semantics of the logic relies on a notion of extended world:

Definition 13.2.1: [World] An *extended world* w is a function from ground atoms to $\{0, 1, *\}$. (When the context is clear in this chapter, we will just call them "worlds.") An extended world w is called *standard* if there is a two-valued world w' from $\mathcal{L}$ such that for every p, $w[p] = 1$ iff $w' \models p$, and $w[p] = 0$ iff $w' \models \neg p$.

Note that an extended world maps all ground atoms including equalities to values, not just the primitive ones as in $\mathcal{L}$. So, for example, we can have $w[P(n)] = 1$ for every standard name n, and still have $w[P(a)] = 0$ for some constant a. Similarly, there are extended worlds where $w[n = n] = 0$.

Since worlds can support both the truth and falsity of sentences, we use two separate support relations, $\models_{\mathrm{T}}$ and $\models_{\mathrm{F}}$ defined as follows:

Definition 13.2.2: [World support] For any world w and qfree sentence ϕ, the relations $w \models_{\mathrm{T}} \phi$ and $w \models_{\mathrm{F}} \phi$ are defined recursively as follows:

1. $w \models_{\mathrm{T}} p$ iff $w[p] \neq 0$;
   $w \models_{\mathrm{F}} p$ iff $w[p] \neq 1$.
2. $w \models_{\mathrm{T}} \neg\phi$ iff $w \models_{\mathrm{F}} \phi$;
   $w \models_{\mathrm{F}} \neg\phi$ iff $w \models_{\mathrm{T}} \phi$.
3. $w \models_{\mathrm{T}} (\phi \vee \psi)$ iff $w \models_{\mathrm{T}} \phi$ or $w \models_{\mathrm{T}} \psi$;
   $w \models_{\mathrm{F}} (\phi \vee \psi)$ iff $w \models_{\mathrm{F}} \phi$ and $w \models_{\mathrm{F}} \psi$.

For a set of qfree sentences Φ, $w \models_{\mathrm{T}} \Phi$ means that $w \models_{\mathrm{T}} \phi$ for every $\phi \in \Phi$.

It is useful to define a notion of *strong entailment* based on the idea of extended worlds:

Definition 13.2.3: Let ϕ and ψ be qfree sentences. Then $\phi \Rightarrow \psi$ iff for all extended worlds w, if $w \models_{\mathrm{T}} \phi$ then $w \models_{\mathrm{T}} \psi$.

Note that strong entailment is a subset of logical entailment (that is, if $\phi \Rightarrow \psi$ then the sentence $(\phi \supset \psi)$ is valid in $\mathcal{L}$), but it is a proper subset: $(p \wedge (\neg p \vee q)) \not\Rightarrow q$, for example. This is because there is an extended world w where $w \models_{\mathrm{T}} (p \wedge (\neg p \vee q))$ but $w \not\models_{\mathrm{T}} q$, namely one where $w[p] = *$ and $w[q] = 0$. There is, in fact, a close connection between the two notions:

Proposition 13.2.4: *Let ϕ and ψ be qfree sentences that use atomic sentences $p_0, \ldots, p_k$. Then $(\phi \supset \psi)$ is valid in $\mathcal{L}$ iff $\phi \Rightarrow (\psi \vee \bigvee(p_i \wedge \neg p_i))$.*

Turning now to epistemic states, here is their definition:

Definition 13.2.5: [Epistemic state] An *extended epistemic state* is any set of extended worlds. (When the context is clear in this chapter, we drop the word "extended.")

13.2.2 Equality and standard names

Extended worlds, while defined for all ground atoms, have no special provisions for equality sentences or for the denotations of terms. These are handled in the logic by highlighting two special sets of formulas:

Definition 13.2.6: $UNA = \{(n \neq n') \mid n \text{ and } n' \text{ are distinct standard names}\}$.

Definition 13.2.7: Let EQ be the following infinite set of formulas:

1. $(x = x)$,
2. $\neg(x = y) \vee (y = x)$,
3. $\neg(x = y) \vee \neg(y = z) \vee (x = z)$,
4. $\neg(x_1 = y_1) \vee \cdots \vee \neg(x_k = y_k) \vee (f(x_1, \ldots, x_k) = f(y_1, \ldots, y_k))$,
   for every k-ary function symbol f,
5. $\neg(x_1 = y_1) \vee \cdots \vee \neg(x_k = y_k) \vee \neg P(x_1, \ldots, x_k) \vee P(y_1, \ldots, y_k)$,
   for every k-ary predicate symbol P.

We let $GEQ = UNA \cup GND(EQ)$.

The main property of $\mathcal{L}$ we will be using is this:

Theorem 13.2.8: *A sentence ϕ is valid in $\mathcal{L}$ iff $\{\neg\phi\} \cup UNA \cup EQ$ is first-order unsatisfiable.*

This is a direct corollary of Theorem 2.8.6.

13.2.3 Truth and validity

We are now ready to define validity in this logic, where we only consider subformulas $\boldsymbol{B}_0\exists\phi$ and $\boldsymbol{O}\forall\phi$, where ϕ is qfree.

Definition 13.2.9: [Validity] For any extended world w, extended epistemic state e and sentence α, the relation $e, w \models \alpha$ is defined recursively as follows:

1. $e, w \models p$ iff $w \models_{\mathrm{T}} p$ for ground atom p;
2. $e, w \models \neg\alpha$ iff $e, w \not\models \alpha$;
3. $e, w \models (\alpha \vee \beta)$ iff $e, w \models \alpha$ or $e, w \models \beta$;
4. $e, w \models \exists x\alpha$ iff for some n, $e, w \models \alpha^x_n$;
5. $e, w \models \boldsymbol{B}_0\exists\phi$ iff there is a substitution θ such that for all $w' \in e$, $w' \models_{\mathrm{T}} \phi\theta$.
6. $e, w \models \boldsymbol{O}\forall\phi$ iff for all w', $w' \in e$ iff $w' \models_{\mathrm{T}} \mathit{GND}(\phi) \cup \mathit{GEQ}$.

We can write $w \models \alpha$ when α is objective, and $e \models \alpha$ when α is subjective. We say that e is *representable* iff $e \models \boldsymbol{O}\phi$ for some sentence ϕ. Finally, for any sentence α, we write $\models \alpha$ and say that α is *valid* iff $e, w \models \alpha$ for every representable e and every standard w.

Rules (1)-(4) are the usual ones (like in $\mathcal{L}$). Note that validity is defined wrt standard worlds only, so that the logic is two-valued except within a belief. Rules (5) and (6) define belief in terms of sets of worlds analogously to what was done in $\mathcal{KL}$ and $\mathcal{OL}$.

Notice there is no special rule for equality in this logic. Outside of belief, standard worlds deliver all the expected properties from $\mathcal{L}$. Within belief, the ground instances of the axioms of equality (including *UNA*) are conceptually added to the knowledge base via Rule (6) to be reasoned with like anything else. So although there are extended worlds w where $w \not\models_{\mathrm{T}} (n = n)$ and $w \not\models_{\mathrm{T}} (n' \neq n)$ for distinct names n and n', both $\boldsymbol{B}_0(n = n)$ and $\boldsymbol{B}_0(n' \neq n)$ end up being valid, since the equality and inequality sentences are in *GEQ*. Similarly, even when $\boldsymbol{B}_0(a = n)$ is true (for some constant a), $\boldsymbol{B}_0(n = a)$ can still be false. (However, as we will see later, the sentence $\boldsymbol{B}_1(n = a)$ will be true because the sentence $(a = n \supset n = a) \in \mathit{GEQ}$.)

13.2.4 Properties of limited belief

Before going on to extended notions of belief, let us consider how $\boldsymbol{B}_0$ compares to $\boldsymbol{K}$ as seen in previous chapters. The main observation is that $\boldsymbol{B}_0$ is closed under strong entailment, but not under logical entailment:

Theorem 13.2.10: *For any qfree sentences ϕ and ψ, if $\phi \Rightarrow \psi$, then $\models (\boldsymbol{B}_0\phi \supset \boldsymbol{B}_0\psi)$. However, there is a ϕ and ψ such that $\models (\phi \supset \psi)$, but $\not\models (\boldsymbol{B}_0\phi \supset \boldsymbol{B}_0\psi)$.*

Proof: For the first part, note that if $e \models \boldsymbol{B}_0\phi$, then for all $w' \in e$, $w' \models_{\mathrm{T}} \phi$, which implies that for all $w' \in e$, $w' \models_{\mathrm{T}} \psi$, since $\phi \Rightarrow \psi$. For the second part, let p and q be distinct atomic sentences. Let $\phi = (p \wedge (\neg p \vee q))$ and $\psi = (\phi \wedge q)$. Then $\models (\phi \equiv \psi)$. Now suppose that $e \models \boldsymbol{O}\phi$. Then $e \models \boldsymbol{B}_0\phi$ but $e \not\models \boldsymbol{B}_0\psi$. So $\not\models (\boldsymbol{B}_0\phi \supset \boldsymbol{B}_0\psi)$. ∎

Corollary 13.2.11: *Let ϕ and ψ be qfree formulas. Suppose that that for every θ, $\phi\theta \Rightarrow \psi\theta$. Then $\models (\boldsymbol{B}_0\exists\phi \supset \boldsymbol{B}_0\exists\psi)$.*

Proof: Suppose $e \models \boldsymbol{B}_0\exists\phi$. Then there is a θ such that $e \models \boldsymbol{B}_0\phi\theta$. Since $\phi\theta \Rightarrow \psi\theta$, by Theorem 13.2.10, $e \models \boldsymbol{B}_0\psi\theta$. Therefore $e \models \boldsymbol{B}_0\exists\psi$. ∎

As a result of this closure under strong entailment, we get the expected belief equivalences:

Corollary 13.2.12: *[Equivalent beliefs] For any qfree sentences ϕ, ψ, and χ, the following sentences are valid:*

$\boldsymbol{B}_0\phi \equiv \boldsymbol{B}_0(\phi \wedge \phi)$;
$\boldsymbol{B}_0\phi \equiv \boldsymbol{B}_0(\phi \vee \phi)$;
$\boldsymbol{B}_0\phi \equiv \boldsymbol{B}_0\neg\neg\phi$;
$\boldsymbol{B}_0(\phi \wedge \psi) \equiv \boldsymbol{B}_0(\psi \wedge \phi)$;
$\boldsymbol{B}_0(\phi \vee \psi) \equiv \boldsymbol{B}_0(\psi \vee \phi)$;
$\boldsymbol{B}_0(\phi \wedge (\psi \wedge \chi)) \equiv \boldsymbol{B}_0((\phi \wedge \psi) \wedge \chi))$;
$\boldsymbol{B}_0(\phi \vee (\psi \vee \chi)) \equiv \boldsymbol{B}_0((\phi \vee \psi) \vee \chi))$;
$\boldsymbol{B}_0(\phi \wedge (\psi \vee \chi)) \equiv \boldsymbol{B}_0((\phi \wedge \psi) \vee (\phi \wedge \chi))$;
$\boldsymbol{B}_0(\phi \vee (\psi \wedge \chi)) \equiv \boldsymbol{B}_0((\phi \vee \psi) \wedge (\phi \vee \chi))$;
$\boldsymbol{B}_0(\neg(\phi \wedge \psi)) \equiv \boldsymbol{B}_0(\neg\phi \vee \neg\psi)$;
$\boldsymbol{B}_0(\neg(\phi \vee \psi)) \equiv \boldsymbol{B}_0(\neg\phi \wedge \neg\psi)$.

Among other things, this corollary shows that a sentence is believed iff its conversion into *CNF* (defined in the next subsection) is believed. For the same reason, we get this:

Corollary 13.2.13: *[Combinations of beliefs] For any qfree sentences ϕ, and ψ, the following sentences are valid:*

$(\boldsymbol{B}_0\phi \vee \boldsymbol{B}_0\psi) \supset \boldsymbol{B}_0(\phi \vee \psi)$;
$\boldsymbol{B}_0(\phi \wedge \psi) \supset (\boldsymbol{B}_0\phi \wedge \boldsymbol{B}_0\psi)$.

All these nice closure properties of belief will continue to hold when we move to higher

levels of k and to more general belief sentences. However, one property here that will not carry over is the following:

Theorem 13.2.14: *For any qfree sentences ϕ and ψ, $\models (\boldsymbol{B}_0\phi \wedge \boldsymbol{B}_0\psi) \supset \boldsymbol{B}_0(\phi \wedge \psi)$.*

Proof: The theorem follows from the following observation: if every $w \in e$ satisfies $w \models_{\mathrm{T}} \phi$ and $w \models_{\mathrm{T}} \psi$, then every $w \in e$ satisfies $w \models_{\mathrm{T}} (\phi \wedge \psi)$. ∎

13.3 Higher levels of belief and satisfying the desiderata

Let us now turn our attention to $\boldsymbol{B}_k\phi$ where $k > 0$. The idea, as mentioned in Section 13.1 is the following: we start with e for $\boldsymbol{B}_0$, but we use a subset $S(e)$ for $\boldsymbol{B}_1$, and a further subset $S(S(e))$ for $\boldsymbol{B}_2$, and so on. Here are the definitions:

Definition 13.3.1: [Unsupported literals] $U(w) = \{p \mid w[p] = 0\} \cup \{\neg p \mid w[p] = 1\}$.

The unsupported literals of w are the literals that w says cannot be true (allowing for $*$).

Definition 13.3.2: [Eliminated world] e eliminates world w iff there is a ground atom p such that for every world $w' \in e$, if $U(w) \subseteq U(w')$, then $w'[p] = *$.

Intuitively, e eliminates w if there is some p such that the claims made by w (in terms of what literals cannot be true) depend on p having value $*$. In other words, if we only kept worlds in e where p had value 0 or 1, no worlds would support the claims made by w.

Definition 13.3.3: [Successor epistemic state] $S(e) = e - \{w \mid e \text{ eliminates } w\}$.

Note that if w is standard, it is never eliminated since for no p do we have $w[p] = *$. In other words, if $w \in e$ and w is standard, then for every k, $w \in S^k(e)$.

Intutitively, S gives us a semantic version of what is known as propositional *Resolution*. Multiple applications of S will correspond to multiple steps of Resolution (and is what will lead to eventual completeness). We can make this precise as follows:

Definition 13.3.4: For any set of ground clauses C, $RP(C)$ is the set of clauses defined by

$$RP(C) = C \cup \{(b \cup d) \mid \text{for some } p,\ (\{p\} \cup b) \in C,\ (\{\neg p\} \cup d) \in C\}.$$

Notice that RP applies one step of propositional Resolution to C, and in general, RP^k applies k steps. Then we have the following correspondence between RP and S:

Lemma 13.3.5: *Let C be any set of ground clauses and let $e = \{w \mid w \models_{\mathrm{T}} C\}$. Then $S^k(e) = \{w \mid w \models_{\mathrm{T}} RP^k(C)\}$.*

Proof: The lemma holds by induction on k. It suffices to show that if $e = \{w \mid w \models_{\mathrm{T}} C\}$ then $S(e) = \{w \mid w \models_{\mathrm{T}} RP(C)\}$.

($\Rightarrow$) We show that if $w \not\models_{\mathrm{T}} RP(C)$ then e eliminates w, and so $w \notin S(e)$. Since $w \not\models_{\mathrm{T}} RP(C)$, there is $(\{p\} \cup b) \in C$, $(\{\neg p\} \cup d) \in C$, such that $w \not\models_{\mathrm{T}} (b \cup d)$. So $(b \cup d) \subseteq U(w)$. Therefore, for any $w' \in e$ such that $U(w) \subseteq U(w')$, it follows that $w' \not\models_{\mathrm{T}} (b \cup c)$ and therefore $w'[p] = *$. Hence e eliminates w.

($\Leftarrow$) We show that if $w \models_{\mathrm{T}} RP(C)$ then e does not eliminate w, and so $w \in S(e)$. To do so, we show that for every p, there is a $w' \in e$ such that $U(w) \subseteq U(w')$ and where $w'[p] \neq *$. First, suppose that $w[p] \neq *$; then let $w' = w$ and the claim is satisfied. Otherwise, if $w[p] = *$, define w' to be like w except on p, where $w'[p] = 1$ if for some $(\{p\} \cup b) \in C$, $w \not\models_{\mathrm{T}} b$, and 0 otherwise. So $U(w) \subseteq U(w')$ and $w'[p] \neq *$. To show that $w' \in e$, we show that for any $d \in C$, $w' \models_{\mathrm{T}} d$. There are three cases.

1. If d does not include p or $\neg p$, then $w' \models_{\mathrm{T}} d$ since $w \models_{\mathrm{T}} d$.
2. If $d = (\{p\} \cup d')$ then there are two subcases: if $w \not\models_{\mathrm{T}} d'$, then $w'[p] = 1$ and so $w' \models_{\mathrm{T}} d$; if $w \models_{\mathrm{T}} d'$, then $w' \models_{\mathrm{T}} d$.
3. If $d = (\{\neg p\} \cup d')$ then there are two subcases: if $w'[p] = 0$, clearly, $w' \models_{\mathrm{T}} d$; if $w'[p] = 1$, then there is an $(\{p\} \cup b) \in C$ where $w \not\models_{\mathrm{T}} b$. Since $w \models_{\mathrm{T}} RP(C)$, $w \models_{\mathrm{T}} (b \cup d')$ and so $w \models_{\mathrm{T}} d'$. It follows that $w' \models_{\mathrm{T}} d$. ∎

Given this definition of S, the change to the logic to handle $\boldsymbol{B}_k$ is small. We generalize the rule for $\boldsymbol{B}_0$ in the semantics as follows:

5. $e, w \models \boldsymbol{B}_k \exists \phi$ iff there are substitutions $\theta_0, \ldots, \theta_k$ such that for all $w' \in S^k(e)$, $w' \models_{\mathrm{T}} (\phi\theta_0 \vee \cdots \vee \phi\theta_k)$.

So, for example, $e \models \boldsymbol{B}_2 \exists x.P(x)$ iff there are ground terms t_0, t_1 and t_2 (not necessarily standard names) such that $e \models \boldsymbol{B}_2(P(t_0) \vee P(t_1) \vee P(t_2))$ iff there are ground terms t_0, t_1 and t_2 such that for every $w \in S(S(e))$, $w \models_{\mathrm{T}} P(t_0)$ or $w \models_{\mathrm{T}} P(t_1)$ or $w \models_{\mathrm{T}} P(t_2)$.

Before looking at the general properties of this new logic, let us consider a simple example involving equality:

We show that $\models \boldsymbol{O}\forall x[x \neq {}^{\#}5 \supset P(x)] \supset \boldsymbol{B}_1 P({}^{\#}7)$.

Let $\phi = (x \neq {}^{\#}5 \supset P(x))$, $C = GND(\phi) \cup GEQ$, and $e = \{w \mid w \models_{\mathrm{T}} C\}$. If $e' \models \boldsymbol{O}\forall x\phi$, then $e' = e$ and so to prove the validity, it is sufficient to show that $e \models \boldsymbol{B}_1 P({}^{\#}7)$. We have that

($^{\#}7 = {}^{\#}5 \vee P(^{\#}7)) \in C$ from the grounding of ϕ, and ($^{\#}7 \neq {}^{\#}5) \in C$ from *UNA*. So $P(^{\#}7) \in RP(C)$. It then follows from Lemma 13.3.5 that $e \models \boldsymbol{B}_1 P(^{\#}7)$.

13.3.1 Satisfying the desiderata

Let us now return to the desiderata from Section 13.1. (The property of expressiveness, however, will have to wait to the next section where we can use the same sentence, possibly with quantifiers, as an argument to both $\boldsymbol{O}$ and $\boldsymbol{B}_k$.)

Cumulativity

Theorem 13.3.6: *For any qfree formula* ψ, $\models (\boldsymbol{B}_k \exists \psi \supset \boldsymbol{B}_{k+1} \exists \psi)$.

Proof: This follows from the fact that $S^{k+1}(e) \subseteq S^k(e)$. ∎

Soundness and eventual completeness

The proof of soundness and eventual completeness uses the Herbrand Theorem and the following property of propositional Resolution.

Proposition 13.3.7: *Let C be a a set of ground clauses and d a non-tautologous ground clause. Then $C \cup \{\neg d\}$ is first-order unsatisfiable iff for some k and $d' \subseteq d$, $d' \in RP^k(C)$.*

As a special case of this proposition, we have the usual <u>*refutation completeness*</u> of Resolution: C is first-order unsatisfiable iff for some k, $[] \in RP^k(C)$.

To make the connection with believed sentences, we need to be able to convert a qfree formula into *CNF*:

Definition 13.3.8: Assume that ϕ is qfree and has been rewritten so that it does not use $\vee$, $\supset$, or $\equiv$. Then $CNF(\phi)$ is a finite set of clauses defined inductively by:

1. $CNF(\phi) = \{\{\phi\}\}$, when ϕ is a literal;
2. $CNF(\phi \wedge \psi) = CNF(\phi) \cup CNF(\psi)$;
3. $CNF(\neg\neg\phi) = CNF(\phi)$;
4. $CNF(\neg(\phi \wedge \psi)) = \{a \cup b \mid a \in CNF(\neg\phi), b \in CNF(\neg\psi)\}$.

Note that for any w and any qfree sentence ϕ, $w \models_{\mathrm{T}} \phi$ iff for every $b \in CNF(\phi)$, $w \models_{\mathrm{T}} b$.

We are now ready to prove the soundness and eventual completeness properties.

Lemma 13.3.9: *Let ψ be a qfree sentence, C a set of ground clauses, $e = \{w \mid w \models_{\mathrm{T}} C\}$. Then $C \cup \{\neg\psi\}$ is first-order unsatisfiable iff there is a k such that $e \models \boldsymbol{B}_k\psi$.*

Proof: ($\Rightarrow$) Suppose $C\cup\{\neg\psi\}$ is first-order unsatisfiable and let d be any non-tautologous clause in $CNF(\psi)$. Then $C \cup \{\neg d\}$ is first-order unsatisfiable and by Proposition 13.3.7, there is an i and a $d' \in RP^i(C)$ such that $d' \subseteq d$. So for any w such that $w \models_{\mathrm{T}} RP^i(C)$, $w \models_{\mathrm{T}} d$. By Lemma 13.3.5, $w \models_{\mathrm{T}} RP^i(C)$ iff $w \in S^i(e)$. So $e \models \boldsymbol{B}_i d$. Now let k be the maximum of these i values over all clauses of $CNF(\psi)$. Then $e \models \boldsymbol{B}_k\psi$.

($\Leftarrow$) Suppose $C \cup \{\neg\psi\}$ is first-order satisfiable. Then there is a world $w \in e$ such that $w \models_{\mathrm{T}} \neg\psi$ and $w[p] \neq *$ for every p, and so $w \not\models_{\mathrm{T}} \psi$. Then for every k, $w \in S^k(e)$ and therefore for every k, $e \not\models \boldsymbol{B}_k\psi$. ∎

Theorem 13.3.10: *Let ϕ and ψ be qfree formulas. Then $\models(\forall\phi \supset \exists\psi)$ iff there is a k such that $\models(\boldsymbol{O}\forall\phi \supset \boldsymbol{B}_k\exists\psi)$.*

Proof: Let $C = GND(CNF(\phi)) \cup GEQ$ and $e = \{w \mid w \models_{\mathrm{T}} C\}$. Note that $e \models \boldsymbol{O}\forall\phi$, and so it is sufficient to show that $\models(\forall\phi \supset \exists\psi)$ iff $e \models \boldsymbol{B}_k\exists\psi$. We have that $\models(\forall\phi \supset \exists\psi)$
iff $\{\forall\phi \wedge \forall\neg\psi\} \cup UNA \cup EQ$ is first-order unsatisfiable by Theorem 13.2.8
iff (by Proposition 13.1.1) there exist a k' and substitutions $\theta_0, \ldots, \theta_{k'}$ such that
$\{\forall\phi, \neg\psi\theta_0, \ldots, \neg\psi\theta_{k'}\} \cup GEQ$ is first-order unsatisfiable
iff $C \cup \{\neg\psi^*\}$ is first-order unsatisfiable, where $\psi^* = \bigvee\psi\theta_i$
iff (by Lemma 13.3.9) there is a k^*, such that $e \models \boldsymbol{B}_{k^*}\psi^*$
iff $e \models \boldsymbol{B}_k\exists\psi$, where k is the maximum of k' and k^*. ∎

Tractability

Looking over the details of the proof of soundness and eventual completeness above, we see the following:

$$\begin{aligned}\models(\boldsymbol{O}\forall\phi \supset \boldsymbol{B}_k\exists\psi) \text{ iff } & \text{ there are substitutions } \theta_0, \ldots, \theta_k \\ & \text{such that for all non-tautologous } d \in CNF(\psi\theta_0 \vee \cdots \vee \psi\theta_k) \\ & \quad \text{there is } d' \in RP^k(GND(CNF(\phi)) \cup GEQ) \text{ such that } d' \subseteq d.\end{aligned}$$

Now we want to show that under certain reasonable assumptions, it will be possible to efficiently decide if $\models(\boldsymbol{O}\forall\phi \supset \boldsymbol{B}_k\exists\psi)$. The problem with this is is that we cannot simply calculate $RP^k(GND(CNF(\phi)) \cup GEQ)$ since this is an infinite set of clauses. To get around this, we use a finite restriction of $UNA \cup EQ$ and we replace RP by RQ, the first-order version of Resolution that handles clauses with variables (defined below). We will end up

using something like this:

$$\begin{array}{l} \models (\boldsymbol{O}\forall\phi \supset \boldsymbol{B}_k\exists\psi) \text{ iff there are substitutions } \theta_0, \ldots, \theta_k \\ \quad \text{such that for all non-tautologous } d \in CNF(\psi\theta_0 \vee \cdots \vee \psi\theta_k) \\ \quad\quad \text{there is } d' \in RQ^k(CNF(\phi) \cup EQ') \text{ and a } \theta \text{ such that } d'\theta \subseteq d. \end{array}$$

where EQ' is $UNA \cup EQ$ restricted to the function and predicate symbols appearing in ϕ or ψ, with UNA restricted to a finite set of standard names. Here we will be able to calculate $RQ^k(CNF(\phi) \cup EQ')$, and the rest will involve guessing the appropriate substitutions. The precise definitions are as follows:

Definition 13.3.11: For any two literals ρ and τ, $MGU[\rho, \tau]$ is the set of most general unifiers of ρ and τ (empty if the two literals do not unify).

Definition 13.3.12: For any set C of clauses, $F(C)$ is union of $F(b)$ for all $b \in C$, where

$$F(b) = \{b\} \cup F(\{b\theta \mid \{\rho, \tau\} \subseteq b, \rho \neq \tau, \theta \in MGU[\rho, \tau]\}).$$

Definition 13.3.13: For any set C of clauses, $RQ(C)$ is defined by:

$$RQ(C) = C \cup \{(b \cup d)\theta \mid \{\rho\} \cup b \in F(C),\ \{\tau\} \cup d \in F(C),\ \theta \in MGU[\bar{\rho}, \tau]\}.$$

In the definitions of F and RQ, we assume the clauses in C use distinct variables, and that just one θ is chosen (if one exists) so that the new clauses also have distinct variables. The main property of this first-order Resolution is the following generalization of Proposition 13.3.7:

Proposition 13.3.14: *Let C be a a set of clauses and d a non-tautologous ground clause. Then $C \cup \{\neg d\}$ is first-order unsatisfiable iff for some k, θ, and $d' \in RQ^k(C)$, $d'\theta \subseteq d$.*

We now prove that calculating whether or not $\models (\boldsymbol{O}\forall\phi \supset \boldsymbol{B}_k\exists\psi)$ can be done efficiently under the following assumptions: the k is small, the query ψ is small, and if the KB ϕ itself is large, it is only because it is a large conjunction of sentences that are themselves small:

Theorem 13.3.15: *There is a procedure for deciding if $\models (\boldsymbol{O}[\phi_1 \wedge \cdots \wedge \phi_N] \supset \boldsymbol{B}_k\psi)$ that runs in time that is polynomial in N under the assumption that for some constant c, $k \leq c$, $|\phi_i| \leq c$, and $|\psi| \leq c$.*

Proof: Here is a sketch of the procedure. First, calculate $C_i = CNF(\phi_i)$. Then calculate EQ' from the given ϕ_i and ψ. Here the main complication is to limit the number of elements from *UNA* to a finite subset *UNA*$'$. It can be shown that *UNA*$'$ can be restricted to those elements from *UNA* which mention the names in ϕ and ψ plus $max\{2^k, (k+1)*|\psi|\}$ new names. (The size of *UNA*$'$ is polynomial since k and ψ are bounded.)

Having a finite EQ' in hand, we calculate $C = RQ^k(C_1 \cup \ldots \cup C_N \cup EQ')$, which will be polynomial since k is bounded. Next, guess at the $(k+1)$ substitutions θ_j and calculate $D = CNF(\bigvee \psi\theta_j)$. (Again, the k and ψ are bounded. The "guessing" of appropriate substitutions can be made determinate by trying all potential MGUs between terms in ψ and terms in $RQ^k(C \cup EQ')$, of which there are only polynomially many.)

Finally, check that for each non-tautologous $d \in D$, there is a $d' \in C$ such that $d'\theta \subseteq d$ for some θ. (This is a special case of what is called theta-subsumption.) ∎

13.4 Handling arbitrary objective beliefs

So far we have investigated the logic of belief for formulas $\boldsymbol{B}_k\exists\psi$ and $\boldsymbol{O}\forall\psi$ where ψ is a qfree formula. In this section, we want to generalize the model to deal with $\boldsymbol{B}_k\phi$ and $\boldsymbol{O}\phi$ where ϕ is any arbitrary objective sentence. To be more precise, we want to consider cases where ϕ is an arbitrary objective formula *that does not use Skolem symbols*. While we will be using Skolem constants and functions throughout this section, they will be playing an auxilliary role to support our handling of the Skolem-free formulas, which are the ones we care about. We assume that apart from the usual function symbols, we have an infinite supply of Skolem function symbols of every arity in the language.

The actual mathematics for carrying out Skolemization is complicated and messy, so we will present the material more informally. Here is the main idea. The logic stays the same except that we generalize the two rules for belief as follows:

5. $e, w \models \boldsymbol{B}_k\phi$ iff there is a dual-Skolemization ψ of ϕ and substitutions $\theta_0, \ldots, \theta_k$ such that for all w', if $w' \in S^k(e)$, then $w' \models_{\mathrm{T}} (\psi\theta_0 \vee \cdots \vee \psi\theta_k)$;
6. $e, w \models \boldsymbol{O}\phi$ iff there is a Skolemization ψ of ϕ such that for all w', $w' \in e$ iff $w' \models_{\mathrm{T}} GND(\psi) \cup GEQ$.

Note that this reduces to the definitions of the previous sections when the argument to $\boldsymbol{B}_k$ is of the form $\exists\phi$ and the argument to $\boldsymbol{O}$ is of the form $\forall\phi$ (where ϕ is qfree); no Skolemization or dual-Skolemization is then required.

But now consider a case where Skolemization is needed: only knowing an existential. Suppose $\boldsymbol{O}\exists x.P(x)$ is true. What we expect when $\exists x.P(x)$ is the only sentence in a knowledge base is that $\exists z.P(z)$ should be believed, but $P(t)$ should not be believed for any t.

This is indeed what we get from the rule above, but only when t is Skolem-free. There will be a Skolem constant a such that $\boldsymbol{O}P(a)$ is true and so $\boldsymbol{B}P(a)$ is true.

Similarly, consider a case of a belief with a universal quantifier. With the rule above, we have that $\boldsymbol{B}_k \forall x.P(x)$ is true iff $\boldsymbol{B}_k P(a)$ is true, where a is a Skolem constant. The assumption here is that the a is a *new* symbol, used nowhere else. In particular, if e is representable and $e \models \boldsymbol{O}\phi$, then a appears nowhere in ϕ (or in any Skolemization resulting from ϕ). So the only way this $\boldsymbol{B}_k P(a)$ can be true is if ϕ entails $P(t)$ for every term t, as when ϕ is something like $\forall z(P(z) \wedge Q(z))$.

Let us consider some of the properties of this generalized notion of belief. We can no longer use Theorem 13.2.10 to prove belief implications and equivalences since the beliefs can now involve quantifiers. However, there is a analogous theorem we can use.

Theorem 13.4.1: *Let ϕ and ψ be Skolem-free sentences. Suppose that for every dual-Skolemization ϕ' of ϕ there is a dual-Skolemization ψ' of ψ such that for every θ, $\phi'\theta \Rightarrow \psi'\theta$. Then $\models(\boldsymbol{B}_k\phi \supset \boldsymbol{B}_k\psi)$.*

Proof: Suppose $e \models \boldsymbol{B}_k\phi$. Then there is a dual-Skolemization ϕ' of ϕ and substitutions $\theta_0,\ldots,\theta_k$ such that $e \models \boldsymbol{B}_k\bigvee\phi'\theta_i$. Moreover, there is a dual-Skolemization ψ' of ψ such that for every θ, $\phi'\theta \Rightarrow \psi'\theta$. It follows that $\bigvee\phi'\theta_i \Rightarrow \bigvee\psi'\theta_i$. So by Theorem 13.2.10, $e \models \boldsymbol{B}_k\bigvee\psi'\theta_i$. Therefore $e \models \boldsymbol{B}_k\psi$. ∎

The corollaries of Theorem 13.2.10 now follow immediately. This again justifies the conversion to *CNF* within a belief. We can also justify the conversion to *prenex form*, where the quantifiers all appear at the front of a sentence with the following:

Theorem 13.4.2: *Let ϕ be a Skolem-free sentence and let ψ be its equivalent sentence in prenex form. Then $\models(\boldsymbol{B}_k\phi \equiv \boldsymbol{B}_k\psi)$.*

Proof: Every dual-Skolemization of ϕ is also a dual-Skolemization of ψ. ∎

We can also prove the failure of closure under conjunction mentioned in Section 13.2.4.

Theorem 13.4.3: *There is a k, a representable state e, and Skolem-free sentences ϕ and ψ such that such that $e \models \boldsymbol{B}_k\phi$, $e \models \boldsymbol{B}_k\psi$ but $e \not\models \boldsymbol{B}_k(\phi \wedge \psi)$.*

Proof: Let $e = \{w \mid w \models_{\mathrm{T}} \{(P(^{\#}1) \vee P(^{\#}2)), (Q(^{\#}1) \vee Q(^{\#}2))\} \cup GEQ\}$. Let $\phi = \exists x.P(x)$ and $\psi = \exists y.Q(y)$. Then $e \models \boldsymbol{B}_1\phi$, $e \models \boldsymbol{B}_1\psi$ but $e \not\models \boldsymbol{B}_1(\phi \wedge \psi)$. ∎

The problem here informally is that $\boldsymbol{B}_1\exists x.P(x)$ is true because of the first two disjuncts, and $\boldsymbol{B}_1\exists y.Q(y)$ is true because of the other two disjuncts, but for their conjunction, we would need to consider four disjuncts, for all combinations of the two variables.

Let us now return to the desiderata from Section 13.1. We can now consider the expressiveness desiderata. We need the following property of Skolemization and its dual:

Proposition 13.4.4: *Let ϕ be a Skolem-free sentence. For any Skolemization ψ of ϕ, there is a dual-Skolemization ψ' of ϕ and a unifying substitution θ^* such that $\psi\theta^* = \psi'\theta^*$.*

Theorem 13.4.5: *[Expressiveness] For any Skolem-free sentence ϕ, there is an epistemic state e such that $e \models \boldsymbol{O}\phi$, and moreover $\models(\boldsymbol{O}\phi \supset \boldsymbol{B}_0\phi)$.*

Proof: Clearly $\boldsymbol{O}\phi$ is satisfiable: Let $e = \{w \mid w \models_{\mathrm{T}} \mathit{GND}(\psi) \cup \mathit{GEQ}\}$, where ψ is a Skolemization of ϕ. Then $e \models \boldsymbol{O}\phi$. Now to show that $\models(\boldsymbol{O}\phi \supset \boldsymbol{B}_0\phi)$, suppose that $e \models \boldsymbol{O}\phi$. Then $e = \{w \mid w \models_{\mathrm{T}} \mathit{GND}(\psi) \cup \mathit{GEQ}\}$ for some Skolemization ψ of ϕ. Let ψ' and θ^* be as in Proposition 13.4.4 and suppose w is any element of e. Then $w \models_{\mathrm{T}} \psi\theta^*$ and so $w \models_{\mathrm{T}} \psi'\theta^*$. It follows that $e \models \boldsymbol{B}_0\psi'\theta^*$ and therefore $e \models \boldsymbol{B}_0\phi$. ∎

Turning now to the other desiderata, we clearly continue to have the property of cumulativity. As for soundness, eventual completeness, and tractability, these continue to hold because of the following key property of Skolemization:

Proposition 13.4.6: *Let ϕ be a Skolem-free sentence and ψ be any of its Skolemizations. Then $\{\phi\} \cup \mathit{UNA} \cup \mathit{EQ}$ is first-order unsatisfiable iff $\{\psi\} \cup \mathit{UNA} \cup \mathit{EQ}$ is first-order unsatisfiable.*

So to check whether $\models (\boldsymbol{O}\phi \supset \boldsymbol{B}_k\psi)$, it is sufficient to check if $\models (\boldsymbol{O}\forall\phi' \supset \boldsymbol{B}_k\exists\psi')$, where ϕ' is a Skolemized version of ϕ, and ψ' is a dual-Skolemized version of ψ (with distinct Skolem symbols, of course). We omit the remaining details.

One final issue to consider involves the interactions between quantifiers and belief operators. These are complicated by the fact that quantifiers outside of belief are interpreted in the normal way using standard names (as in previous chapters), whereas quantifiers inside of belief are interpreted by Skolemization and substitution of arbitrary terms. The existential case is easy:

Theorem 13.4.7: *For any k and any Skolem-free formula ϕ with one free variable x, the sentence $(\exists x\, \boldsymbol{B}_k\phi \supset \boldsymbol{B}_k\exists x\phi)$ is valid, but its converse is not valid.*

Proof: For the first part, suppose $e \models \exists x \boldsymbol{B}_k \phi$. Then for some n, $e \models \boldsymbol{B}_k \phi_n^x$ so that there is a dual-Skolemization ϕ' of ϕ_n^x and substitutions θ_i such that $e \models \boldsymbol{B}_k \bigvee \phi' \theta_i$. But then $\phi' = \psi_n^x$ where ψ is a dual-Skolemization of ϕ. So $e \models \boldsymbol{B}_k \exists x \phi$. For the second part, let $e = \{w \mid w \models_{\mathrm{T}} \{(P(^{\#}1) \vee P(^{\#}2))\} \cup GEQ\}$. Then $e \models \boldsymbol{B}_1 \exists x.P(x)$, but $e \not\models \exists x \boldsymbol{B}_1 P(x)$. ∎

For the universal case, let us begin with the non-closure property:

Theorem 13.4.8: *There is a Skolem-free formula ϕ with one free variable x such that $(\forall x \boldsymbol{B}_k \phi \supset \boldsymbol{B}_k \forall x \phi)$ is not valid.*

Proof: Let $e = \{w \mid w \models_{\mathrm{T}} GEQ\}$ and let $\phi = (x \neq {}^{\#}1 \vee x \neq {}^{\#}2)$. Then for every n, $e \models \boldsymbol{B}_0 \phi_n^x$, and so $e \models \forall x \boldsymbol{B}_0 \phi$. But for any constant a, $e \not\models \boldsymbol{B}_0 \phi_a^x$, and so $e \not\models \boldsymbol{B}_0 \forall x.\phi$. ∎

For the closure part, we need some new notation and two lemmas:

Definition 13.4.9: For any extended world w, any standard name n, and any constant a, let w_n^a be the extended world defined by $w_n^a[p] = w[p_n^a]$.

Lemma 13.4.10: *For any qfree sentence ϕ, $w_n^a \models_{\mathrm{T}} \phi$ iff $w \models_{\mathrm{T}} \phi_n^a$.*

Proof: The proof is by induction on the length of ϕ. ∎

Lemma 13.4.11: *Let e be an epistemic state such that $e \models \boldsymbol{O} \forall \phi$ for some qfree ϕ that does not mention constant a. Then for any qfree sentence ψ and any standard name n, if $e \models \boldsymbol{B}_k \psi$ then $e \models \boldsymbol{B}_k \psi_n^a$.*

Proof: Here we prove the case only for $k = 0$. First observe that we are using a here as a constant (nullary function), so that a does not appear in EQ. So $w \in e$ iff $w \models_{\mathrm{T}} UNA$ and for every θ, $w \models_{\mathrm{T}} EQ\theta$ and $w \models_{\mathrm{T}} \phi\theta$. Now assume that $e \models \boldsymbol{B}_0 \psi$. We will show that $e \models \boldsymbol{B}_0 \psi_n^a$ by showing that if $w \in e$ then $w \models_{\mathrm{T}} \psi_n^a$. So suppose that $w \in e$ and let θ be any substitution. Then (θ_n^a) is also a substitution and therefore $w \models_{\mathrm{T}} EQ(\theta_n^a)$ and $w \models_{\mathrm{T}} \phi(\theta_n^a)$. Since neither EQ nor ϕ mention a, $w \models_{\mathrm{T}} (EQ\theta)_n^a$ and $w \models_{\mathrm{T}} (\phi\theta)_n^a$. By Lemma 13.4.10, $w_n^a \models_{\mathrm{T}} EQ\theta$ and $w_n^a \models_{\mathrm{T}} \phi\theta$. Since this holds for any θ, and $w_n^a \models_{\mathrm{T}} UNA$, $w_n^a \in e$. Because $e \models \boldsymbol{B}_k \psi$, $w_n^a \models_{\mathrm{T}} \psi$ and therefore $w \models_{\mathrm{T}} \psi_n^a$ by Lemma 13.4.10. ∎

Theorem 13.4.12: *For any k and any Skolem-free formula ψ with one free variable x, the sentence $(\boldsymbol{B}_k \forall x.\psi \supset \forall x \boldsymbol{B}_k \psi)$ is valid.*

Proof: Assume $e \models \boldsymbol{B}_k \forall x.\psi$. Then there is a Skolem constant a such that $e \models \boldsymbol{B}_k \psi_a^x$. Let n be any standard name. Since a is a constant used nowhere else, by Lemma 13.4.11, $e \models \boldsymbol{B}_k (\psi_a^x)_n^a$, and so $e \models \boldsymbol{B}_k \psi_n^x$. Since this holds for any n, $e \models \forall x \boldsymbol{B}_k \psi$. ∎

13.5 Bibliographic notes

The notion of tractable reasoning developed in this chapter is the culmination of a long line of research that started with [130]. This early work used as semantic primitive a (possibly infinite) set of ground clauses called *setups* instead of a set of possible worlds. The idea was that these clauses, together with simple derivations such as weakening of clauses in a setup or conjunctions of clauses, formed the beliefs at level 0. In addition, setups were closed under unit propagation, which is resolution restricted to the case where one of the input clauses consists of a single literal, thus adding another simple form of inference at level 0. (Since unit propagation is in general undecidable, function symbols were not considered.) At higher levels additional beliefs were obtained by splitting cases. Believing ϕ at level $i \geq 0$ amounted to showing that for a given setup s there is a literal l such that ϕ is believed at level $i-1$ at both the setup $s \cup \{l\}$ and $s \cup \{\bar{l}\}$. For a certain class of knowledge bases called proper$^+$, which consisted essentially of first-order sentences without existential quantifiers, it was shown that reasoning was decidable using an evaluation procedure that was first introduced in [96] and which itself was inspired by a procedure for proper KBs discussed in Chapter 13. The complexity of reasoning in this framework was further studied in [132].

The idea of defining belief levels in terms of setups and case splitting was later refined to account for introspection [102], function symbols [105, 171], and actions [103]. A precursor of the approach in this chapter, again using clauses as semantic primitive, appeared in [106].

A possible-world approach to tractable reasoning that differs from ours was proposed in [73]. Here epistemic states are sets of three-valued worlds using a variant of neighborhood semantics [145, 173]. Belief levels are again defined in terms of splitting on literals, and tractability obtains at every level. However, the work is limited to the propositional case. Earlier work [112, 30, 153, 91] on tractable reasoning makes use of four-valued worlds, which are also the semantic basis of *tautological entailment* [37], a fragment of relevance logic [2]. While some of this work considers the first-order case, for example [153, 91], it remains limited as even simple inferences such as from p and $(p \supset q)$ infer q are ruled out.

Beginning with [26], there has also been work on tractable entailment relations of increasing complexity, again limited to a propositional language. Perhaps the most advanced such proposal is [25], which is based on a three-valued nondeterministic semantics first

considered in [24]. The author defines a k-consequence relation, which features splitting on arbitrary formulas and closure under unit propagation. The k-consequence relation is eventually complete and a proof-theoretic account is also provided.

13.6 Where do we go from here

The logic of belief proposed in this chapter is weaker than the more traditional epistemic logic considered in the rest of the book. But it does have a number of desirable properties, such as the desiderata from Section 13.1 and the equivalences noted in Corollary 13.2.12. One unexpected limitation is the failure of this notion of belief to be closed under conjunction and universal quantification:

- $\not\models (\boldsymbol{B}_k\phi \wedge \boldsymbol{B}_k\psi \supset \boldsymbol{B}_k(\phi \wedge \psi))$;
- $\not\models (\forall x\, \boldsymbol{B}_k\phi \supset \boldsymbol{B}_k \forall x \phi)$.

These two items are related. In fact, it is possible to show that there is a form of "eventual closure" that does hold:

- If $e \models (\boldsymbol{B}_k\phi \wedge \boldsymbol{B}_k\psi)$, then there is a $k' \geq k$ such that $e \models \boldsymbol{B}_{k'}(\phi \wedge \psi))$;
- If $e \models \forall x\, \boldsymbol{B}_k\phi$. then there is a $k' \geq k$ such that $e \models \boldsymbol{B}_{k'}\forall x \phi$;

For the example used in the proof of Theorem 13.4.3, we have $e \models \boldsymbol{B}_1\phi$ and $e \models \boldsymbol{B}_1\psi$ and $e \not\models \boldsymbol{B}_1(\phi \wedge \psi)$, but $e \models \boldsymbol{B}_3(\phi \wedge \psi)$.

Of course, the logic of belief presented here is not the only one that would satisfy the desiderata listed in Section 13.1. Assuming we can preserve tractability, we might want a version of $e \models \boldsymbol{B}_k\phi$ that holds for *fewer* ϕ (as long as we do not lose eventual completeness) but is easier to compute, or we might want a version that holds for *additional* ϕ (as long as we do not lose soundness) even if it were somewhat harder to compute. To see one way these two options might work out, let us consider the quantifier-free version of the logic and the close correspondence between S^k and RP^k as given by Lemma 13.3.5.

For the first option, if the KB is in clausal form and the query ϕ is a non-tautologous clause, then the k in $\boldsymbol{B}_k\phi$ can be thought of as an upper bound on the *depth* of a Resolution proof (of a subclause of ϕ), taken as a tree. This is because RP is performing all its Resolution steps in parallel. It may be more practical to have the k be an upper bound on the *size* of a Resolution proof. This would push belief in clauses up to higher values of k, thus making the lower values of k easier to compute. To achieve this effect, we can replace RP^k by a set of clauses RP_k defined as follows:

Definition 13.6.1: For any set of ground clauses C, the sets of clauses $RP_i(C)$ are defined inductively by:

1. $RP_0(C) = C$;
2. $RP_{i+1}(C) = RP_i(C) \cup$
   $\{(b \cup d) \mid \text{for some } p, \{p\} \cup b \in RP_r(C),\ \{\neg p\} \cup d \in RP_s(C),\ r + s = i\}$.

Of course we would need a new definition of sets of worlds S_k to keep the correspondence with RP_k (analogous to Lemma 13.3.5). With this in place, however, it should be possible to prove eventual completeness as before.

Regarding the second option, note that if KB $= \{\{p\}, \{\neg p, q\}, \{\neg q, r\}, \{\neg r, s\}\}$, we get that $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_k s)$ only for $k \geq 3$. We might prefer to have this easy form of *linear* reasoning separate from the more general application of Resolution (which is more like the difficult process of splitting cases). In the propositional case, this can be achieved by replacing RP^k by a set of clauses RP_k now defined as follows:

Definition 13.6.2: For any set of clauses C, the sets of clauses $RP_k(C)$ are defined as $UP(RP^k(C))$, where $UP(C)$ is the least set of clauses C' such that $C \subseteq C'$ and if $\{\rho\} \in C'$ and $(\{\bar{\rho}\} \cup b) \in C'$, then $b \in C'$.

The UP operation here is what does the linear reasoning. We would again need a new definition of sets of worlds S_k to keep the correspondence with this version of RP_k. With this in place, for the example above, we would get that $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_0 s)$

Note that UP is a special case of Resolution (with unit clauses) and so is guaranteed to preserve soundness. In the propositional case, it is also tractable: there is a linear time procedure for deciding if a clause is in $UP(C)$. In the first-order case, however, the unrestricted application of Resolution to unit clauses is undecidable. (The Prolog programming language is based on Resolution with unit clauses, and its halting problem is undecidable.) This means we need a more restricted definition of UP.

One possibility is to restrict UP so that it only applies to literals ρ whose arguments are variables or standard names. This would be enough to preserve tractability and still allow linear chains of reasoning with explicitly named individuals. For example, if we have KB $= \{\{p(^{\#}3)\}, \{\neg p(x) \vee q(x)\}, \{\neg q(x), p(f(x))\}\}$, then we would get as a result $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_0\, q(^{\#}3))$ and $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_0\, p(f(^{\#}3)))$, but not $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_0\, q(f(^{\#}3)))$. Another possibility is to allow terms in ρ that use function symbols, but only apply the unit Resolution if the depth of *nesting* of functions in the resulting clause is no higher than what it was in ρ. This would allow certain linear chains of reasoning even with individuals that are not identified. For example, if KB $= \{\{p(f(a))\}, \{\neg p(x), q(x)\}\}$, then we would get $\models (\boldsymbol{O}\text{KB} \supset \boldsymbol{B}_0\, q(f(a)))$. Allowing the terms to get arbitrarily nested is what appears to lead to undecidability.

13.7 Exercises

1. Prove $\models \boldsymbol{O}\exists x \forall y\, P(x, y) \supset \boldsymbol{B}_k \forall y \exists x\, P(x, y)$ but $\not\models \boldsymbol{O}\forall y \exists x\, P(x, y) \supset \boldsymbol{B}_k \exists x \forall y\, P(x, y)$.
2. Prove that if ϕ and ψ are quantifier-free, then $\models \boldsymbol{B}_k(\phi \wedge \psi) \equiv (\boldsymbol{B}_k\phi \wedge \boldsymbol{B}_k\psi)$.
3. Prove that if $k = 0$ then $\models \boldsymbol{B}_k(\phi \wedge \psi) \equiv (\boldsymbol{B}_k\phi \wedge \boldsymbol{B}_k\psi)$.
4. Prove that for any qfree sentences ϕ and ψ, if $\boldsymbol{B}_0\phi \supset \boldsymbol{B}_0\psi$ then $\phi \Rightarrow \psi$. (This is the converse to Theorem 13.2.10.)
5. Define nested belief by using a modified version of RES to reduce it to objective belief.
6. Define nested belief in the more advanced way so that $\boldsymbol{B}_2(...\boldsymbol{B}_5...)$ does not require determining level 5 beliefs.

14 Knowledge and Action

In previous chapters, we considered representing and reasoning with knowledge, where an epistemic state was characterized as a set of world states. Although we dealt with an uncountably infinite set of world states, these were intended as models of the different ways the world might be imagined to be at some point in time. The epistemic state of the system might change of course, as the result of a **TELL** operation, for example, but the underlying world state (and which sentences were actually true or false) was taken to be unchanging. What we never considered, in other words, was the possibility that the world itself might also be changing from one state to another. In this chapter, we want to consider what it would mean to represent knowledge about a changing world and, in particular, one that changes as the result of actions we might also have some separate knowledge about.

Consider a robot operating in the world. When a robot performs the action of moving, for example, this causes its location (and that of any object it is carrying) to change in the world itself. Obviously such actions should also affect what a robot knows about the world: after moving, the robot should know that its current location is no longer what it was. Here we have an example of a robot acquiring new beliefs not as a result of a **TELL** operation (at least not directly), but as a result of performing an action. We will also see that there are sensing actions, whose effect is not to change the world, but only to change what the system knows about the world. For example, a robot might perform the action of looking inside a room and thereby find out if there is anything inside.

To be able to describe how knowledge changes as the result of actions in the world, we need to first be clear about how the world itself changes as the result of those actions. One popular way of representing actions and their effects is to use the language of the *situation calculus*. What we will present in Section 14.1 is a new language called $\mathcal{ES}$ which integrates portions of the situation calculus with $\mathcal{OL}$. This gives us a language for talking about action, as well as knowing and only-knowing. We illustrate the properties of $\mathcal{ES}$ with a simple robotic example in Section 14.2. In Section 14.3, we discuss the general principles that allow us to determine what is known after performing an action (including sensing actions) in terms of what was true before. Ultimately, what is known after a sequence of actions will reduce to some function of what was known initially.

14.1 The language $\mathcal{ES}$

The language $\mathcal{ES}$ is a generalization of the language $\mathcal{OL}$ that includes additional facilities for talking about actions and their *effects*.[1] Specifically, we assume three main additions:

objects vs. actions There are two sorts of terms in $\mathcal{ES}$: ordinary objects and actions. The object terms are the variables, standard names, and function applications, as before. We will continue to use $\mathcal{N}$ for the standard names of objects. For actions, we start with an infinite collection of symbols called <u>action types</u> each of which has an arity (just like a function or predicate). For simplicity we assume that the arguments of an action type are of type object. The action terms are either action variables or of the form $A(t_1, \ldots, t_k)$ where A is an action type, the t_i are object terms, and the k is the arity of A. (As usual, we leave out the parentheses when $k = 0$.) The standard names for actions are the terms $A(t_1, \ldots, t_k)$ where the t_i are standard object names.

dynamic formulas In addition to the formulas of $\mathcal{OL}$, we assume there are two new special formulas in $\mathcal{ES}$. If t is a term of sort action and α is a formula, then $[t]\alpha$ is a formula that can be read as "after action t, α is true." Similarly, the expression $\Box\alpha$ is a formula that can be read as "after every sequence actions, α is true".

binary sensing Each action gets to return a binary sensing result after it is executed. There is a distinguished unary predicate SF which takes an action as argument and where $SF(t)$ can be read as "action t returns a binary sensing result of 1," so that $\neg SF(t)$ can be read as "action t returns a binary sensing result of 0."[2]

To give a preview of how the dynamic formulas of $\mathcal{ES}$ can be used, consider the following sentence (where *cup*5 is a constant, *Broken* is a predicate,[3] and *drop* is an action type):

$$\neg Broken(cup5) \ \wedge \ [drop(cup5)]\, Broken(cup5).$$

This says that the object *cup*5 is not broken (currently) but that it will be broken after the action of dropping it. To draw conclusions about what will or will not hold as actions take place, a sentence like the following might be believed:

$$\begin{aligned} \forall a \forall x.\Box([a]Broken(x) \ \equiv \\ (a = drop(x) \wedge Fragile(x)) \ \vee \\ (Broken(x) \wedge a \neq repair(x))) \end{aligned}$$

1 It is common to also want to talk about the *preconditions* of actions, that is, the conditions under which an action can be executed. These present no special problems and for simplicity, we omit them here. See Section 14.4 for how these can be incorporated into $\mathcal{ES}$.

2 For simplicity, we are assuming that sensing involves obtaining a *binary* reading only from the surrounding environment. We leave the more general case of sensing results as an exercise.

3 To emphasize that functions or predicates may be changed as the result of actions, they are sometimes called *fluents* in the context of $\mathcal{ES}$.

In English: after every sequence of actions, an object x will be broken after doing action a iff a is the dropping of x when x is fragile or x was already broken and a is not the action of repairing it. Sentences like this are called *successor state axioms* (SSAs), as they describe precisely how a predicate or function changes in the successor state after an action. For predicates that never change, we might have an SSA like this:

$$\forall a \forall x . \Box([a]Fragile(x) \equiv Fragile(x))$$

This says that whether or not an object is fragile is unaffected by any action. As we will see later, SSAs play an important role in specifying the dynamics of a domain.

To see how sensing can be used in $\mathcal{ES}$, consider the following sentence (where *examine* is an action type):

$$\neg \boldsymbol{K} Broken(cup5) \wedge \neg \boldsymbol{K} \neg Broken(cup5) \wedge [examine(cup5)]\, \boldsymbol{K} \neg Broken(cup5).$$

This says that initially it is not known whether or not *cup*5 is broken, but that after examining it, it is then known to be unbroken. Note that the *examine* action does not cause the cup to be unbroken, like the *repair* action mentioned above; instead of changing the world state, it changes the epistemic state to one where the true state of the cup is known. The connection between properties (like *Broken*) and sensing actions (like *examine*) is formalized in what are called *sensed fluent axioms*), described later in Section 14.2.

In what follows, we will use the following terminology: a formula with no $\Box$ operators is called *bounded;* a formula with no $\Box$ or $[t]$ operators is called *static;* a formula with no $\boldsymbol{K}$, $\boldsymbol{O}$, $\Box$, $[t]$, or *SF* is called a *fluent* formula; a formula with no $\boldsymbol{K}$, or $\boldsymbol{O}$ is called an *objective* formula; a formula where every function, predicate, $\Box$, and $[t]$ occurs within the scope of a $\boldsymbol{K}$ or $\boldsymbol{O}$ is called a *subjective* formula.

14.1.1 The semantics

As we saw above, a sentence can say that something holds at one point in time but fails to hold at another. So the semantics needs to specify not only what is true *initially*, but what is true after any sequence of actions. We will use the notation $e, w, z \models \alpha$ (with the additional argument z) to mean that α is true after the sequence of actions z, given an initial epistemic state e and initial world state w.

More precisely, let $\mathcal{Z}$ be the set of all finite sequences of standard action names, including $\langle\,\rangle$, the empty sequence. Then

- a world $w \in W$ is any function from the primitive sentences and $\mathcal{Z}$ to $\{0, 1\}$, and from the primitive object terms and $\mathcal{Z}$ to standard names of objects;
- an epistemic state $e \subseteq W$ is any set of worlds.

Note that the worlds and epistemic states of $\mathcal{OL}$ can be thought of as special cases of their counterparts in $\mathcal{ES}$ by ignoring actions and all action sequences other than the empty one.

The idea of coreferring standard names in $\mathcal{ES}$ works almost exactly the same as in $\mathcal{L}$ except that we need to take into account both a world and a sequence of actions: given any term t without variables, a world w, and an action sequence z, we define $|t|^z_w$ (read: the coreferring standard name for t given w and z) inductively by:

1. If $t \in \mathcal{N}$, then $|t|^z_w = t$;
2. When h is a function, $|h(t_1, \ldots, t_k)|^z_w = w[h(n_1, \ldots, n_k), z]$, where $n_i = |t_i|^z_w$;
3. When A is an action type, $|A(t_1, \ldots, t_k)|^z_w = A(n_1, \ldots, n_k)$, where $n_i = |t_i|^z_w$.

To interpret what is known or only-known after a sequence of actions has taken place, we define $w' \simeq_z w$ (read: w' agrees with w agree on the sensing throughout action sequence z) inductively by the following:

1. $w' \simeq_{\langle\rangle} w$ for all w and w';
2. $w' \simeq_{z \cdot n} w$ iff $w' \simeq_z w$ and $w'[SF(n), z] = w[SF(n), z]$.

Note that $\simeq_z$ is an equivalence relation and will be used in the specification of $\boldsymbol{K}\alpha$ and $\boldsymbol{O}\alpha$. (This is the only place in the semantics where the *SF* predicate is used.)

Putting all these parts together, here is the semantic definition of truth. Given a sentence α of $\mathcal{ES}$, an epistemic state $e \subseteq W$ and a world $w \in W$, we define $e, w \models \alpha$ (read: α is true at e and w) as $e, w, \langle\rangle \models \alpha$, where for any $z \in \mathcal{Z}$ we have:

1. $e, w, z \models P(t_1, \ldots, t_k)$ iff $w[P(n_1, \ldots, n_k), z] = 1$, where $n_i = |t_i|^z_w$;
2. $e, w, z \models (t_1 = t_2)$ iff n_1 and n_2 are identical, where $n_i = |t_i|^z_w$;
3. $e, w, z \models \neg\alpha$ iff $e, w, z \not\models \alpha$;
4. $e, w, z \models (\alpha \vee \beta)$ iff $e, w, z \models \alpha$ or $e, w, z \models \beta$;
5. $e, w, z \models \exists x \alpha$ iff $e, w, z \models \alpha^x_n$, for some std. name n of the right sort for x;
6. $e, w, z \models [t]\alpha$ iff $e, w, z \cdot n \models \alpha$, where $n = |t|^z_w$;
7. $e, w, z \models \Box\alpha$ iff $e, w, z \cdot z' \models \alpha$, for every $z' \in \mathcal{Z}$;
8. $e, w, z \models \boldsymbol{K}\alpha$ iff for every w' such that $w' \simeq_z w$, if $w' \in e$ then $e, w', z \models \alpha$;
9. $e, w, z \models \boldsymbol{O}\alpha$ iff for every w' such that $w' \simeq_z w$, $w' \in e$ iff $e, w', z \models \alpha$.

As before, when α is *objective*, we can leave out the e and write $w \models \alpha$. Similarly, when α is *subjective*, we can leave out the w and write $e \models \alpha$.

A set of sentences Γ is said to be satisfiable in $\mathcal{ES}$ iff for some world w and epistemic state e, we have that $e, w \models \alpha$ for all $\alpha \in \Gamma$. The notions of logical implication and validity in $\mathcal{ES}$ are then defined in the usual way.

14.1.2 Properties of $\mathcal{ES}$

It is easy to see that $\mathcal{ES}$, when restricted to static sentences not mentioning action terms, is exactly the same as $\mathcal{OL}$, since the semantic rules of $\mathcal{ES}$ are essentially the same as those of $\mathcal{OL}$ for the static fragment of the language.

Theorem 14.1.1: *Let α be a static sentence that does not mention action terms. Then α is valid in $\mathcal{ES}$ iff α is valid in $\mathcal{OL}$.*

The proof is left as an exercise.

Let us now consider the dynamic aspects of the new logic, beginning with the operator $[\cdot]$. The first property of the following theorem says that $[\cdot]$ is closed under Modus ponens. The others result from the fact that the effects of an action are deterministic, allowing us to freely move Boolean connectives and quantifiers in and out of $[\cdot]$.

Theorem 14.1.2: *Let n be an action standard name and α and β arbitrary sentences and γ a formula with at most one free variable x.*

1. $\models [n]\alpha \wedge [n](\alpha \supset \beta) \supset [n]\beta$;
2. $\models [n]\neg\alpha \equiv \neg[n]\alpha)$;
3. $\models [n](\alpha \vee \beta) \equiv ([n]\alpha \vee [n]\beta)$;
4. $\models [n]\exists x\gamma \equiv \exists x[n]\gamma$.

Proof:

1. Let $e, w \models [n]\alpha \wedge [n](\alpha \supset \beta)$ for any epistemic state e, world w. Then $e, w, n \models \alpha \wedge (\alpha \supset \beta)$. Hence $e, w, n \models \beta$ and thus $e, w \models [n]\beta$.
2. $e, w \models [n]\neg\alpha$ iff $e, w, n \models \neg\alpha$ iff $e, w, n \not\models \alpha$ iff $e, w \not\models [n]\alpha$ iff $e, w \models \neg[n]\alpha$.
3. $e, w \models [n](\alpha \vee \beta)$ iff $e, w, n \models (\alpha \vee \beta)$ iff $e, w, n \models \alpha$ or $e, w, n \models \beta$ iff $e, w \models [n]\alpha \vee [n]\beta$.
4. $e, w \models \exists x[n]\gamma$ iff $e, w \models [n]\gamma^x_m$ for some name m iff $e, w, n \models \gamma^x_m$ for some name m iff $e, w, n \models \exists x\gamma$ iff $e, w \models [n]\exists x\gamma$. ∎

For $\Box$ we get the following properties:

Theorem 14.1.3:

1. $\models \Box\alpha \wedge \Box(\alpha \supset \beta) \supset \Box\beta$;
2. $\models \Box\alpha \supset \alpha$;
3. $\models \Box(\alpha \wedge \beta) \equiv \Box\alpha \wedge \Box\beta$;

4. $\models \forall x \Box \alpha \equiv \Box \forall x \alpha$;
5. $\models \exists x \Box \alpha \supset \Box \exists x \alpha$;
6. $\models \Box \alpha \supset \Box\Box \alpha$;
7. $\models \Box \alpha \equiv \alpha \wedge \forall a [a] \Box \alpha$;
8. $\models \alpha \wedge \Box(\alpha \supset \forall a [a] \alpha) \supset \Box \alpha$.

Proof:

1. Let $e, w \models \Box \alpha \wedge \Box(\alpha \supset \beta)$. Then $e, w, z \models \alpha \wedge (\alpha \supset \beta)$ for an arbitrary action sequence z. Hence $e, w, z \models \beta$ and thus $e, w \models \Box \beta$.
2. Let $e, w \models \Box \alpha$. Thus, in particular, $e, w, \langle\,\rangle \models \alpha$ and we are done.
3. $e, w \models \Box(\alpha \wedge \beta)$ iff $e, w, z \models (\alpha \wedge \beta)$ for all z iff $e, w, z \models \alpha$ for all z and $e, w, z \models \beta$ for all z iff $e, w \models \Box \alpha$ and $e, w \models \Box \beta$ iff $e, w \models \Box \alpha \wedge \Box \beta$.
4. $e, w \models \forall x \Box \alpha$ iff $e, w, z \models \alpha^x_n$ for all z and all n iff $e, w \models \Box \forall x \alpha$.
5. Let $e, w \models \exists x \Box \alpha$. Then for some n and for all z, $e, w, z \models \alpha^x_n$. Therefore, for all z there is an n such that $e, w, z \models \alpha^x_n$ and, hence, $e, w \models \Box \exists x \alpha$.
6. Let $e, w \models \Box \alpha$, that is, $e, w, z \models \alpha$ for all z. But then for all z and all z', $e, w, z \cdot z' \models \alpha$. Therefore $e, w, z \models \Box \alpha$ for all z and hence $e, w \models \Box\Box \alpha$.
7. This property, known as *Iteration* in Dynamic Logic, easily follows from the definition of $\Box$.
8. This property, also known as the *Induction Axiom* in Dynamic Logic, is left as an exercise.

It is not hard to show that the converse of (5.) does not hold in general. Consider a world w with the following property for unary fluent P: if $|z|$ is even then $w[P(n), z] = 1$ iff $n = {}^{\#}1$, and if $|z|$ is odd then $w[P(n), z] = 1$ iff $n = {}^{\#}2$. Then clearly, $w \models \Box \exists x P(x)$ (choose either ${}^{\#}1$ or ${}^{\#}2$), but there is no single name that works for all z. Hence $w \not\models \exists x \Box P(x)$.

While property (6.) of the above theorem is the analogue of *positive introspection,* an analogue of negative introspection does not hold for $\Box$: let p be a primitive atom and let w be a world such that $w[p, \langle\,\rangle] = 0$ and $w[p, z] = 1$ for all $z \neq \langle\,\rangle$. Then clearly $w \models \neg \Box p$ since $w, \langle\,\rangle \models \neg p$, yet for all $z \neq \langle\,\rangle$ we have $w, z \models \Box p$. Therefore, $\not\models \neg \Box p \supset \Box \neg \Box p$.

KNOWLEDGE

The interpretation of knowledge in $\mathcal{ES}$ is quite similar to $\mathcal{KL}$. One subtle difference worth noting is that we do not simply require truth in all elements of e, the given set of "possible worlds." In fact, e represents the *initial* state of knowledge, and as knowledge is acquired through action, some of those initial worlds will no longer be considered possible. This is

reflected in the $\simeq_z$ relation. In a nutshell, we look for truth in all elements of e that agree with the real world w in terms of sensing. It will then follow that after doing a sequence of actions, the agent will know the correct values of the sensing results in the real world (and everything it can conclude from that).

Apart from this it is not hard to see that knowledge in $\mathcal{ES}$ inherits all the properties from $\mathcal{KL}$, not just initially, but after any sequence of actions.

Theorem 14.1.4:

1. $\models \Box(\boldsymbol{K}\alpha \wedge \boldsymbol{K}(\alpha \supset \beta) \supset \boldsymbol{K}\beta)$;
2. $\models \Box(\boldsymbol{K}\alpha \supset \boldsymbol{K}\boldsymbol{K}\alpha)$;
3. $\models \Box(\neg\boldsymbol{K}\alpha \supset \boldsymbol{K}\neg\boldsymbol{K}\alpha)$;
4. $\models \Box(\forall x.\boldsymbol{K}\alpha \supset \boldsymbol{K}\forall x.\alpha)$;
5. $\models \Box(\exists x.\boldsymbol{K}\alpha \supset \boldsymbol{K}\exists x.\alpha)$.

Proof:

1. Let $e, w, z \models \boldsymbol{K}\alpha \wedge \boldsymbol{K}(\alpha \supset \beta)$. Then for all $w' \simeq_z w$, if $w' \in e$ then $e, w', z \models \alpha$ and $e, w', z \models (\alpha \supset \beta)$. Hence, $e, w', z \models \beta$ and, therefore, we have that $e, w, z \models \boldsymbol{K}\beta$.
2. Let $e, w, z \models \boldsymbol{K}\alpha$. Let w' and w'' be worlds in e such that $w' \simeq_z w$ and $w'' \simeq_z w'$. Since $\simeq_z$ is an equivalence relation, we have $w'' \simeq_z w$ and, therefore, $e, w'', z \models \alpha$ by assumption. As this is true for all $w'' \in e$ with $w'' \simeq_z w'$, we have $e, w', z \models \boldsymbol{K}\alpha$ and, hence, $e, w, z \models \boldsymbol{K}\boldsymbol{K}\alpha$.
3. Let $e, w, z \models \neg\boldsymbol{K}\alpha$. Thus for some w', $w' \simeq_z w$, $w' \in e$ and $e, w', z \not\models \alpha$. Let w'' be any world such that $w'' \simeq_z w'$ and $w'' \in e$. Clearly, $e, w'', z \models \neg\boldsymbol{K}\alpha$. Since $w'' \simeq_z w$, $e, w, z \models \boldsymbol{K}\neg\boldsymbol{K}\alpha$ follows.
4. Let $e, w, z \models \forall x.\boldsymbol{K}\alpha$. Hence for all names n of the right sort, $e, w, z \models \boldsymbol{K}\alpha_n^x$ and thus for all $w' \simeq_z w$, if $w' \in e$ then for all names n of the right sort, $e, w, z \models \alpha_n^x$, from which $e, w, z \models \boldsymbol{K}(\forall x.\alpha)$ follows.
5. Let $e, w, z \models \exists x.\boldsymbol{K}\alpha$. Then $e, w, z \models \boldsymbol{K}\alpha_n^x$ for some name n. By the definition of $\boldsymbol{K}$, it follows that $e, w, z \models \boldsymbol{K}\exists x.\alpha$. ∎

The fact that the properties of $\boldsymbol{K}$ hold after any sequence of actions is no coincidence. It can be shown, with some effort, that any valid sentence of $\mathcal{ES}$ remains valid if we put a $\Box$ in front of it.

Lemma 14.1.5: *If for all* $e, w,\ \ e, w, \langle\,\rangle \models \alpha$ *then for all* $e, w, z,\ \ e, w, z \models \alpha$.

The proof is left as an exercise.

Theorem 14.1.6: *If* $\models \alpha$ *then* $\models \Box\alpha$.

Proof: Assume $\models \alpha$. Then by the lemma, for all $e, w, z,\ \ e, w, z \models \alpha$. Therefore, for all $e, w, z, z',\ \ e, w, z \cdot z' \models \alpha$. So for all $e, w, z,\ \ e, w, z \models \Box\alpha$. ∎

In the following sections we will make use of only-knowing, but only in a very special way to characterize what an agent in a dynamic world knows initially, before any actions have occurred.

14.2 Basic action theories

Let us now turn to the more pragmatic issue of how to use the logic to model an agent's knowledge about a dynamic world.

Definition 14.2.1: Given a set of fluents $\mathcal{F}$, a set $\Sigma \subseteq \mathcal{ES}$ of sentences is called a *basic action theory* over $\mathcal{F}$ iff $\Sigma = \Sigma_0 \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}$ where Σ mentions only fluents in $\mathcal{F}$ and

1. Σ_0 is any set of fluent sentences not mentioning any actions;
2. Σ_{post} is a set of sentences of the form
   $$\forall a \forall \vec{x}.\Box[a]F(\vec{x}) \equiv \gamma_F \ \text{ or } \ \forall a \forall y \forall \vec{x}.\Box[a]f(\vec{x}) = y \equiv \gamma_f,$$
   one for each relational fluent F and functional fluent f, respectively, and where γ_F and γ_f are fluent formulas.[4]
3. Σ_{sense} is a singleton sentence of the form $\forall a.\Box SF(a) \equiv \varphi$, where φ is a fluent formula.

The idea here is that Σ_0 expresses what is true initially (in the initial situation) and Σ_{post} is a set of successor state axioms, one per fluent. Σ_{sense} then captures the outcome of sensing actions. For actions like $drop(x)$, which do not return any useful sensing information, SF can be defined to be vacuously true (see below for an example).

Since an agent's beliefs may differ from what is true, we will, in general, need two basic action theories: Σ for what is true in the world, including its dynamics, and Σ' for what the agent believes to be true. The two are allowed to differ arbitrarily and even contradict each other to allow for false beliefs. A state of affairs can then be characterized by sentences of

4 We assume that $\Box$ has lower syntactic precedence than the logical connectives and $[t]$ has higher precedence. So $\Box[a]F(\vec{x}) \equiv \gamma_F$ abbreviates $\Box(([a]F(\vec{x})) \equiv \gamma_F)$.

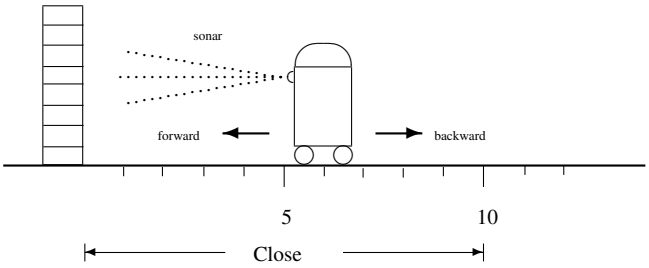

Figure 14.1: A simple robot

the form Σ to denote what is actually true, and $\boldsymbol{O}\Sigma'$ to denote what the agent only-believes to be true.[5] We will be interested in the what is entailed by such theories.

As an example, imagine a robot that lives in a 1-dimensional world, and that can move towards or away from a fixed wall. The robot also has a sonar sensor that tells it when it gets close to the wall, say, less than 10 units away. See Figure 14.1. So we might imagine three actions, *forward* and *backward* which move the robot one unit towards and away from the wall, and a *sonar* sensing action which tells the robot if it is close to the wall but has no effect on the world. We have a single fluent, *distance*, which gives the actual distance from the robot to the wall.[6]

Let us consider informally how sensing relates knowledge to truth here. We start in some initial epistemic state e and world w. Initially, before any actions have taken place, the action sequence z is $\langle\rangle$. We might suppose that $w[\mathit{distance}, \langle\rangle] = 6$ as in the diagram, that is, $e, w, \langle\rangle \models (\mathit{distance} < 10)$. If the robot does not know where it is, there may be a $w^* \in e$ where $w^*[\mathit{distance}, \langle\rangle] = 13$ and hence $e, w, \langle\rangle \models \neg\boldsymbol{K}(\mathit{distance} < 10)$. Now suppose the robot performs a *sonar* action. In this case, we would expect that $w[\mathit{SF}(\mathit{sonar}), \langle\rangle] = 1$, but $w^*[\mathit{SF}(\mathit{sonar}), \langle\rangle] = 0$. In other words, if the sonar is doing its job, in w it would tell us that the robot is close to the wall and in w^* it would tell us that the robot is far from the wall. So if we now let $z = \langle \mathit{sonar} \rangle$, we see that $w^* \not\simeq_z w$, since they disagree on the *SF* value. In fact, for every w' such that $w' \simeq_z w$, we will have that $w'[\mathit{SF}(\mathit{sonar}), \langle\rangle] = 1$. Since the definition of $\boldsymbol{K}$ uses $\simeq$, when we consider what is known after doing the *sonar* action, the robot will believe (correctly) that it is close to the

5 As usual, when we use Σ as part of a sentence we mean the conjunction of all the finitely many sentences contained in Σ.

6 Here and below, we assume that simple arithmetic involving $<$, $+$, and $-$ is given to us for free.

wall: $e, w, \langle sonar \rangle \models \boldsymbol{K}(distance < 10)$.

Let us now make all this precise. We begin our formalization by defining the sensing results for the actions:

$$\begin{aligned}
&\forall a \Box\, SF(a) \;\equiv \\
&\quad a = forward \;\wedge\; \text{TRUE} \quad \vee \\
&\quad a = backward \;\wedge\; \text{TRUE} \quad \vee \\
&\quad a = sonar \;\wedge\; distance < 10.
\end{aligned}$$

Since *backward* and *forward* are not expected to return any useful sensing information, *SF* is vacuously true for them, while *SF*(*sonar*) says that the sonar returns 1 precisely when the distance to the wall is less than 10. All that is left to do is defining a successor state axiom for our only fluent:

$$\begin{aligned}
&\forall a \forall x . \Box\, [a](distance = x) \;\equiv \\
&\quad a = forward \;\wedge\; distance = x \;\wedge\; x = 0 \quad \vee \\
&\quad a = forward \;\wedge\; distance = x + 1 \quad \vee \\
&\quad a = backward \;\wedge\; distance = x - 1 \quad \vee \\
&\quad a \neq forward \;\wedge\; a \neq backward \;\wedge\; distance = x .
\end{aligned}$$

In other words, the distance to the wall increases or decreases by 1 depending on whether *backward* or *forward* is executed, or it remains as before for all other actions.

Now we are ready to consider some specifics having to do with what is true initially by defining an action theory. Let *Close* stand for the formula "$distance < 10$." Let ϕ denote the conjunction of the sentences above. We assume that ϕ is true and the robot knows it. We also assume the robot is located initially 6 units away from the wall, but that the robot has no idea where it is. So, we let $\Sigma = \{\phi\} \cup \{distance = 6\}$ and $\Sigma' = \{\phi\}$. Then we get this:

Example 14.2.2: The following are logical entailments of

$\Sigma \wedge \boldsymbol{O}\Sigma'$:

1. $Close \;\wedge\; \neg \boldsymbol{K} Close \;\wedge\; [forward] \neg \boldsymbol{K} Close$
   the robot is close to the wall, but does not know it, and continues not to know it after moving forward;
2. $[sonar]\,(\boldsymbol{K} Close \;\wedge\; [forward] \boldsymbol{K} Close)$
   after reading the sonar, the robot knows it is close, and continues to know it after moving forward;
3. $[sonar]\,[backward] \neg \boldsymbol{K} Close$
   after reading the sonar and then moving backward, the robot no longer knows that it is close to the wall;

4. $[\mathit{backward}]\,[\mathit{sonar}]\,\boldsymbol{K}\mathit{Close}$
   after moving backward and then reading the sonar, the robot knows that it is close to the wall;
5. $[\mathit{sonar}]\,[\mathit{forward}]\,[\mathit{backward}]\,\boldsymbol{K}\mathit{Close}$
   after reading the sonar, moving forward, and then backward, the robot knows that it is still close to the wall;
6. $[\mathit{sonar}]\,\boldsymbol{K}([\mathit{forward}]\,\mathit{Close})$
   after reading the sonar, the robot knows that it will remain close after moving forward;
7. $\neg\boldsymbol{K}([\mathit{sonar}]\,\boldsymbol{K}\mathit{Close})$
   the robot does not know initially that it will know that it is close after reading the sonar;
8. $\boldsymbol{K}([\mathit{sonar}]\,(\boldsymbol{K}\mathit{Close} \vee \boldsymbol{K}\neg\mathit{Close}))$
   the robot does know initially that after reading the sonar, it will then know whether or not it is close to the wall;
9. $\boldsymbol{K}(\neg\Box\neg(\boldsymbol{K}\mathit{Close} \vee \boldsymbol{K}\neg\mathit{Close}))$
   the robot does know initially that after some action sequence it will know whether or not it is close to the wall;
10. $\boldsymbol{K}([\mathit{sonar}]\,[\mathit{backward}]\,\neg\boldsymbol{K}\mathit{Close})$
    the robot knows initially that it will not know that it is close after reading the sonar and moving backwards.

Proof: The proofs of these are similar. Here we will only do item 3. Let $z = \langle \mathit{sonar} \cdot \mathit{backward}\rangle$, and suppose that $e, w \models \Sigma \wedge \boldsymbol{O}\Sigma'$; we must show that $e, w, z \models \neg\boldsymbol{K}\mathit{Close}$. Because $e \models \boldsymbol{O}\Sigma'$, the robot has no idea how far from the wall it is; in particular, we have that $e \models \forall x.\neg\boldsymbol{K}(\mathit{distance} \neq x)$, that is, there exists $w' \in e$ such that $w' \simeq_{\langle\rangle} w$ and $w'[\mathit{distance}, \langle\rangle] = 9$. Since $9 < 10$, we also have that $w' \simeq_z w$. However, $w'[\mathit{distance}, z] = 10$. So there exists $w' \in e$ such that $w' \simeq_z w$ and $w', z \models \neg\mathit{Close}$. Therefore, $e, w, z \models \neg\boldsymbol{K}\mathit{Close}$. ∎

14.3 Projection by regression

The examples of the previous section all involve *projection* as a fundamental reasoning task, that is, determining what holds after a number of actions have occurred, as in

$$\Sigma \wedge \boldsymbol{O}\Sigma' \models [\mathit{sonar}]\,[\mathit{backward}]\,\neg\boldsymbol{K}\mathit{Close}.$$

When we are not concerned with knowledge, things are somewhat simpler as we only need

a single basic action theory as in

$\Sigma \models [\mathit{forward}]\,[\mathit{backward}]\,\mathit{Close}$.

We will start by considering this simpler version of the projection problem, before looking at the general case involving knowledge. But first, why is projection problematic? It is because it seems to require non-standard forms of reasoning as both sides of the entailment mention modal operators, which need to be dealt with. In a way, the situation is not unlike the one in Chapter 7, where we addressed the problem of reasoning about knowledge, which at first sight seemed to require heavy doses of modal reasoning involving (nested) beliefs. The solution there was to reduce the problem to classical first-order reasoning with the help of the Representation Theorem. While the technique to solve the projection problem is quite different, the general idea is the same in that we transform the projection problem into one where only classical reasoning is needed. The only restriction is that the query needs to be a bounded objective sentence, that is, no $\Box$ operators are allowed on the query side. The core idea is, roughly, to successively replace all fluents in a query by the right-hand side of their successor state axioms until there are no more actions left. This form of *regression* transforms the query into a fluent formula, which in the end needs to be evaluated against the description of the initial situation (Σ_0), again a purely classical reasoning task.

As we will see, regression can also be extended to deal with queries involving knowledge, resulting in a static formula mentioning $\boldsymbol{K}$-operators. These can then be dealt with as before by applying the Representation Theorem.

14.3.1 Regressing objective formulas

Here we consider regression to determine entailments of the form $\Sigma \models \alpha$, where Σ is a basic action theory and α is any bounded objective sentence. To start with, we assume, from now on, that all basic action theories and queries are *rectified*, that is, that each quantifier has a distinct variable. This is needed for regression to work properly.[7] To simplify the formal details, we will define regression only for formulas in the following normal form *NF*.

Definition 14.3.1: A sentence α is in *NF* if it is rectified and every function symbol f in α occurs only in equality expressions of the form $(f(n_1, \ldots, n_k) = n)$, where the n_i and n here are either variables or standard names.

It is easy to show that every sentence can be transformed into an equivalent one in *NF* and the transformation is linear in the size of the original sentence. For example, the normal

7 See also the proof of Lemma 14.3.8 below, where this is needed to establish the induction for $\forall$.

form of $F(f(b))$ is $\exists x \exists y.(b = x) \wedge (f(x) = y) \wedge F(y)$. Note that, for any formula in *NF*, if a term t appears in $[t]$ or as an argument to a function or predicate, then t is either a variable or a standard name. In the following we will make use of sequences which consist of action variables or action standard names. We will reserve the symbol r to denote such sequences. (We continue to use z to denote the special case where all elements of the sequence are standard names.)

In our account, any bounded, objective sentence α in *NF* is considered regressable. By the transformation above any bounded, objective sentence becomes regressable by first converting it into *NF* and then applying regression to the result.

Definition 14.3.2: Let α be in *NF* and Σ a basic action theory. We define $\mathcal{R}[\alpha]$, *the regression of* α *wrt* Σ, to be $\mathcal{R}[\langle\rangle, \alpha]$, where for any sequence r consisting of action variables or standard names, $\mathcal{R}[r, \alpha]$ is defined inductively on α by:

1. $\mathcal{R}[r, \forall x \alpha] = \forall x \mathcal{R}[r, \alpha]$;
2. $\mathcal{R}[r, (\alpha \wedge \beta)] = (\mathcal{R}[r, \alpha] \wedge \mathcal{R}[r, \beta])$;
3. $\mathcal{R}[r, \neg\alpha] = \neg\mathcal{R}[r, \alpha]$;
4. $\mathcal{R}[r, [t]\alpha] = \mathcal{R}[r \cdot t, \alpha]$;
5. $\mathcal{R}[r, SF(t)] = \mathcal{R}[r, \varphi_t^a]$;
6. $\mathcal{R}[r, F(t_1, \ldots, t_k)]$ for relational fluent F is defined inductively on r by:
   (a) $\mathcal{R}[\langle\rangle, F(t_1, \ldots, t_k)] = F(t_1, \ldots, t_k))$;
   (b) $\mathcal{R}[r \cdot t, F(t_1, \ldots, t_k)] = \mathcal{R}[r, (\gamma_F)_{t\,t_1}^{a\,x_1} \cdots {}_{t_k}^{x_k}]$;
7. $\mathcal{R}[r, (t_1 = t_2)] = (t_1 = t_2)$ if t_1 and t_2 do not mention functional fluents;
8. $\mathcal{R}[r, (f(n_1, \ldots, n_k) = n)]$ for functional fluent f is defined inductively by:
   (a) $\mathcal{R}[\langle\rangle, (f(n_1, \ldots, n_k) = n)] = (f(n_1, \ldots, n_k) = n)$;
   (b) $\mathcal{R}[r \cdot t, (f(n_1, \ldots, n_k) = n)] = \exists y.(\gamma_f)_{t\,n_1}^{a\,x_1} \cdots {}_{n_k}^{x_k} \wedge (y = n)$.

Note that this definition uses the right-hand sides of the successor state, and sense condition axioms from Σ.

It is not hard to show that $\mathcal{R}$ always transforms a bounded objective formula into a fluent formula.

Lemma 14.3.3: *Let α be a bounded objective formula and r a sequence of action variables or standard names. Then there is a unique fluent formula ϕ such that $\mathcal{R}[r, \alpha] = \phi$.*

Proof: The proof is simple but tedious and we will skip the details here. Perhaps the

only interesting aspect is the structure of the proof itself, which is also used in other proofs of properties of regression below. First, the lemma is proved for static formulas only. This is achieved by an induction on the length of r and a sub-induction on the length of α, counting the number of logical operators and where occurrences of $SF(t)$ are counted as the length of $\varphi_t^a + 1$, respectively. Note, in particular, that the induction is well-behaved because the formulas φ, γ_F, and γ_f are themselves fluent formulas, that is, they are static and do not mention SF.

Having proved the lemma for static α, the case of bounded formulas is established by another simple induction on the number of $[t]$-operators in α. ∎

Using the semantics of $\mathcal{ES}$, we will now prove the regression theorem for objective sentences, that is, show that it is possible to reduce reasoning with formulas that contain $[t]$ operators to reasoning with fluent formulas in the initial state.

We begin by defining for any world w and basic action theory Σ another world w_Σ which is like w except that it satisfies the Σ_{sense} and Σ_{post} sentences of Σ.

Definition 14.3.4: Let w be a world, $z \in \mathcal{Z}$, and Σ a basic action theory with fluents $\mathcal{F}$. Then w_Σ is a world satisfying the following conditions:

1. for $h \notin \mathcal{F}$ (predicate or function), $w_\Sigma[h(n_1, \ldots, n_k), z] = w[h(n_1, \ldots, n_k), z]$;
2. for predicate $F \in \mathcal{F}$, $w_\Sigma[F(n_1, \ldots, n_k), z]$ is defined inductively:
   (a) $w_\Sigma[F(n_1, \ldots, n_k), \langle\rangle] = w[F(n_1, \ldots, n_k), \langle\rangle]$;
   (b) $w_\Sigma[F(n_1, \ldots, n_k), z \cdot m] = 1$ *iff* $w_\Sigma, z \models (\gamma_F)^{a\ v_1}_{mn_1} \cdots {}^{v_k}_{n_k}$.
3. for function $f \in \mathcal{F}$, $w_\Sigma[f(n_1, \ldots, n_k), z]$ is defined inductively:
   (a) $w_\Sigma[f(n_1, \ldots, n_k), \langle\rangle] = w[f(n_1, \ldots, n_k), \langle\rangle]$;
   (b) $w_\Sigma[f(n_1, \ldots, n_k), z \cdot m] = n$ *iff* $w_\Sigma, z \models (\gamma_f)^{a\ y\, v_1}_{mnn_1} \cdots {}^{v_k}_{n_k}$.
4. $w_\Sigma[SF(n), z] = 1$ iff $w_\Sigma, z \models \varphi_n^a$.

Note that this again uses the γ, and φ formulas from Σ. Then we get the following simple lemmas:

Lemma 14.3.5: *For any w, w_Σ exists and is uniquely defined.*

Proof: w_Σ clearly exists. The uniqueness follows from the fact that φ is a fluent formula and that for all fluents in $\mathcal{F}$, once their initial values are fixed, then the values after any number of actions are uniquely determined by Σ_{post}. ∎

Lemma 14.3.6: *If* $w \models \Sigma_0$ *then* $w_\Sigma \models \Sigma$.

Proof: The lemma follows directly from the definition of w_Σ, we have that $w_\Sigma \models \forall a \Box SF(a) \equiv \varphi$, $w_\Sigma \models \forall a \forall \vec{x} \Box [a] F(\vec{x}) \equiv \gamma_F$, and $w_\Sigma \models \forall a \forall \vec{x} \forall y \Box [a] f(\vec{x}) = y \equiv \gamma_f$. ∎

Lemma 14.3.7: *If* $w \models \Sigma$ *then* $w = w_\Sigma$.

Proof: If $w \models \Sigma$, that is, $w \models \forall a \Box SF(a) \equiv \varphi$, $w \models \forall a \forall \vec{x} \Box [a] F(\vec{x}) \equiv \gamma_f$, and $w \models \forall a \forall \vec{x} \forall y \Box [a] f(\vec{x}) = y \equiv \gamma_f$, then w satisfies the definition of w_Σ. ∎

The following property of regression is used to prove the main lemma needed for the Regression Theorem. Given a sequence of action variables or standard names r, let r_n^x denote r with all occurrences of variable x replaced by standard name n.

Lemma 14.3.8: *For any bounded objective formula* α *and sequence of action variables or standard names* r, $\mathcal{R}[r, \alpha]_n^x = \mathcal{R}[r_n^x, \alpha_n^x]$.

Proof: The proof is long but simple and follows the structure of the proof of Lemma 14.3.3. Here we only consider static α and three cases: fluent predicates, assuming that the lemma holds for $|r| = k - 1$ and $\forall$, assuming in the sub-induction that the lemma holds for formulas of length $m - 1$.

1. Let $r = r' \cdot t$. Then $\mathcal{R}[r, F(\vec{t})]_n^x = \mathcal{R}[r', \gamma_F {}^{a\,\vec{u}}_{t\,\vec{t}}]_n^x$ (def. of $\mathcal{R}$) $= \mathcal{R}[r_n'^x, (\gamma_F {}^{a\,\vec{u}}_{t\,\vec{t}})_n^x]$ (by induction) $= \mathcal{R}[r_n'^x, (\gamma_F {}^{a\,\vec{u}}_{t_n^x\,\vec{t}_n^x})]$ (since x not in γ_F) $= \mathcal{R}[(r' \cdot t)_n^x, F(\vec{t})_n^x]$.
2. $\mathcal{R}[r, \forall y.\alpha]_n^x = (\forall y.\mathcal{R}[r, \alpha])_n^x = \forall y.\mathcal{R}[r, \alpha]_n^x$ (since $x \neq y$) $= \forall y.\mathcal{R}[r_n^x, \alpha_n^x]$ (by induction on $|\alpha|$) $= \mathcal{R}[r_n^x, (\forall y.\alpha)_n^x]$. ∎

Lemma 14.3.9: *Let* α *be any bounded, objective sentence in NF and* $z \in \mathcal{Z}$. *Then* $w \models \mathcal{R}[z, \alpha]$ *iff* $w_\Sigma, z \models \alpha$.

Proof: As before, the proof is rather straightforward and uses the same induction scheme as Lemma 14.3.3. Assuming the lemma holds for z of length $k - 1$, we only consider two cases, atoms with functional fluents and $\forall$.

1. Note that, by the definition of *NF*, ground atoms mentioning functional fluents have the form $f(n_1, \ldots, n_k) = n$, where n and n_i are standard names. Then: $w_\Sigma, z \cdot m \models f(n_1, \ldots, n_k) = n$ iff (by definition of w_Σ),
   $w_\Sigma, z \models \exists y.(\gamma_f)_{m\,n_1}^{a\;v_1} \ldots {}_{n_k}^{\;v_k} \wedge y = n$ iff (by induction),

$w \models \mathcal{R}[z, \exists y.(\gamma_f)^{a\ v_1\ \ v_k}_{m\ n_1\ \ n_k} \wedge y = n]$ iff (by definition of $\mathcal{R}$),
$w \models \mathcal{R}[z \cdot m, f(n_1, \ldots, n_k) = n]$.

2. $w \models \mathcal{R}[z, \forall x.\alpha]$ iff $w \models \forall x.\mathcal{R}[z, \alpha]$ iff $w \models \mathcal{R}[z, \alpha]^x_n$ for all n of the right sort iff (by Lemma 14.3.8), $w \models \mathcal{R}[z, \alpha^x_n]$ for all n iff (by sub-induction on $|\alpha|$), $w_\Sigma, z \models \alpha^x_n$ for all n iff $w_\Sigma, z \models \forall x.\alpha$. ∎

Theorem 14.3.10: *[Objective Regression] Let* $\Sigma = \Sigma_0 \cup \Sigma_{\text{post}} \cup \Sigma_{\text{sense}}$ *be a basic action theory and let* α *be an objective, bounded sentence. Then* $\mathcal{R}[\alpha]$ *is a fluent sentence and satisfies*

$\Sigma \models \alpha$ *iff* $\Sigma_0 \models \mathcal{R}[\alpha]$.

Proof: Suppose $\Sigma_0 \models \mathcal{R}[\alpha]$. We prove that $\Sigma \models \alpha$. Let w be any world such that $w \models \Sigma$. Then, $w \models \Sigma_0$, and so $w \models \mathcal{R}[\alpha]$. By Lemma 14.3.9, $w_\Sigma \models \alpha$. By Lemma 14.3.7, $w_\Sigma = w$, and so $w \models \alpha$.

Conversely, suppose $\Sigma \models \alpha$. We need to prove that $\Sigma_0 \models \mathcal{R}[\alpha]$. Let w be any world such that $w \models \Sigma_0$. From Lemma 14.3.6, $w_\Sigma \models \Sigma$, and so $w_\Sigma \models \alpha$. By Lemma 14.3.9, $w \models \mathcal{R}[\alpha]$. ∎

14.3.2 Regressing knowledge

Let us now turn to the more general case of regression for bounded sentences which may refer to the agent's knowledge. As we discussed in Section 14.2, this means that we need to consider two basic action theories Σ and Σ' for what is true in the world and for what the agent believes, respectively.

The following theorem can be thought of as a successor-state axiom for knowledge, which will allow us to extend regression to formulas containing $\boldsymbol{K}$. Note that, in contrast to the successor state axioms for fluents, this is a *theorem* of the logic not a stipulation as part of a basic action theory:

Theorem 14.3.11: $\models \forall a.\Box[a]\boldsymbol{K}\alpha \equiv SF(a) \wedge \boldsymbol{K}(SF(a) \supset [a]\alpha) \vee \neg SF(a) \wedge \boldsymbol{K}(\neg SF(a) \supset [a]\alpha)$.

Proof: For both directions of the equivalence we will only consider the case where $\neg SF(n)$ holds for an arbitrary action name n. The other case is completely analogous.

To prove the only-if direction, let $e, w, z \models [n]\boldsymbol{K}\alpha^a_n$ for action name n. We write α' for α^a_n. Suppose $e, w, z \models \neg SF(n)$. It suffices to show that $e, w, z \models \boldsymbol{K}(\neg SF(n) \supset [n]\alpha')$. So suppose $w' \simeq_z w$, $w' \in e$, and $w'[SF(n), z] = 0$. Thus $w'[SF(n), z] = w[SF(n), z]$ and,

hence, $w' \simeq_{z \cdot n} w$. Since $e, w, z \models [n]\boldsymbol{K}\alpha'$ by assumption, $e, w', z \cdot n \models \alpha'$, from which $e, w', z \models [n]\alpha'$ follows.

Conversely, let $e, w, z \models \neg SF(n) \wedge \boldsymbol{K}(\neg SF(n) \supset [n]\alpha')$. We need to show that $e, w, z \models [n]\boldsymbol{K}\alpha'$, that is, $e, w, z \cdot n \models \boldsymbol{K}\alpha'$. Let $w' \simeq_{z \cdot n} w$ and $w' \in e$. Then $w'[SF(n), z] = w[SF(n), z] = 0$ by assumption. Hence $e, w', z \models \neg SF(n)$. Therefore, by assumption, $e, w', z \cdot n \models \alpha'$, from which $e, w, z \models [n]\boldsymbol{K}\alpha'$ follows. ∎

We consider this a successor state axiom for knowledge in the sense that it tells us for any action a what will be known after doing a in terms of what was true before. In this case, knowledge after a depends on what was known before doing a about what the future would be like after doing a, contingent on the sensing information provided by a. For example, if after doing *sonar* the robot knows it is close to the wall, then before doing *sonar*, the robot already knew a conditional: if the *sonar* returns a 1 on completion, then this indicates that the robot will be close to the wall.

We are now ready to extend regression to deal with knowledge. More precisely, we are interested in regressing bounded basic formulas in *NF*. Instead of being defined relative to a basic action theory Σ, the regression operator $\mathcal{R}$ will be defined relative to a *pair* of basic action theories $\langle \Sigma', \Sigma \rangle$ where, as above, Σ' represents the beliefs of the agent. We allow Σ and Σ' to differ arbitrarily and indeed to contradict each other, so that agents may have false beliefs about what the world is like, including its dynamics. The idea is to regress *wrt* Σ outside of $\boldsymbol{K}$ operators and *wrt* Σ' inside. To be able to distinguish between these cases, $\mathcal{R}$ now carries the two basic action theories with it as extra arguments.

Rule 1–10 of the new regression operator $\mathcal{R}$ are exactly as before (Definition 14.3.2) except for the extra arguments Σ' and Σ. Then we add the following:

11. $\mathcal{R}[\Sigma', \Sigma, r, \boldsymbol{K}\alpha]$ *is defined inductively on* r *by:*

- *(a)* $\mathcal{R}[\Sigma', \Sigma, \langle\,\rangle, \boldsymbol{K}\alpha] = \boldsymbol{K}(\mathcal{R}[\Sigma', \Sigma', \langle\,\rangle, \alpha])$*;*
- *(b)* $\mathcal{R}[\Sigma', \Sigma, r \cdot t, \boldsymbol{K}\alpha] = \mathcal{R}[\Sigma', \Sigma, r, \beta_t^a]$*, where* β *is the right-hand side of the equivalence in Theorem 14.3.11.*

For simplicity, we write $\mathcal{R}[\alpha]$ instead of $\mathcal{R}[\Sigma', \Sigma, \langle\,\rangle, \alpha]$. Next we extend Lemma 14.3.6 to knowledge, where $e_\Sigma = \{w_\Sigma \mid w \in e\}$ for a given epistemic state e and basic action theory Σ:

Lemma 14.3.12: *If* $e \models \boldsymbol{O}\Sigma_0$ *then* $e_\Sigma \models \boldsymbol{O}\Sigma$.

Proof: Let $e \models \boldsymbol{O}\Sigma_0$, that is, for all w, $w \in e$ iff $w \models \Sigma_0$. We need to show that for all w, $w \in e_\Sigma$ iff $w \models \Sigma$.

Let $w \in e_\Sigma$. By definition, there is a $w' \in e$ such that $w = w'_\Sigma$. Since $w' \models \Sigma_0$, by Lemma 14.3.6, $w'_\Sigma \models \Sigma$, that is, $w \models \Sigma$.

Conversely, let $w \models \Sigma$. Then $w \models \Sigma_0$ and hence, by assumption, $w \in e$. By Lemma 14.3.7, $w = w_\Sigma$ and thus $w \in e_\Sigma$. ∎
We now turn to the generalization of Lemma 14.3.9 for knowledge.

Lemma 14.3.13: *Let α be any bounded basic sentence in NF. Then $e, w \models \mathcal{R}[\Sigma', \Sigma, z, \alpha]$ iff $e_{\Sigma'}, w_\Sigma, z \models \alpha$.*

Proof: The proof is by induction on z with a sub-induction on α.

Let $z = \langle\rangle$. The proof for *SF*, atoms, and the connectives $\neg$, $\wedge$, and $\forall$ is exactly analogous to Lemma 14.3.9.
For formulas $\boldsymbol{K}\alpha$ we have: $e_{\Sigma'} \models \boldsymbol{K}\alpha$ iff
for all $w \in e_{\Sigma'}$, $e_{\Sigma'}, w \models \alpha$ iff (by definition of $e_{\Sigma'}$),
for all $w \in e$, $e_{\Sigma'}, w_{\Sigma'} \models \alpha$ iff (by induction),
for all $w \in e$, $e, w, \models \mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha]$ iff
$e \models \boldsymbol{K}(\mathcal{R}[\Sigma', \Sigma', \langle\rangle, \alpha])$ iff (by definition of $\mathcal{R}$),
$e \models \mathcal{R}[\Sigma', \Sigma, \langle\rangle, \boldsymbol{K}\alpha]$.
This concludes the base case $z = \langle\rangle$.

Now consider the case of $z \cdot n$, which again is proved by a sub-induction on α. The proof is exactly like the sub-induction for the base case except for $\boldsymbol{K}$, for which we have the following: $e_{\Sigma'}, w_\Sigma, z \cdot n \models \boldsymbol{K}\alpha$ iff (by Theorem 14.3.11),
$e_{\Sigma'}, w_\Sigma, z \models \beta_n^a$ (where the β is from Theorem 14.3.11)
iff (by the main induction),
$e, w \models \mathcal{R}[\Sigma', \Sigma, z, \beta_n^a]$ iff (by definition of $\mathcal{R}$),
$e, w \models \mathcal{R}[\Sigma', \Sigma, z \cdot n, \boldsymbol{K}\alpha]$, which completes the proof. ∎

Finally, here is the general regression theorem:

Theorem 14.3.14: *[Generalized Regression] Let Σ and Σ' be basic action theories, and α be a bounded basic sentence in NF. Then $\mathcal{R}[\alpha]$ is a static sentence and satisfies*
$\Sigma \wedge \boldsymbol{O}\Sigma' \models \alpha$ *iff* $\Sigma_0 \wedge \boldsymbol{O}\Sigma_0' \models \mathcal{R}[\alpha]$.

Proof: To prove the only-if direction, let us suppose that $\Sigma \wedge \boldsymbol{O}\Sigma' \models \alpha$ and that $e, w \models \Sigma_0 \wedge \boldsymbol{O}\Sigma_0'$. Thus $w \models \Sigma_0$ and, by Lemma 14.3.6, $w_\Sigma \models \Sigma$. Also, $e \models \boldsymbol{O}\Sigma_0'$ and thus, by Lemma 14.3.12, $e_{\Sigma'} \models \boldsymbol{O}\Sigma'$. Therefore, $e_{\Sigma'}, w_\Sigma \models \Sigma \wedge \boldsymbol{O}\Sigma'$. By assumption, $e_{\Sigma'}, w_\Sigma \models \alpha$ and, by Lemma 14.3.13, $e, w \models \mathcal{R}[\alpha]$.

Conversely, suppose $\Sigma_0 \wedge \boldsymbol{O}\Sigma_0' \models \mathcal{R}[\alpha]$ and let $e, w \models \Sigma \wedge \boldsymbol{O}\Sigma'$. Then $w \models \Sigma_0$ and $e \models \boldsymbol{O}\Sigma_0'$. Then, by assumption, $e, w \models \mathcal{R}[\alpha]$. Then $e_{\Sigma'}, w_\Sigma \models \alpha$ by Lemma 14.3.13. By Lemma 14.3.7, $w_\Sigma = w$ and, since $e \models \boldsymbol{O}\Sigma'$, $e_{\Sigma'} = e$. Therefore, $e, w \models \alpha$. ∎

This theorem shows that determining what is true and what is known after any (bounded) number of actions have occurred can always be reduced to reasoning about what is true and known in the initial state.

Note that if action terms t within α only occur inside action operators, then $\mathcal{R}[\alpha]$ does not mention any action terms, that is, $\mathcal{R}[\alpha]$ is in the language of $\mathcal{OL}$ because the regression operator removes action terms from the formula (see Rule 4 of Definition 14.3.2). In such cases, regression-based reasoning can then be reduced to reasoning in $\mathcal{OL}$ using Theorem 14.1.1:

Corollary 14.3.15: *Let Σ_0 and Σ_0' be defined as above. Let α be a regressable sentence where action terms only occur inside action operators. Then*

$$\Sigma \wedge \boldsymbol{O}\Sigma' \models \alpha \quad \textit{iff} \quad \Sigma_0 \wedge \boldsymbol{O}\Sigma_0' \quad \textit{logically implies} \quad \mathcal{R}[\alpha] \textit{ in } \mathcal{OL}.$$

With that we can now go a step further and leverage the Representation Theorem from Chapter 7 to reduce the problem to one about reasoning in $\mathcal{L}$ alone:

Corollary 14.3.16: $\Sigma \wedge \boldsymbol{O}\Sigma' \models \alpha$ *iff* Σ_0 *logically implies* $\|\mathcal{R}[\alpha]\|_{\Sigma_0'}$ *in* $\mathcal{L}$.

14.4 Bibliographic notes

This chapter is based on [100], where $\mathcal{ES}$ was introduced as a fragment of the situation calculus, which itself dates back to John McCarthy [139] and was later re-formulated by Ray Reiter [162]. Perhaps the main difference between our treatment and the original situation calculus is that in $\mathcal{ES}$ situations are only part of the semantics, whereas they are explicitly referred to in the situation calculus. For example, $[drop(cup5)]Broken(cup5)$ could be expressed as $Broken(cup5, do(drop(cup), s))$, where s is the current situation and $do(drop(cup5), s)$ refers to the situation that results from dropping $cup5$. Having access to situations within the language makes the situation calculus more expressive than $\mathcal{ES}$. However, as we saw, $\mathcal{ES}$ is still strong enough to address the projection problem, which is one of the main uses of action formalisms. As mentioned earlier, it is easy to incorporate explicit action preconditions into $\mathcal{ES}$. In [100] this was done by adding a special fluent predicate *Poss* to the language with a single argument of type action. Basic action theories are then extended by adding a sentence of the form $\Box Poss(a) \equiv \pi$, where π is a fluent formula expressing the preconditions of the various actions considered in the BAT. Regression can then be adapted by replacing any occurrence of $Poss(t)$ in a regressable sentence by π_t^a. The situation calculus is also the basis for the action programming language Golog [121, 98], which has been employed for the control of robots [44]. In [97] we

showed how to define Golog using $\mathcal{ES}$ as the base logic.

While the situation calculus has received a lot of attention in the reasoning about action community over the years, there are, of course, a number of alternative formalisms, including close relatives like the fluent calculus [68, 182] and more distant cousins like [80, 52, 169].

$\mathcal{ES}$ is also closely related to dynamic logic [63]. For example, De Giacomo and Lenzerini [28] and later Demolombe et al. [32] show how to express successor state axioms in an extension of dynamic logic. There are also epistemic extensions of dynamic logic such as [64] and [31]. In the language of [64], it is possible to express things like [*forward*][*sonar*]$\boldsymbol{K}$*Close* using an almost identical syntax and where $\boldsymbol{K}$ also has a possible-world semantics. While most approaches remain propositional, there are some first-order treatments such as [31, 32], which, like $\mathcal{ES}$, are inspired by the desire to capture fragments of the situation calculus in modal logic.

$\mathcal{ES}$ itself has also been used and extended in various ways. In [103], we considered a form of limited belief with actions, where reasoning can be shown to be decidable for a variant of proper$^+$ knowledge bases discussed in the bibliographic notes of Chapter 13. In [99], we explored only-knowing after actions with forgetting. The same paper also included Moore-style default reasoning in the presence of actions. Finally, $\mathcal{ES}$ was extended in order to express temporal properties over possibly infinite action sequences, which then served as the basis for the verification of Golog programs [21, 22].

14.5 Where do we go from here?

The reader may have noticed that in our examples and the discussion about regression we only considered only-knowing for initial situations, that is, before any actions have occurred. While the semantics of $\boldsymbol{O}$ is well defined for arbitrary situations, it may not be all that useful for non-initial situations. For example, suppose an agent only-knows a basic action theory and then performs action A. A reasonable question to ask is what the agent should only-know after the action occurred. Clearly, we would expect the agent to still know the successor state axioms and the sensed-fluent axiom as they are situation-independent. The description of the initial situation (Σ_0) would likely have to change as A may have affected fluents mentioned in Σ_0. And what about knowledge of the past? After all, the agent just performed A and it seems reasonable to expect the agent to know that and perhaps what was true in the past. The trouble is that we could not even express such things as we have no way to refer to the past. When we assume that an agent forgets everything about the past, things become a little easier and we explored a suitable modification of only-knowing for that case elsewhere. The story without forgetting still needs to be told.

Having laid out a framework for handling knowledge and action opens the door to try and generalize other topics from the book as well. In the case of defaults, a natural question to ask is how they should behave when actions are performed? Or what does it mean for actions themselves to have default effects? Tractable reasoning also offers new challenges when actions are involved. One approach would be to first use regression to reduce a query about the future to a query about the initial situation, and then simply use the ideas we presented for the static case. A downside of this view is that regression is an inherently syntactic operation. The hard question is what a semantic account, perhaps along the lines of some form of extended worlds, would look like.

14.6 Exercises

1. Prove Theorem 14.1.1.
2. Prove that the induction axiom is valid (Theorem 14.1.3, item 8).
3. Prove Lemma 14.1.5.
4. Prove Theorem 14.1.1
5. Modify the definition of *SF* to allow for sensing arbitrary values such as the actual distance to a wall.
6. Prove statements 2 and 8 of Example 14.2.2 semantically.
7. Prove statement 3 of Example 14.2.2 using regression.

References

[1] A. R. Anderson and N. D. Belnap, Jr., First Degree Entailments. *Math. Annalen*, **149**, 302–319, 1963.

[2] A. R. Anderson and N. D. Belnap, Jr., *Entailment, The Logic of Relevance and Necessity*. Princeton University Press, 1975.

[3] P. Atzeni, S. Ceri, S. Paraboschi, and R. Torlone,MI *Database Systems: Concepts, Languages, and Architectures*. McGraw-Hill Publishing Co., Berkshire, 2000.

[4] F. Baader, I. Horrocks, and U. Sattler, Description Logics. In: F. van Harmelen, V. Lifschitz, and B. W. Porter (editors), *Handbook of Knowledge Representation*, pages 135–179, Elsevier, 2008.

[5] J. Barwise and J. Perry, *Situations and Attitudes*. MIT Press, Cambridge, 1983.

[6] S. Ben-David and Y. Gafni, All we believe fails in impossible worlds. Manuscript, Department of Computer Science, Technion, 1989.

[7] E. Börger and E. Grädel and Y. Gurevich, *The Classical Decision Problem*. Springer-Verlag, 1997.

[8] R. Brachman, R. Fikes and H. J. Levesque, KRYPTON: A functional approach to knowledge representation. *IEEE Computer*, **16**(10), 67–73, 1983. Reprinted in [10].

[9] R. Brachman and H. J. Levesque, Competence in knowledge representation. *Proceedings of the National Conference of the American Association for Artificial Intelligence (AAAI-82)*, AAAI Press/MIT Press, Cambridge, 189–192, 1982.

[10] R. Brachman and H. J. Levesque (eds.), *Readings in Knowledge Representation*. Morgan Kaufmann Publishers, Inc., San Francisco, 1985.

[11] R. Brachman and H. J. Levesque, *Knowledge Representation and Reasoning*. Elsevier, 2004.

[12] R. Brachman and J. Schmolze, An overview of the KL-ONE knowledge representation system. *Cognitive Science*, **9**(2), 171–216, 1985.

[13] M. Brand and R. Harnish (eds.), *The Representation of Knowledge and Belief*. The University of Arizona Press, Tucson, 1986.

[14] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, The interactive museum tour-guide robot. *Proceedings of the National Conference on Artificial Intelligence (AAAI-98)*, AAAI Press/MIT Press, Cambridge, 11–18, 1998.

[15] M. Cadoli and M. Schaerf. On the complexity of entailment in propositional multivalued logics. *Annals of Mathematics and Artificial Intelligence*, 18(1):29–50, 1996.

[16] B. Chellas, *Modal Logic: An Introduction*. Cambridge University Press, Cambridge, 1980.

[17] J. Chen, The logic of only knowing as a unified framework for non-monotonic reasoning. *Fundamenta Informaticae*, **21**, 205–220, 1994.

[18] J. Chen, The generalized logic of only knowing that covers the notion of epistemic specifications. *Journal of Logic and Computation*, **7**(2), 159–174, 1997.

[19] C. Cherniak, *Minimal Rationality*. MIT Press, Cambridge, 1986.

[20] J. Claßen and G. Lakemeyer. Foundations for Knowledge-Based Programs using ES. In *Proc. KR 2006*, 318–328, 2006.

[21] J. Claßen and G. Lakemeyer. A Logic for Non-Terminating Golog Programs. In *Proc. KR 2008*, 589–599, 2008.

[22] J. Claßen, B. Zarrieß, and G. Lakemeyer Situation Calculus Meets Description Logics. In C. Lutz, U. Sattler, C. Tinelli, A.-Y. Turhan, and F. Wolter (Eds.) *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday.* LNCS 11560, Springer 2019.

[23] B. Cohen, and G. Murphy, Models of concepts. *Cognitive Science*, **8**(1), 27–59, 1984.

[24] J.M. Crawford and D. Etherington. A non-deterministic semantics for tractable inference. In *Proc. of AAAI-98*, 286–291, 1998.

[25] M. D'Agostino. An informational view of classical logic. Theor. Comput. Sci. 606, 79–97, 2015.

[26] M. Dalal. Semantics of an anytime family of reasoners. In *Proc. of the 12th European Conference on Artificial Intelligence (ECAI-96)*, pages 360–364, 1996.

[27] A. Darwiche and J. Pearl, Symbolic causal networks. *Proceedings of the 12th National Conference on*

Artificial Intelligence (AAAI-94), AAAI Press/MIT Press, Cambridge, 238–244, 1994.

[28] G. De Giacomo and M. Lenzerini. PDL-based framework for reasoning about actions. In *Proc. of AI*IA*, Springer LNAI 992, 103–114, 1995.

[29] G. De Giacomo, Y. Lespérance, and H. J. Levesque, Efficient Reasoning in Proper Knowledge Bases with Unknown Individuals. *Proc. of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, 827–832, 2011.

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

[31] R. Demolombe. Belief change: from Situation Calculus to Modal Logic. *IJCAI Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC'03),* Acapulco, Mexico, 2003.

[32] R. Demolombe, A. Herzig, and I.J. Varzinczak. Regression in modal logic. *J. of Applied Non-Classical Logics*, 13(2):165-185, 2003.

[33] D. Dennett, *The Intentional Stance*. MIT Press, Cambridge, 1987.

[34] D. Dennett, Precis of *The Intentional Stance*. *Brain and Behavioral Sciences*, **16**(2), 289–391, 1993.

[35] F. Donini, D. Nardi, and R. Rosati, Ground nonmonotonic modal logics. *J. of Logic and Computation* **7(4)**, 523–548, 1997.

[36] H. Dreyfus, *What Computers Still Can't Do: A Critique of Artificial Reason*. MIT Press, Cambridge, 1992.

[37] J. M. Dunn, Intuitive semantics for first-degree entailments and coupled trees. *Philosophical Studies* **29**, 149–168, 1976.

[38] P. Edwards (ed.), *The Encyclopedia of Philosophy*. Macmillan Publishing Co., New York, 1967.

[39] J. J. Elgot-Drapkin, Step-logic and the three-wise-men problem. *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI-91)*, AAAI-Press/MIT Press, Cambridge, 412–417, 1991.

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

[41] H. Enderton, *A Mathematical Introduction to Logic*. Academic Press, New York, 1972.

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

[43] R. Fagin, J. Y. Halpern, and M. Y. Vardi, A nonstandard approach to the logical omniscience problem. *Artificial Intelligence* **79**(2), 203-240, 1996.

[44] A. Ferrein and G. Lakemeyer. Logic-based robot control in highly dynamic domains. Robotics Auton. Syst. 56(11): 980-991, 2008.

[45] N. Findler (ed.), *Associative Networks: Representation and Use of Knowledge by Computers*. Academic Press, New York, 1979.

[46] A. Frisch, *Knowledge Retrieval as Specialized Inference*. Ph. D. Thesis, Department of Computer Science, University of Rochester.

[47] A. Frisch, Inference without chaining. *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI-87)*, Morgan Kaufmann, San Francisco, 515–519, 1987.

[48] A. Frisch, The substitutional framework for sorted deduction: fundamental results on hybrid reasoning. *Artificial Intelligence* **49**, 161–198, 1991.

[49] P. Gärdenfors, *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, 1988.

[50] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.

[51] M. Gelfond, Strong introspection. *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, AAAI Press/MIT Press, Cambridge, 386–391, 1991.

[52] Michael Gelfond and Vladimir Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17:301–321, 1993.

[53] E. Gettier, Is justified true belief knowledge. *Analysis*, **23**, 121–123, 1963. Reprinted in [57].

[54] G. Gottlob, Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, **2**, 397–425, 1992.

[55] G. Gottlob, Translating default logic into standard autoepistemic logic. *Journal of the ACM*, **42**(4), 711–740, 1995.

[56] C. Green, *The Application of Theorem-Proving to Question Answering Systems*. Ph. D. thesis, Department of Electrical Engineering, Stanford University, Stanford, 1969.

[57] A. Griffiths (ed.), *Knowledge and Belief*. Oxford University Press, London, 1967.

[58] J. Y. Halpern, Reasoning about only knowing with many agents. *Proceedings of the National Conference on Artificial Intelligence (AAAI'93)*, AAAI-Press/MIT Press, Cambridge, 655–661, 1993.

[59] J. Y. Halpern and G. Lakemeyer, Levesque's axiomatization of only knowing is incomplete. *Artificial Intelligence* **74**(2), 381–387, 1995.

[60] J. Y. Halpern and G. Lakemeyer, Multi-agent only knowing. *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-VI)*, Morgan Kaufmann, 251–265, 1996.

[61] J. Y. Halpern and Y. Moses, Towards a theory of knowledge and ignorance. *Proceedings of the AAAI Workshop on Non-monotonic Logic*, 125–143, 1984. Reprinted in K. R. Apt (ed.), *Logics and Models of Concurrent Systems*, Springer-Verlag, Berlin/New York, 459–476, 1985.

[62] J.Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992.

[63] D. Harel. Dynamic Logic. In D. Gabbay and F. Guenther (Eds.), *Handbook of Philosophical Logic*, Vol. 2, D. Reidel Publishing Company, 497–604, 1984.

[64] A. Herzig, J. Lang, D. Longin, and T. Polacsek, A logic for planning under partial observability. In *Proc. AAAI-2000*, AAAI Press, 2002.

[65] J. Hintikka,, *Knowledge and Belief*. Cornell University Press, Ithaca, 1962.

[66] J. Hintikka, Impossible worlds vindicated. *Journal of Philosophical Logic* **4**, 475–484, 1975.

[67] G. Hirst, Existence assumptions in knowledge representation. *Artificial Intelligence*, **49**, 199–242, 1991.

[68] S. Hölldobler and J. Schneeberger. A new deductive approach to planning. *New Generation Computing*, 8:225–244, 1990.

[69] G. Hughes, and M. Cresswell, *An Introduction to Modal Logic*. Methuen and Co., London, 1968.

[70] I. L. Humberstone, I. L., A more discriminating approach to modal logic. *Journal of Symbolic Logic* **51**(2), 503–504, 1986. (Abstract only.) There is also an expanded, but unpublished, manuscript.

[71] D. Kaplan, Quantifying-in. [127], 112–144, 1971.

[72] K. Katsuno and A. Mendelzon, Propositional knowledge base revision and minimal change. *Artificial Intelligence*, **52**, 263–294, 1991.

[73] T.Q. Klassen, S.A. McIlraith, and H.J. Levesque, Towards Tractable Inference for Resource-Bounded Agents. AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning, 89–95, AAAI Press, 2015.

[74] S. Kleene, On a notation for ordinal numbers Journal of Symbolic Logic, 3, 150–155, 1938.

[75] K. Konolige, A computational theory of belief introspection. *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*, Morgan Kaufmann, San Francisco, 502–508, 1985.

[76] K. Konolige, *A Deduction Model of Belief*. Research Notes in Artificial Intelligence, Pitman, London, 1986.

[77] K. Konolige, On the relation between default logic and autoepistemic theories. *Artificial Intelligence* **35**(3), 343–382, 1988. (Errata: *Artificial Intelligence* **41**, 115, 1989.)

[78] K. Konolige, On the relation between autoepistemic logic and circumscription (preliminary report). *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1213–1218, 1989.

[79] K. Konolige, Quantification in autoepistemic logic. *Fundamenta Informaticae* **15**(3-4), 1991.

[80] R. Kowalski and M. Sergot. A logic based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[81] S. Kripke, A completeness theorem in modal logic. *Journal of Symbolic Logic*, **24**, 1–14, 1959.

[82] S. Kripke, Is there a problem with substitutional quantification? G. Evans and J. McDowell (eds.), *Truth and Meaning*, Clarendon Press, Oxford, 325–419, 1976.

[83] S. Kripke, *Naming and Necessity*. Harvard University Press, Cambridge, 1980.

[84] G. Lakemeyer, *Models of Belief for Decidable Reasoning in Incomplete Knowledge Bases*. Ph. D. thesis, Dept. of Computer Science, University of Toronto, 1990. (A revised version appeared as: Technical Report KRR-TR-92-5, University of Toronto, 1992.)

[85] G. Lakemeyer, All you ever wanted to know about Tweety. *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, Morgan Kaufmann, San Mateo, CA, 639–648, 1992.

[86] G. Lakemeyer, On perfect introspection with quantifying-in. *Fundamenta Informaticae*, **17**(1,2), 75–98, 1992.

[87] G. Lakemeyer, All they know: a study in multi-agent autoepistemic reasoning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI '93)*, 376–381, 1993.

[88] G. Lakemeyer, All they know about. *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93)*, AAAI Press/MIT Press, Cambridge, 662–667, 1993.

[89] G. Lakemeyer, Limited reasoning in first-order knowledge bases. *Artificial Intelligence 71*, 1–42, 1994.

[90] G. Lakemeyer, A logical account of relevance. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Morgan Kaufmann, 853–859, 1995.

[91] G. Lakemeyer, Limited reasoning in first-order knowledge bases with full introspection. *Artificial Intelligence* **84**, 209–255, 1996.

[92] G. Lakemeyer, Relevance from an epistemic perspective. *Artificial Intelligence* **97**(1–2), 137–167, 1997.

[93] G. Lakemeyer and H. J. Levesque, A tractable knowledge representation service with full introspection. *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge*, Morgan Kaufmann, San Francisco, 145–159, 1988.

[94] G. Lakemeyer and H. J. Levesque, $\mathcal{AOL}$: a logic of acting, sensing, knowing, and only knowing. *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, Morgan Kaufmann, San Francisco, 316–327, 1998.

[95] G. Lakemeyer and H. J. Levesque, Query evaluation and progression in $\mathcal{AOL}$ knowledge bases. *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, Morgan Kaufmann, San Francisco, 124–131, 1999.

[96] G. Lakemeyer and H.J. Levesque, Evaluation-based reasoning with disjunctive information in first-order knowledge bases. In *Proc. of KR-02*, pages 73–81, 2002.

[97] G. Lakemeyer and H. J. Levesque, *A useful fragment of the situation calculus*, *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, 490–496, 2005.

[98] G. Lakemeyer and H.J. Levesque, Cognitive Robotics. In: F. van Harmelen, V. Lifschitz, B.W. Porter (Eds.): Handbook of Knowledge Representation. Foundations of Artificial Intelligence 3, Elsevier, 869–886, 2008.

[99] G. Lakemeyer and H.J. Levesque, A Semantical Account of Progression in the Presence of Defaults. In *Proc. of IJCAI 2009*, 842–847, 2009.

[100] G. Lakemeyer and H.J. Levesque, A semantic characterization of a useful fragment of the situation calculus with knowledge. Artif. Intell. 175(1): 142–164, 2011.

[101] G. Lakemeyer and H.J. Levesque, Only-Knowing meets Nonmonotonic Modal Logic.. *Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-12),* Rome, 2012.

[102] G. Lakemeyer and H.J. Levesque, Decidable Reasoning in a Logic of Limited Belief with Introspection and Unknown Individuals. *23rd International Joint Conference on Artificial Intelligence (IJCAI),* 2013.

[103] G. Lakemeyer and H.J. Levesque, Decidable Reasoning in a Fragment of the Epistemic Situation

Calculus. *Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-14),* Vienna, 2014.

[104] G. Lakemeyer and H. J. Levesque, Only Knowing. In H. van Ditmarsch, J. Y. Halpern, W. van der Hoek, and B. Kooi (Eds.) Handbook of Epistemic Logic, College Publications, 2015.

[105] G. Lakemeyer and H.J. Levesque, Decidable Reasoning in a Logic of Limited Belief with Function Symbols. In *Proc of KR-16*, pages 288-297, 2016.

[106] G. Lakemeyer and H.J. Levesque. A Tractable, Expressive, and Eventually Complete First-Order Logic of Limited Belief. In *Proc. IJCAI-19*, pages 1764-1771, 2019.

[107] G. Lakemeyer and H.J. Levesque, A First-Order Logic of Limited Belief Based on Possible Worlds. *Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR)*, pages 624–635, 2020.

[108] G. Lakemeyer and S. Meyer, Enhancing the power of a decidable first-order reasoner. *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, Morgan Kaufmann, San Francisco, 403–414, 1994.

[109] H. Leblanc, On dispensing with things and worlds. M. Munitz (ed.), *Logic and Ontology*, New York University Press, New York, 241–259, 1973.

[110] H. Leblanc, Alternatives to standard first-order semantics. D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, Volume 1, Kluwer Academic Press, 189–274, 1983.

[111] H. J. Levesque, *A Formal Treatment of Incomplete Knowledge Bases*. Ph. D. thesis, Dept. of Computer Science, University of Toronto, 1981.

[112] H. J. Levesque, A logic of implicit and explicit belief. *Proceedings of the 4th National Conference on Artificial Intelligence (AAAI-84)*, AAAI Press/MIT Press, Cambridge, 198–202, 1984.

[113] H. J. Levesque, Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, **23**, 155–212, 1984.

[114] H. J. Levesque, Knowledge representation and reasoning. *Annual Review of Computer Science 1986*, Annual Reviews Inc., Palo Alto, 255–287, 1986.

[115] H. J. Levesque, Logic and the complexity of reasoning. *The Journal of Philosophical Logic*, **17**, 355–389, 1988.

[116] H. J. Levesque, A knowledge-level account of abduction. *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, Morgan Kaufmann, San Francisco, 1061–1067, 1989.

[117] H. J. Levesque, All I know: a study in autoepistemic logic. *Artificial Intelligence* **42**, 263–309, 1990.

[118] H. J. Levesque, A Completeness Result for Reasoning with Incomplete First-Order Knowledge Bases. *Proc. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, 14–23, 1998.

[119] H. J. Levesque and G. Lakemeyer. The Logic of Knowledge Bases. First Edition, MIT Press, 2001.

[120] H. J. Levesque, F. Pirri, and R. Reiter, Foundations for the situation calculus. Linköping Electronic Articles in Computer and Information Science, **3**, 1998, available at `http://www.ep.liu.se/ea/cis/1998/018/`.

[121] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl., Golog: A logic programming language for dynamic domains. *Journal of Logic Programming,* **31**, 59–84, 1997.

[122] D. Lewis, *Counterfactuals*. Blackwell, Oxford, 1973.

[123] V. Lifschitz, Minimal belief and negation as failure. *Artificial Intelligence* **70**, 53–72, 1994.

[124] F. Lin and R. Reiter, Forget it!, *Proceedings of the AAAI Fall Symposium on Relevance*, AAAI Press, 154–159, 1994.

[125] F. Lin and R. Reiter, How to progress a database. *Artificial Intelligence*, **92**, 131-167, 1997.

[126] F. Lin and Y. Shoham, Epistemic semantics for fixed-point non-monotonic logics. *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK-III)*, Morgan Kaufmann, San Francisco, 111–120, 1990.

[127] L. Linsky (ed.), *Reference and Modality*. Oxford University Press, Oxford, 1971.

[128] L. Linsky, *Names and Descriptions*. University of Chicago Press, Chicago, 1977.

[129] B. Liskov and S. Zilles, Programming with abstract data types. *SIGPLAN Notices*, **9**(4), 1974.

[130] Y. Liu, G. Lakemeyer, and H.J. Levesque. A Logic of Limited Belief for Reasoning with Disjunctive Information. *Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR),* 587–597, 2004.

[131] Y. Liu and H.J. Levesque. A tractability result for reasoning with incomplete first-order knowledge bases. In *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 83–88, 2003.

[132] Y. Liu and H.J. Levesque. Tractable reasoning in first-order knowledge bases with disjunctive information. *In Proc. of the Twentieth National Conference on Artificial Intelligence (AAAI),* 2005.

[133] J. Lloyd, *Foundations of Logic Programming*, Second Edition. Springer Verlag, New York, 1987.

[134] W. Marek and M. Truszczynski, Relating autoepistemic and default logics. *Proc. of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, Morgan Kaufmann, San Francisco, 276–288, 1989.

[135] W. Marek and M. Truszczynski, Modal logic for default reasoning. *Annals of Mathematics and Artificial Intelligence* **1**, 275–302, 1990.

[136] W. Marek and M. Truszczynski, Autoepistemic logic. *Journal of the ACM*, **38**(3), 588–619, 1991.

[137] V. Marek, G. Schwarz, and M. Truszczynski, Modal Nonmonotonic Logics: Ranges, Characterization, Computation. *Journal of the ACM*, 40(4): 963-990, 1993.

[138] J. McCarthy, Programs with common sense. M. Minsky (ed.), *Semantic Information Processing*, MIT Press, Cambridge, 403–418, 1963. Reprinted in [10].

[139] J. McCarthy, *Situations, Actions and Causal Laws.* Technical Report, Stanford University, 1963. Also in M. Minsky (ed.), *Semantic Information Processing*, MIT Press, Cambridge, MA, 410–417, 1968.

[140] J. McCarthy, *Modality - Si! Modal Logic - No!*. Unpublished manuscript, available at `http://www-formal.stanford.edu/jmc/modality.html`, 1999.

[141] J. McCarthy and P. Hayes, Some philosophical problems from the standpoint of artificial intelligence. B. Meltzer and D. Michie (eds.), *Machine Intelligence 4*, Edinburgh Press, Edinburgh, Scotland, 463–502, 1969.

[142] D. McDermott and J. Doyle, Non-monotonic logic I. *Artificial Intelligence* **13**, 41–72, 1980.

[143] D. McDermott, Non-monotonic logic II. *Journal of the ACM*, **29**(1), 33–57, 1982.

[144] E. Mendelson, *Introduction to Mathematical Logic*. Van Nostrand Reinhold Company, New York, 1964.

[145] R. Montague. Pragmatics. In Klibansky, R. (Ed.), *Contemporary Philosophy*, Firenze: La Nuova Italia Editrice, pages 102—122, 1968.

[146] R. C. Moore,, The role of logic in knowledge representation and commonsense reasoning. *Proceedings of the National Conference of the American Association for Artificial Intelligence (AAAI-82)*, AAAI Press/MIT Press, Cambridge, 428–433, 1982.

[147] R. C. Moore, Possible world semantics for autoepistemic logic. *Proceedings of the 1st Non-Monotonic Reasoning Workshop*, New Paltz, NY, 344–354, 1984.

[148] R. C. Moore, Semantical considerations on nonmonotonic logic. *Artificial Intelligence* **25**, 75–94, 1985.

[149] R. C. Moore, A formal theory of knowledge and action. J. R. Hobbs and R. C. Moore (eds.), *Formal Theories of the Commonsense World*. Ablex, Norwood, NJ, 319–358, 1985.

[150] I. N. F. Niemelä, Constructive tightly grounded autoepistemic reasoning. *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Morgan Kaufmann, San Francisco, 399–404, 1991.

[151] I. N. F. Niemelä, On the decidability and complexity of autoepistemic reasoning. *Fundamenta Informaticae*, **17**, 117–155, 1992.

[152] N. Nilsson, *Principles of Artificial Intelligence*. Tioga Publishing Company, Palo Alto, 1980.

[153] P. Patel-Schneider. A decidable first-order logic for knowledge representation. In *Proc. of IJCAI-85*, pages 455–458, 1985.

[154] P. F. Patel-Schneider, *Decidable, Logic-Based Knowledge Representation*. Ph. D. thesis, Department of

Computer Science, University of Toronto, 1987.

[155] I. Pratt-Hartmann, Total Knowledge *Proceedings of the National Conference of the American Association for Artificial Intelligence (AAAI-00)*, AAAI Press/MIT Press, Cambridge, 423–428, 2000.

[156] Z. Pylyshyn, *Computation and Cognition: Toward a Foundation for Cognitive Science*. MIT Press, Cambridge, 1984.

[157] W. Quine, Quantifiers and propositional attitudes. [127], 101–111, 1971.

[158] R. Reiter, On closed world data bases. H. Gallaire and J. Minker (eds.), *Logic and Data Bases*. Plenum Press, 55–76, 1978.

[159] R. Reiter, A logic for default reasoning. *Artificial Intelligence* **13**(1–2), 81–132, 1980.

[160] R. Reiter, What should a database know?. *Journal of Logic Programming*, **14**(1-2), 127–153, 1992.

[161] R. Reiter, The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. V. Lifschitz (ed.), *Artificial Intelligence and Mathematical Theory of Computation*, Academic Press, 359–380, 1991.

[162] R. Reiter, *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. MIT Press, forthcoming.

[163] R. Rosati, Complexity of only knowing: the propositional case. *Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-97)*, 76-91, LNAI 1265, Springer-Verlag, 1997.

[164] R. Rosati, On the decidability and complexity of reasoning about only knowing. *Artificial Intelligence*, **116**(1-2), 193–215, 2000.

[165] R. Routley and R. K. Meyer, The semantics of entailment, I. H. Leblanc (ed.), *Truth, Syntax, and Semantics*, North-Holland, 194–243, 1973.

[166] R. Routley and V. Routley, Semantics of first degree entailment. *Noûs* **6**, 335–359, 1972.

[167] B. Russell, On denoting. *Mind*, **14**, 479–493, 1905.

[168] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020.

[169] Erik Sandewall. *Features and Fluents. The Representation of Knowledge about Dynamical Systems*. Oxford University Press, 1994.

[170] R. B. Scherl and H. J. Levesque, The frame problem and knowledge producing actions. *Proceedings of the National Conference on Artificial Intelligence (AAAI-93)*, AAAI Press/MIT Press, Cambridge, 689–695, 1993.

[171] C. Schwering. A Reasoning System for a First-Order Logic of Limited Belief. In *Proc. of IJCAI-17*, 1247–1253, 2017.

[172] C. Schwering and G. Lakemeyer Projection in the Epistemic Situation Calculus with Belief Conditionals. In *Proc. of AAAI-15*, 1583–1589, 2015.

[173] D. Scott. Advice on Modal Logic. In Lambert, K. (Ed.), *Philosophical Problems in Logic*, Dordrecht, Holland: D.Reidel, 1970.

[174] K. Segerberg, Some modal reduction theorems in autoepistemic logic. *Uppsala Prints and Preprints in Philosophy*, Uppsala University, 1995.

[175] B. Smith, *Reflection and Semantics in a Procedural Language*. Ph. D. thesis and Technical Report MIT/LCS/TR-272, MIT, Cambridge, 1982.

[176] G. Shvarts, Autoepistemic modal logic. *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK-III)*, Morgan Kaufmann, San Francisco, 97–109, 1990.

[177] R. Smullyan, *First-Order Logic*. Springer-Verlag, New York, 1968.

[178] R. F. Stärk, On the existence of fixpoints in Moore's autoepistemic logic and the non-monotonic logic of McDermott and Doyle. *Proceedings of the 4th Workshop on Computer Science Logic*, LNCS 533, Springer Verlag, Berlin, 354–365, 1991.

[179] R. Stalnaker, *Inquiry*. MIT Press, Cambridge, 1987.

[180] R. Stalnaker, A note on non-monotonic modal logic. *Artificial Intelligence* **64**(2), 183–196, 1993.

[181] D. Subramaniam, R. Greiner, and J. Pearl (eds.), *Artificial Intelligence, Special Issue on Relevance*, **97**(1–2), 1997.

[182] Michael Thielscher. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence*, 111(1–2):277–299, 1999.

[183] M. Truszczynski, Modal interpretations of default logic. *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Morgan Kaufmann, San Francisco, 393–398, 1991.

[184] M. Y. Vardi, The complexity of relational query languages (Extended Abstract). *Proc. of the 14th annual ACM symposium on Theory of computing STOC-82*, 137-–146, 1982.

[185] A. Waaler, *Logical Studies in Complementary Weak S5*. Ph. D. Thesis, University of Oslo, 1995.

[186] T. Winograd, Frame representations and the declarative/procedural controversy. D. Bobrow and A. Collins (eds.), *Representation and Understanding: Studies in Cognitive Science*, Academic Press, New York, 185–210, 1975.

Index